. 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
0 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.

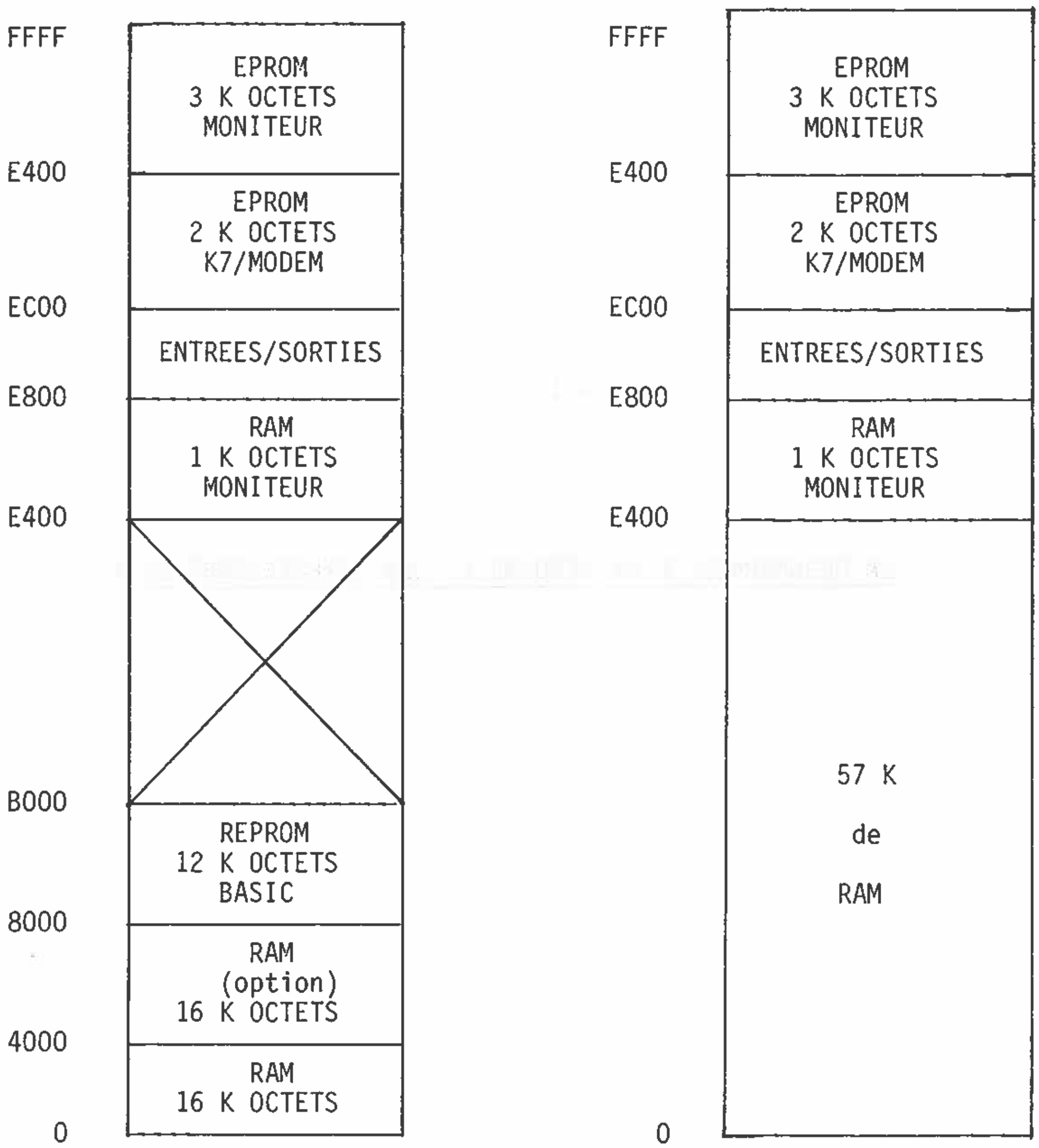
S O M M A I R E

- I -

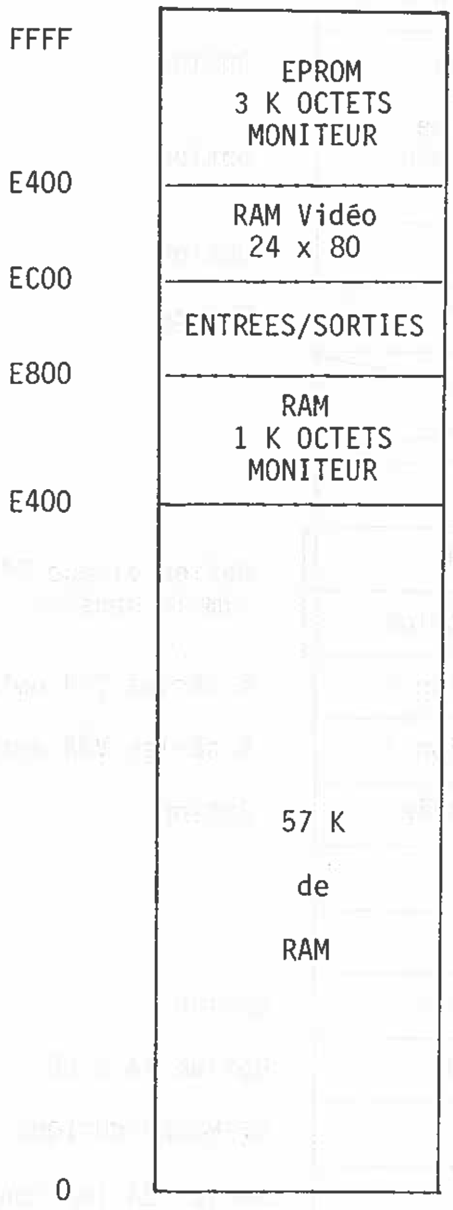
- LA GEOGRAPHIE DE LA MEMOIRE ET DES ENTREES-SORTIES -

VERSION DE BASE

VERSION 16 x 64



VERSION 24 x 80



Adresse	Contenu	Remarque
EBFF	Validation mémoires	
EBF8		
EB30	1795	Option disque 8" ou 5" double densité
EB20	DMA	
EB00	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	Option disque 5" simple densité
E8F0	Disque 5" sélection	
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	$\overline{GF}$ 6845 du 24 x 80	Option 24 x 80
E870	$\overline{GE}$ VIA	K7/MODEM/MUSIQUE
E860	$\overline{GD}$ 6551	Série V24 (option)
E850	$\overline{GC}$ VIA	Sortie parallèle et clavier
E840	$\overline{GB}$ 8279	Clavier 2
E830	$\overline{GA}$ 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

Données

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	

Alimentations

Adresses

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/

Données

SCHEMA DE BROCHAGE D'UN CONNECTEUR

Signification des codes précédents :

VCC	: + 5V
0V	: masse
RD/	: lecture
WD/	: écriture
VMA/	: valid memory address
Φ 2	: 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)

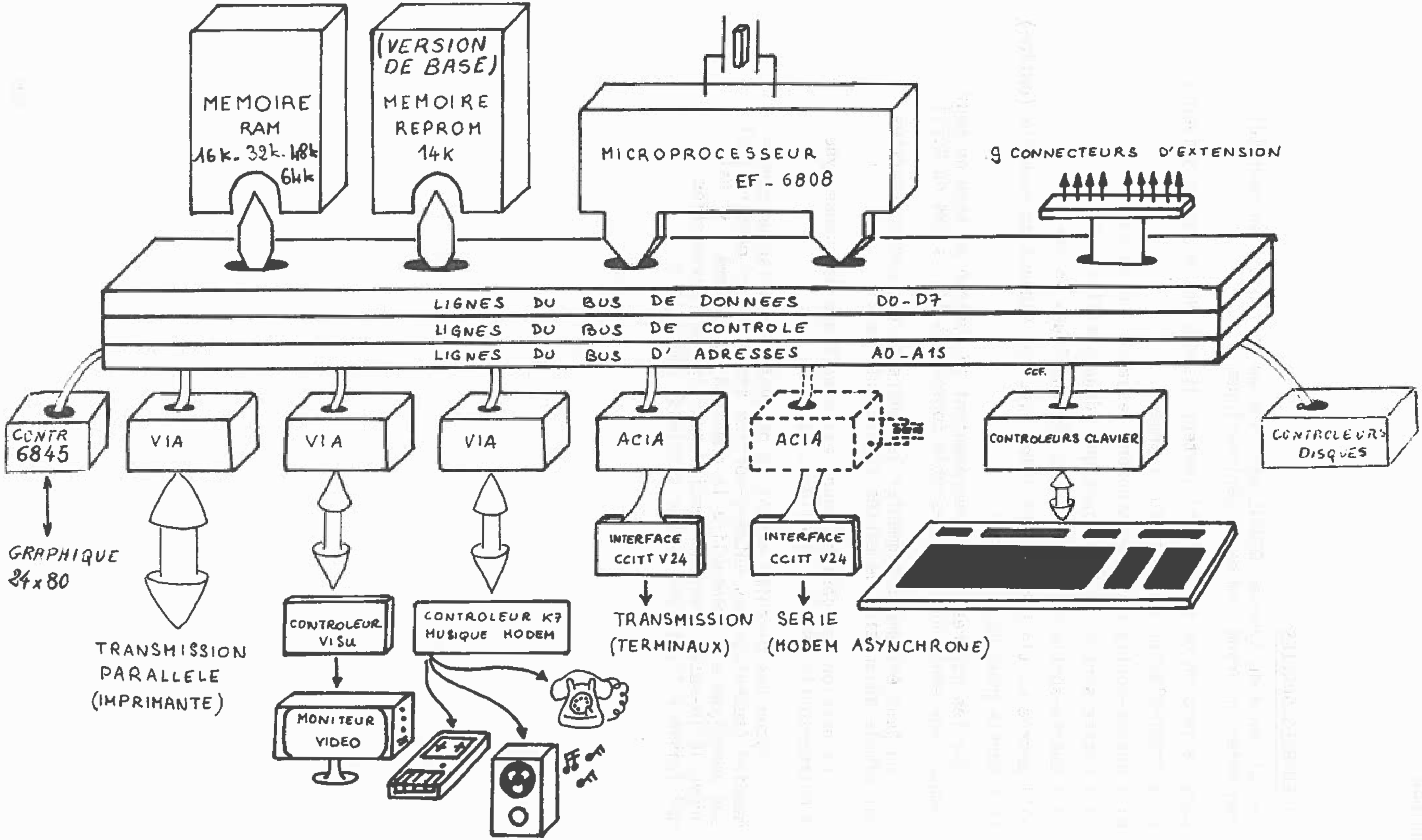
f) 1 sortie pour haut parleur.

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, $\overline{BA}$ (Bus Available), $\overline{Halt}$ (Halte), $\overline{IRQ}$ (Interrupt Request), $\overline{NMI}$ (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 $\overline{VMA}$ (Valid Memory Acc ess), protégé du Reset par IC 6D, est envoyé sur le bus.

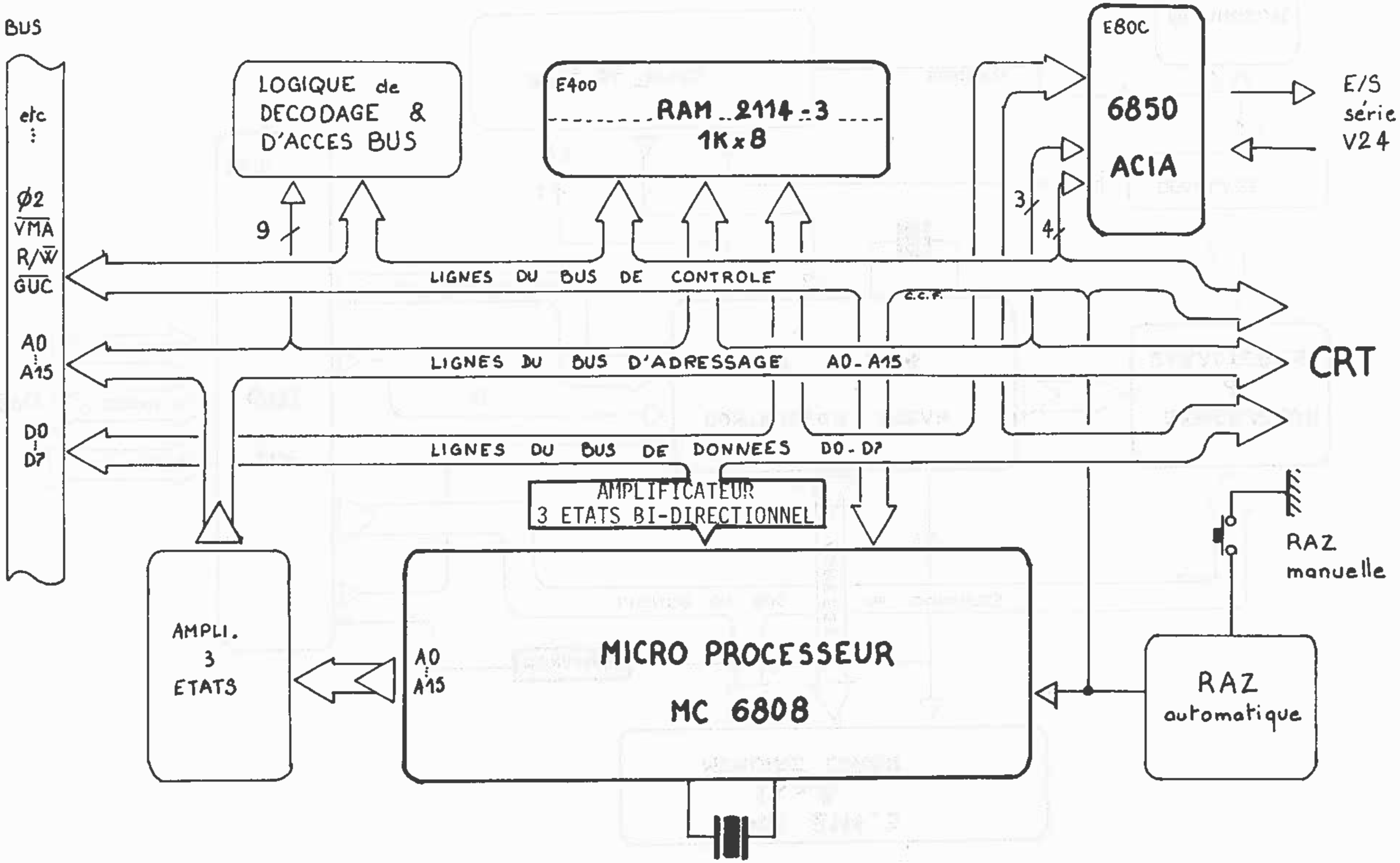
Il existe un dispositif de remise à zéro automatique, constitué par IC 6E et quelques composants passifs. Le transistor Q 1 permet de piloter l'ensemble des cartes connectées sur le bus.
Un bouton-poussoir placé sur le coté droit sous le GOUPIL autorise la réinitialisation manuelle du système (bouton 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 $\overline{GUC}$ (Groupe Unité Centrale) généré sur la carte d'entrée-sortie.

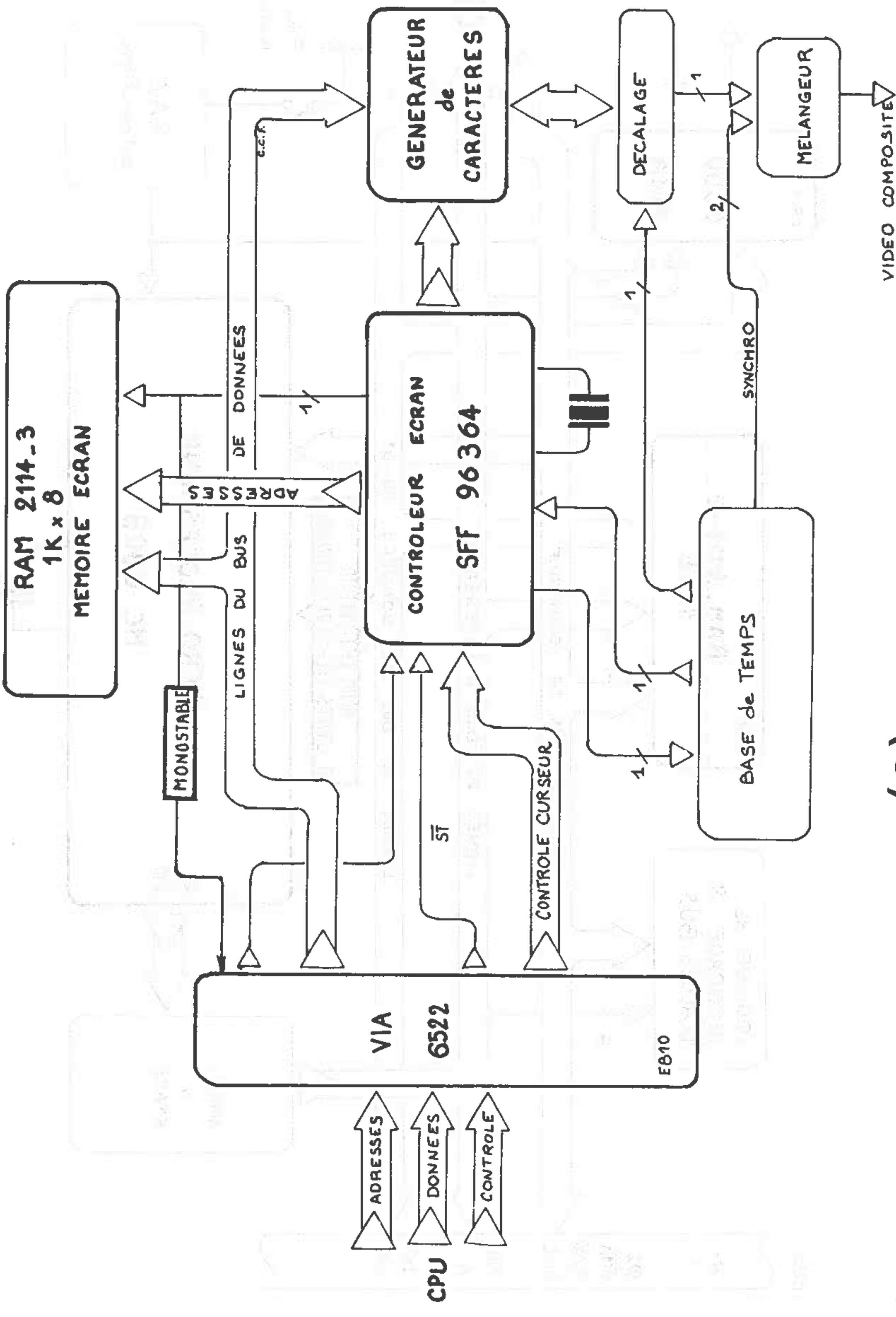
Retenons l'adresse de l'ACIA : E80C
et l'adresse de la RAM : E400.

Il faut remarquer que le boitier IC6A génère sur le bus six signaux de décodage appelés : $\overline{GA}$, $\overline{GB}$, $\overline{GC}$, $\overline{GD}$, $\overline{GE}$, $\overline{GF}$.
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.


BUS
etc
Ø2
VMA
R/W̄
GUC
A0
A15
D0
D7
LOGIQUE de
DECODAGE &
D'ACCES BUS
9
E400
RAM 2114-3
1K x 8
E80C
6850
ACIA
3
4
E/S
série
V24
LIGNES DU BUS DE CONTROLE
c.c.F.
LIGNES DU BUS D'ADRESSAGE A0-A15
CRT
LIGNES DU BUS DE DONNEES D0-D7
AMPLIFICATEUR
3 ETATS BI-DIRECTIONNEL
RAZ
manuelle
AMPLI.
3
ETATS
A0
A15
MICRO PROCESSEUR
MC 6808
RAZ
automatique

CARTE CPU (1)

smt

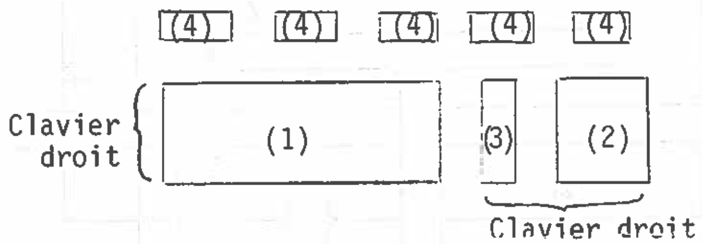


CARTE CPU (2)

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 héxadécimal, plus clavier de fonctions, plus clavier de gestion de la page).



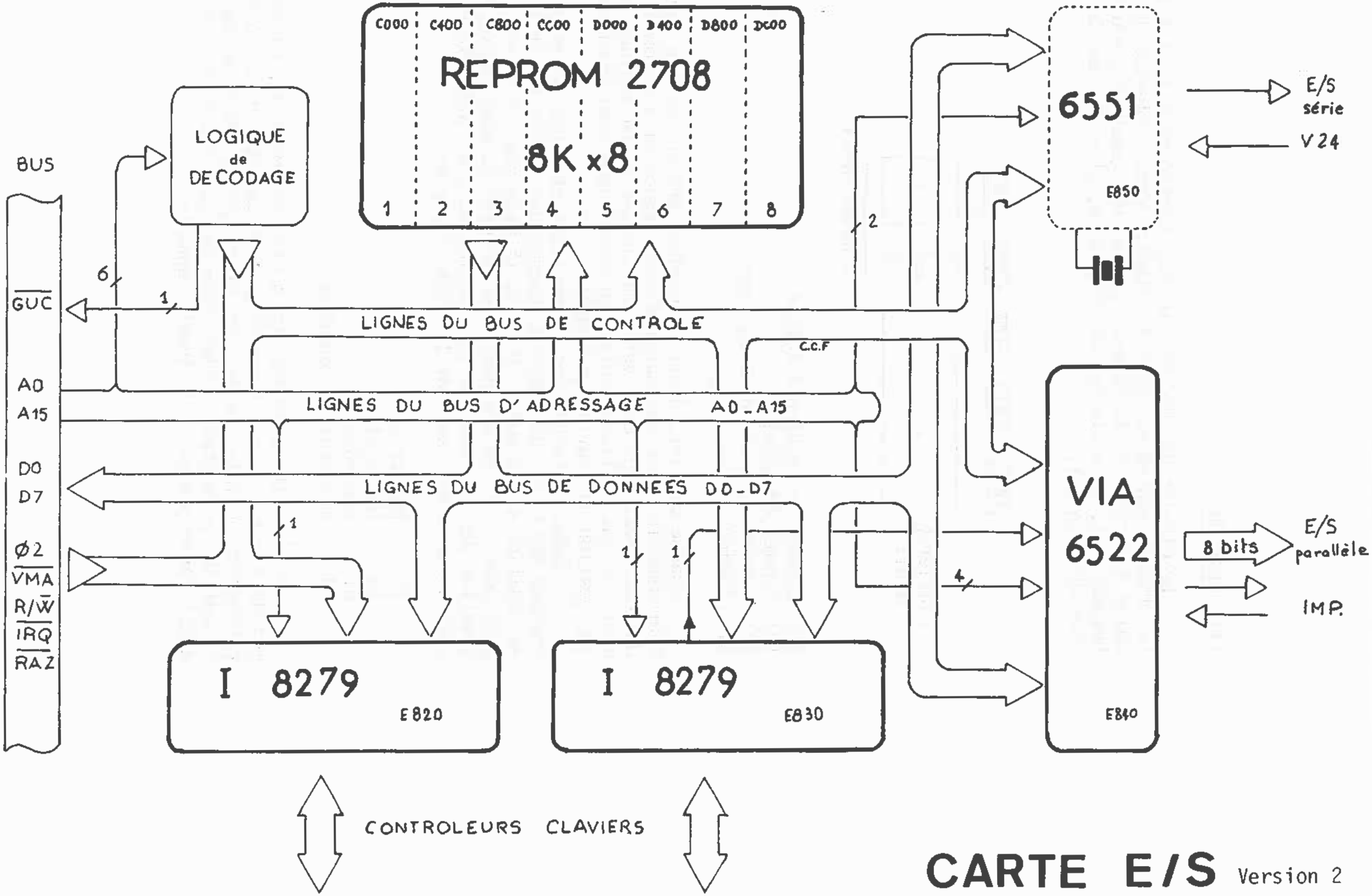
(1) clavier de machine à écrire
(2) clavier héxadécimal
(3) clavier de gestion de la page
(4) clavier de fonction

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 :

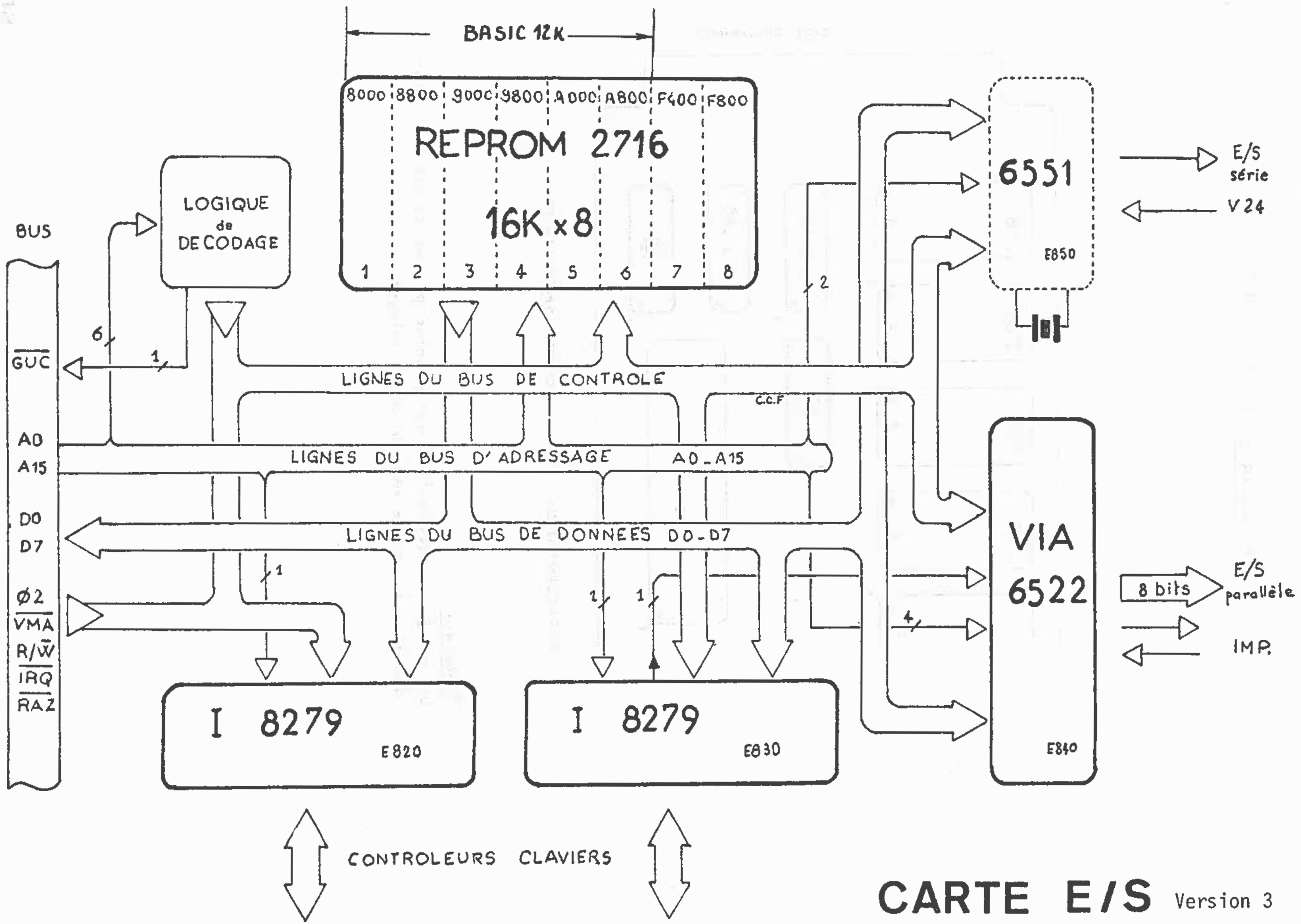
11	Majuscule
10	Majuscule
01	Contrôle
00	Majuscule + Contrôle

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

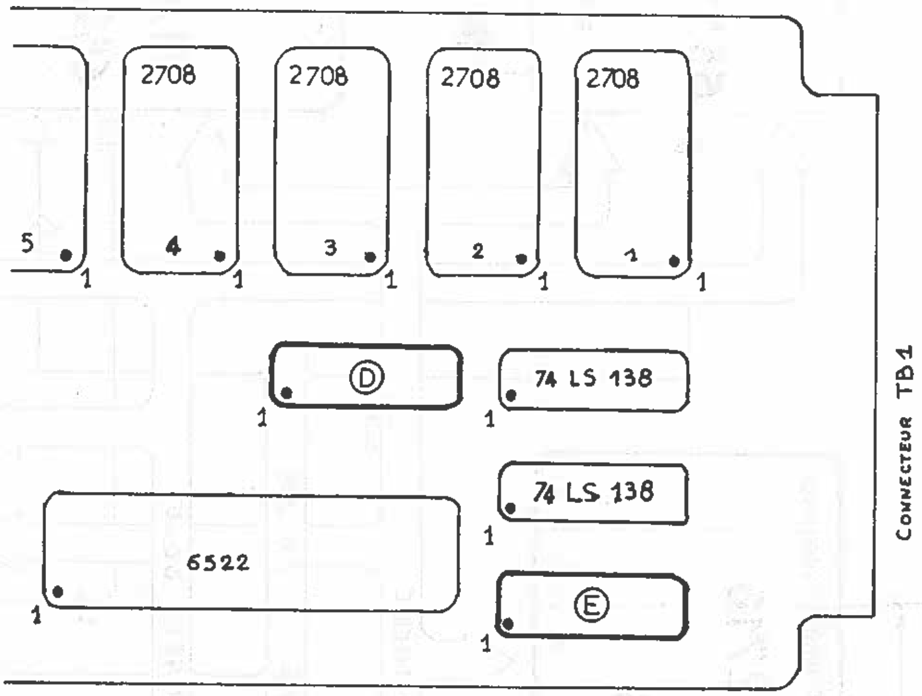


CARTE E/S Version 3

smt

UTILISATION DES PLOTS PROGRAMMABLES D ET E

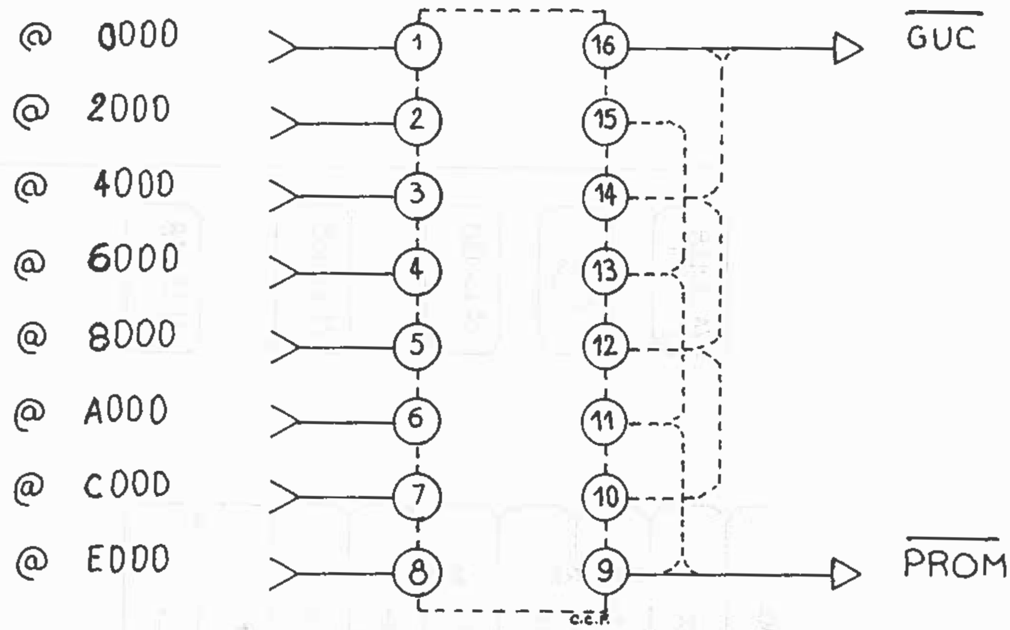
POSITION SUR LA CARTE E/S Version 2



REPRESENTATION VUE DU COTE COMPOSANTS

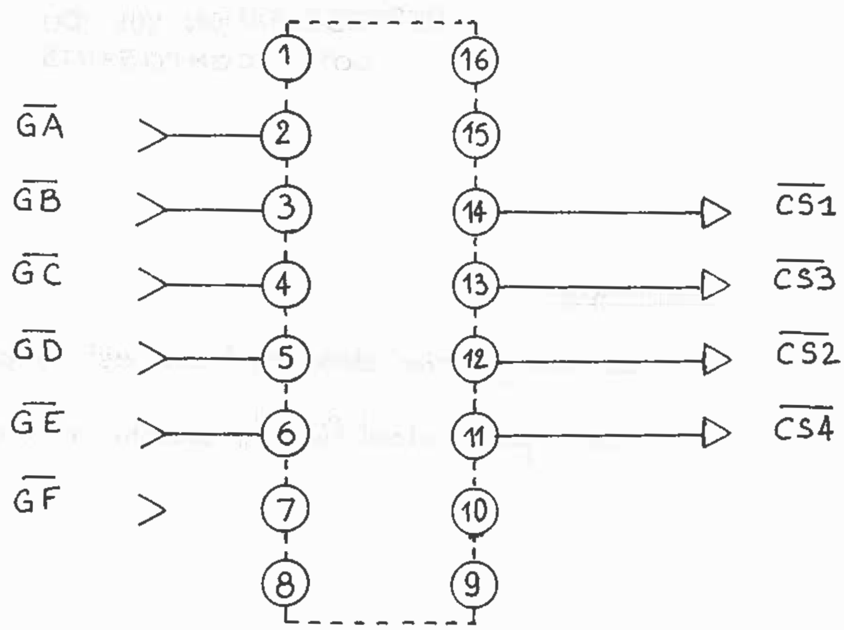
Remarque :
Le coin gauche des boitiers est accentué par un cercle noir pour identifier la broche n°1 du circuit intégré.

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



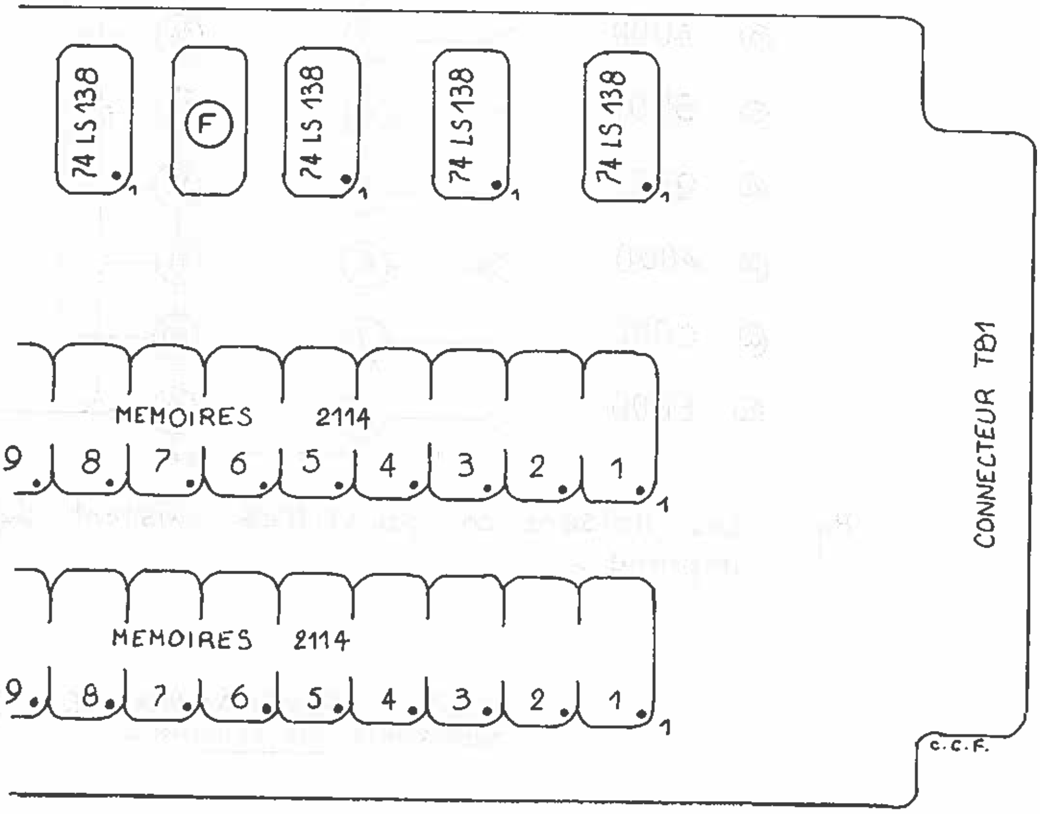
Rq: Les liaisons en pointillés existent déjà sur le circuit imprimé.

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



UTILISATION du PLOT PROGRAMMABLE (F)

POSITION SUR LA CARTE MEMOIRE STATIQUE



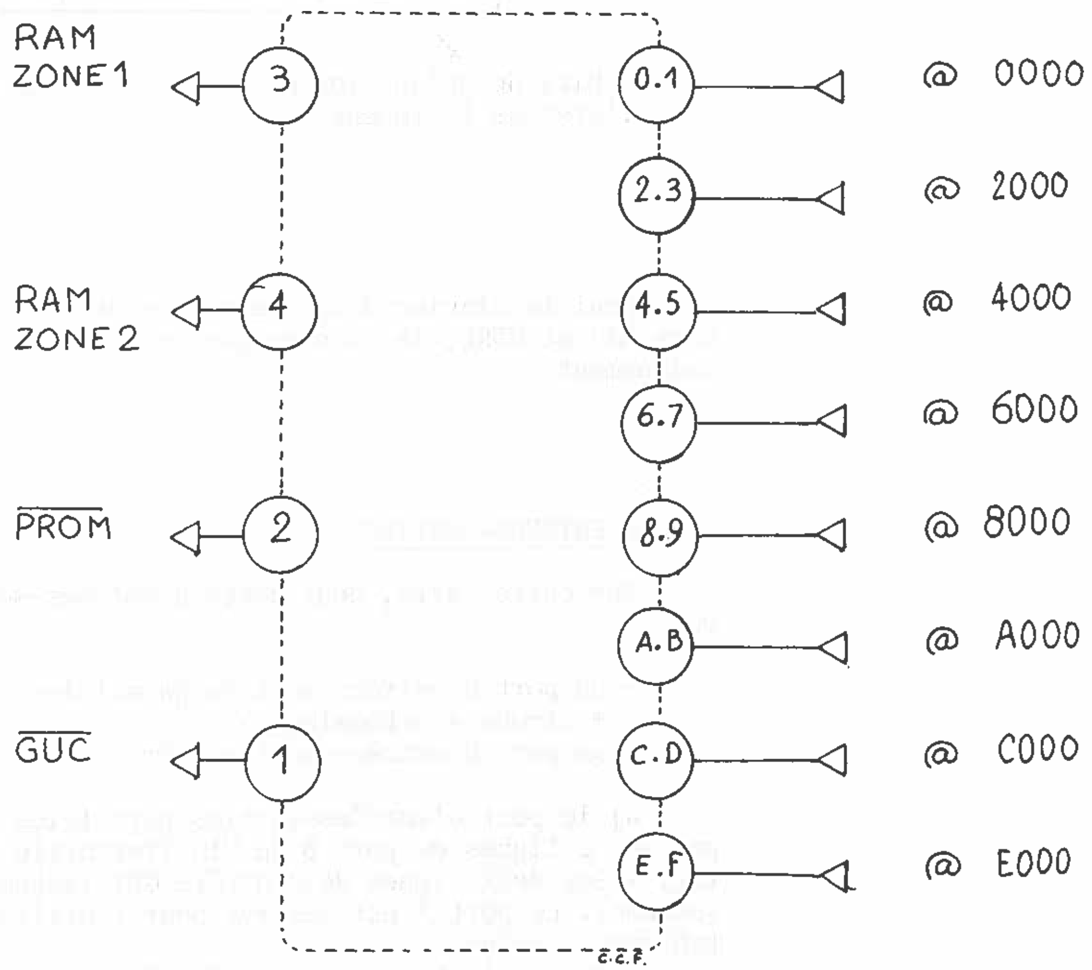
REPRESENTATION VUE DU
COTE COMPOSANTS

Remarque:

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

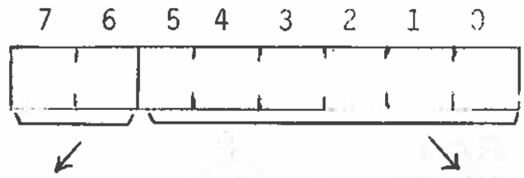
PLOT PROGRAMMABLE (F)

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 (GD et 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/$\overline{\text{Acknowlegde}}$
2	PA_1	11	PB_0/Bit 1	20	CB_2/$\overline{\text{Strobe}}$
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)

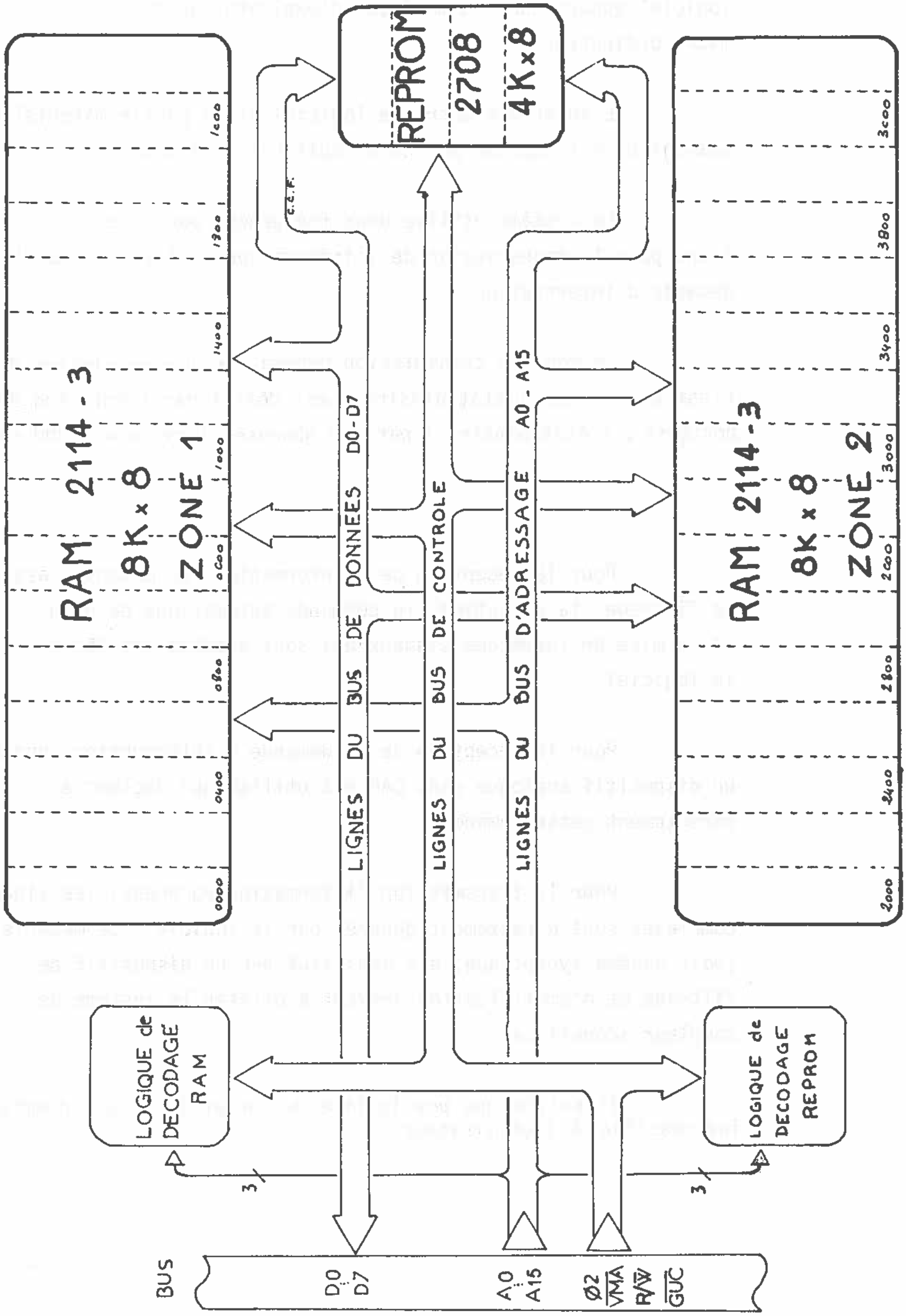
Le décodage des mémoires RAM et REPROM est effectué par quatre 74LS138 qui sont des décodeurs 3 parmi 8 et qui permettent d'accéder aux 16 k RAM et aux 4 k REPROM n'importe où dans l'espace adressable du microprocesseur.

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

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

SMT

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).

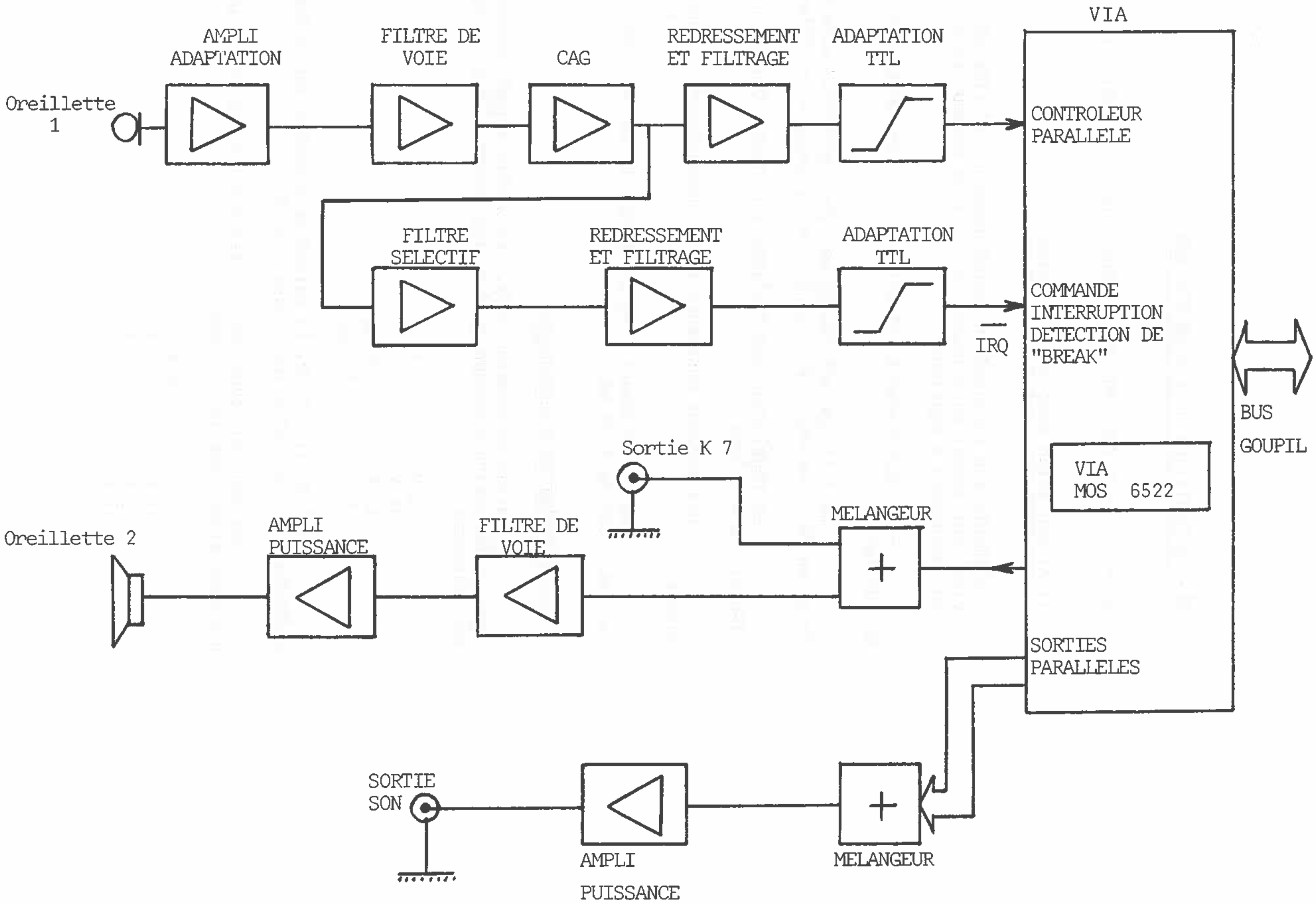
Pour la réception de l'information, ce matériel assure le filtrage, la détection, la commande automatique de gain et la mise en forme des signaux qui sont ensuite traités par le logiciel.

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 régler le niveau d'émission inaccessible à l'utilisateur.

SYNOPTIQUE DU COUPLEUR ACOUSTIQUE GOUPIL



SMT

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 :

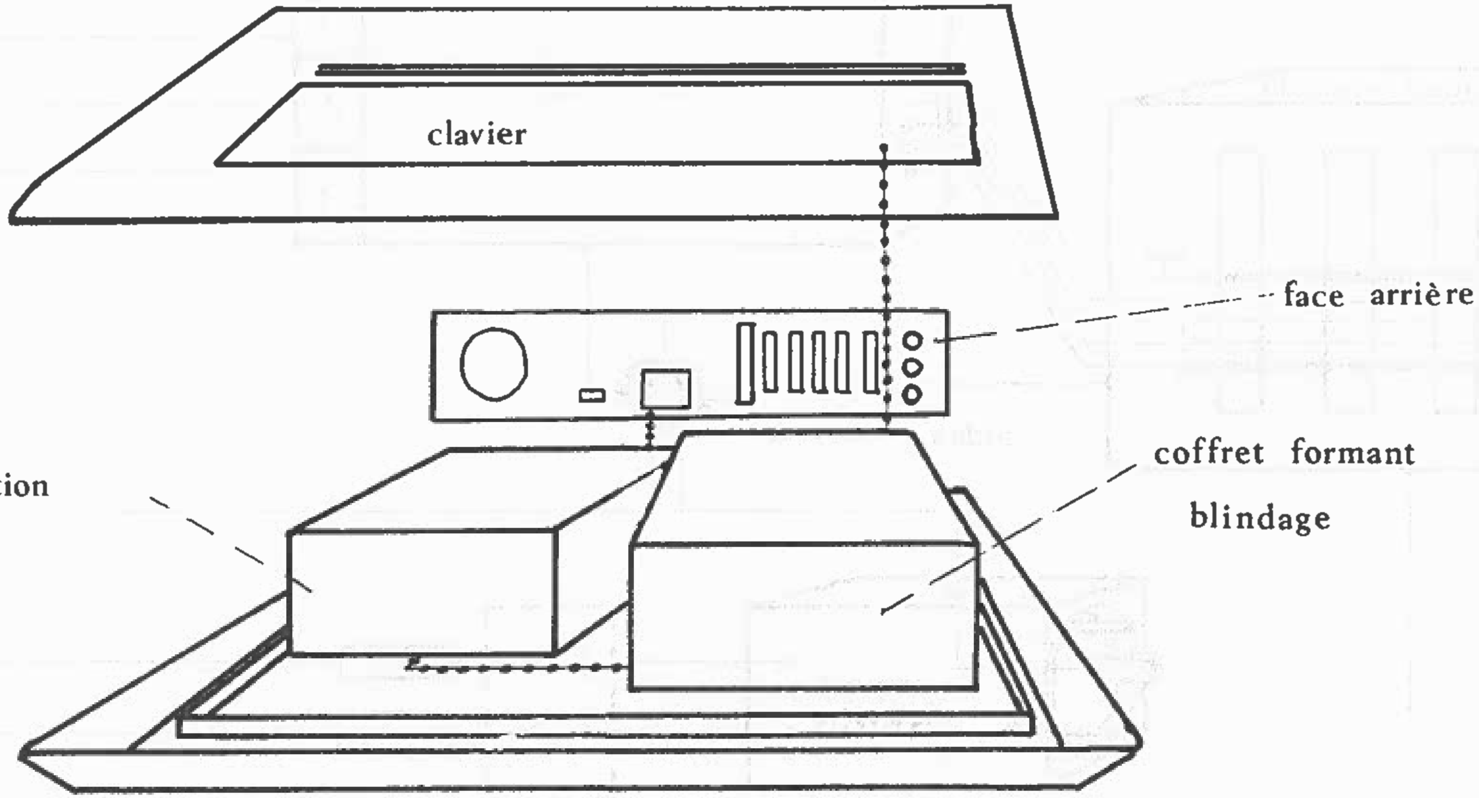
. 5 V	6 A
. 12 V	2 A
. -12 V	500 ma
. - 5 V	500 ma

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

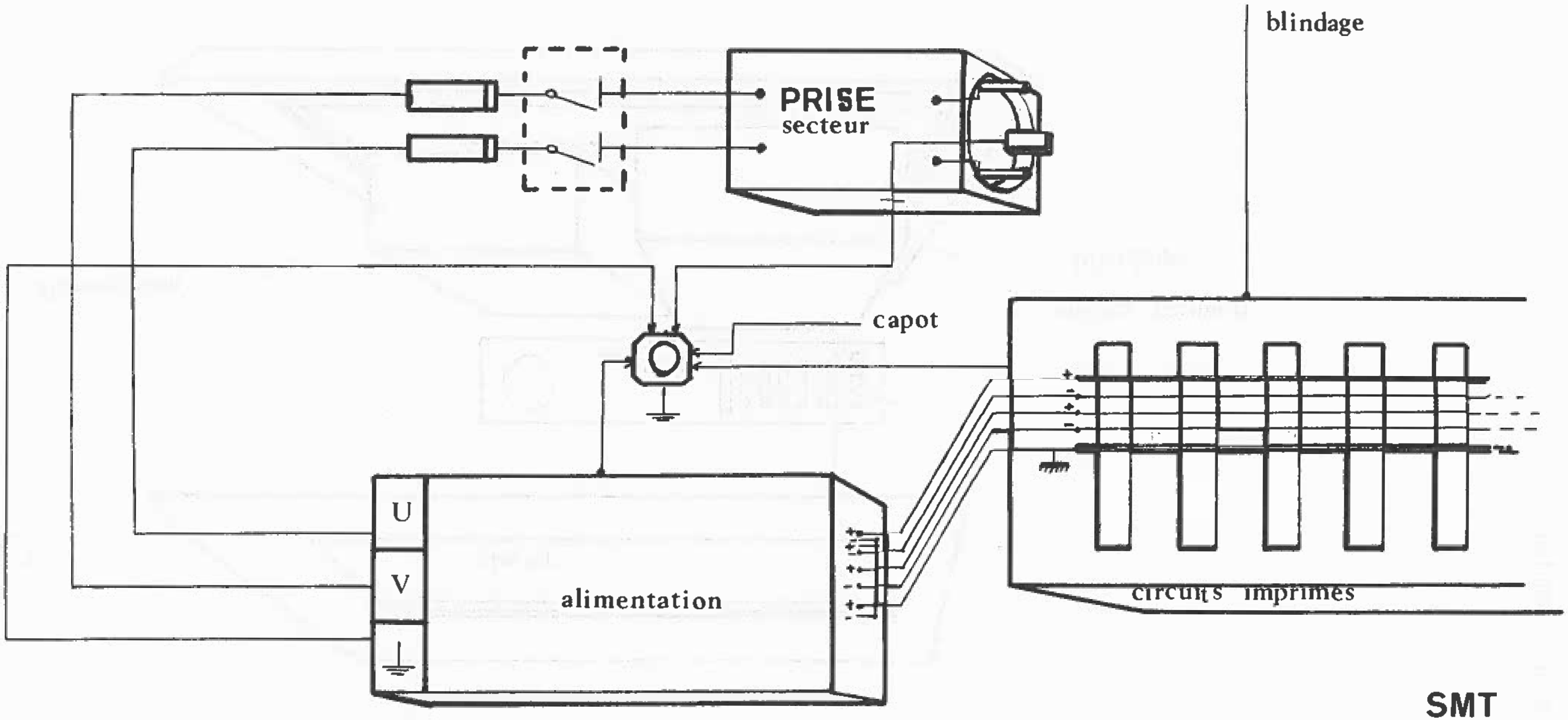


SCHEMA PHYSIQUE DES CIRCUITS
D'ALIMENTATION

LIAISONS EQUIPOTENTIELLES SIMPLES

SCHEMA FONCTIONNEL DES CIRCUITS D'ALIMENTATION

A 2



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
                    positionnement sans écriture sur écran
LDAA    # 0
STAA    $E7FB

STAA    $E7F8       positionnement avec écriture
```

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

<u>Description du registre de commande C</u>

<u>Bit 7</u> : 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.

<u>Bits 0, 1, 2</u> : 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

<u>Bits 3, 4, 5</u> 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 :

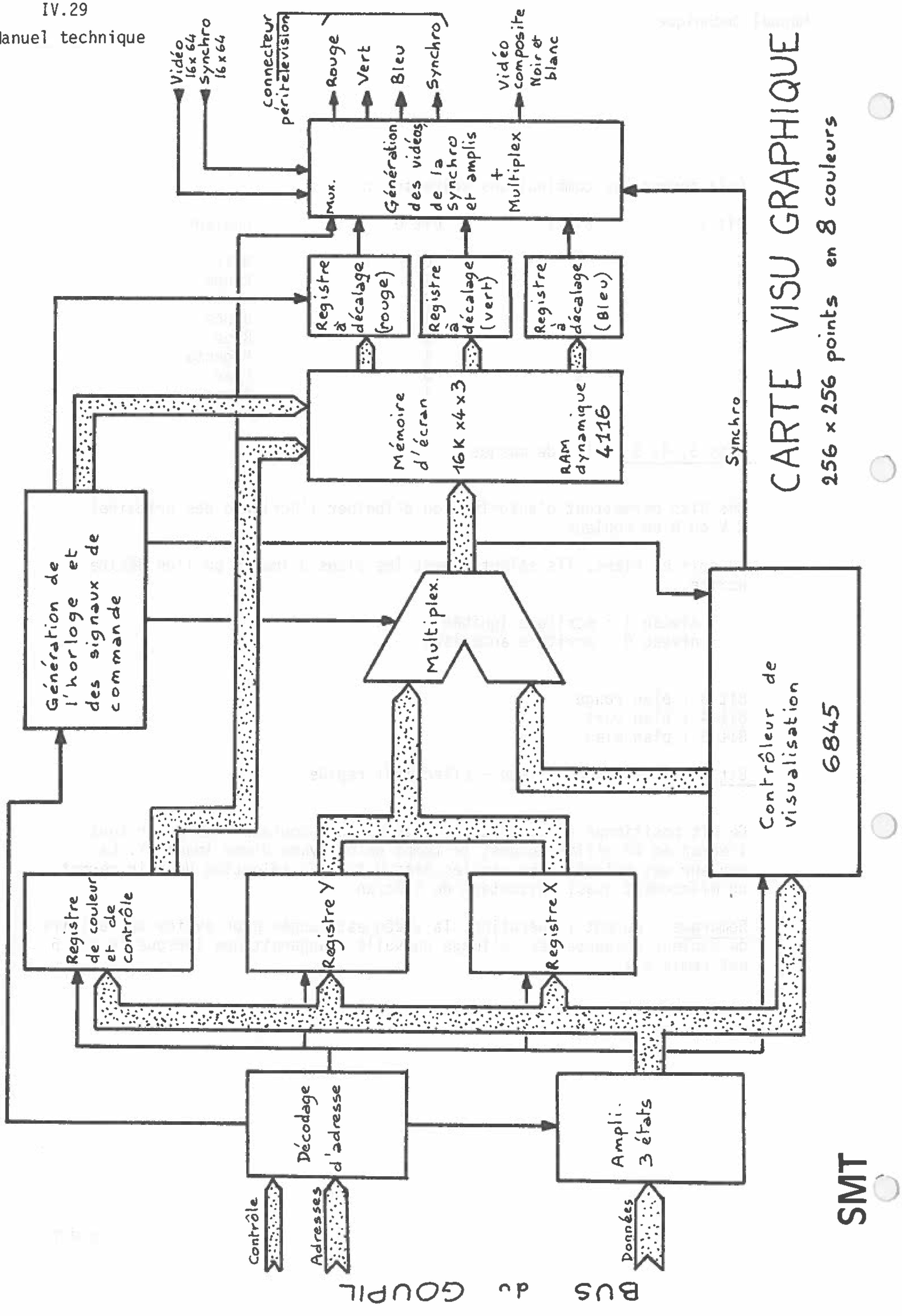
niveau 1 : écriture inhibée
niveau 0 : écriture autorisée

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

<u>Bit 6 :</u> création d'un fond - effacement rapide

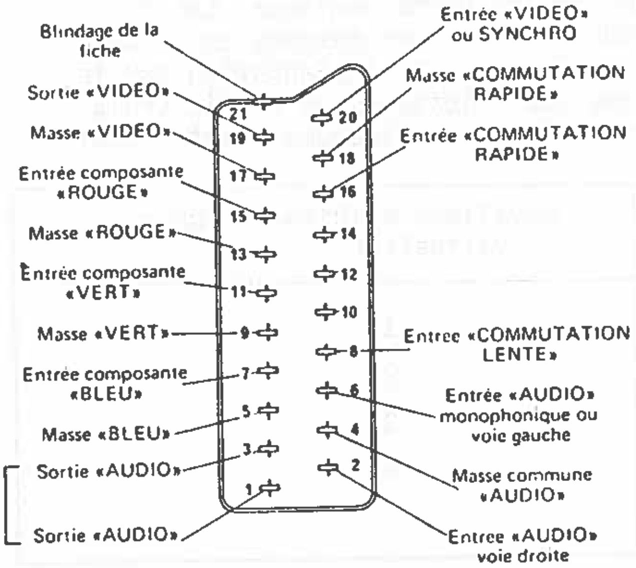
Ce bit positionné à 0 permet l'écriture d'une couleur donnée sur tout l'écran en 20 millisecondes, le temps de balayage d'une image TV. La couleur est sélectionnée par les bits 0 à 5. La sélection du noir permet un effacement quasi instantané de l'écran.

<u>Remarque</u> : 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
Vidéo 16x64
Synchro 16x64
connecteur péritélevision
Rouge
Vert
Bleu
Synchro
Vidéo composite Noir et blanc
Mux.
Génération des vidéos, de la Synchro et amplis + Multiplex
Registre à décalage (rouge)
Registre à décalage (vert)
Registre à décalage (Bleu)
Mémoire d'écran 16K x4 x3
RAM dynamique 4116
Synchro
Génération de l'horloge et des signaux de commande
Multiplex.
Contrôleur de visualisation 6845
Registre de couleur et de contrôle
Registre Y
Registre X
Décodage d'adresse
Ampli. 3 états
Contrôle
Adresses
Données
BUS du GOUPIL

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.

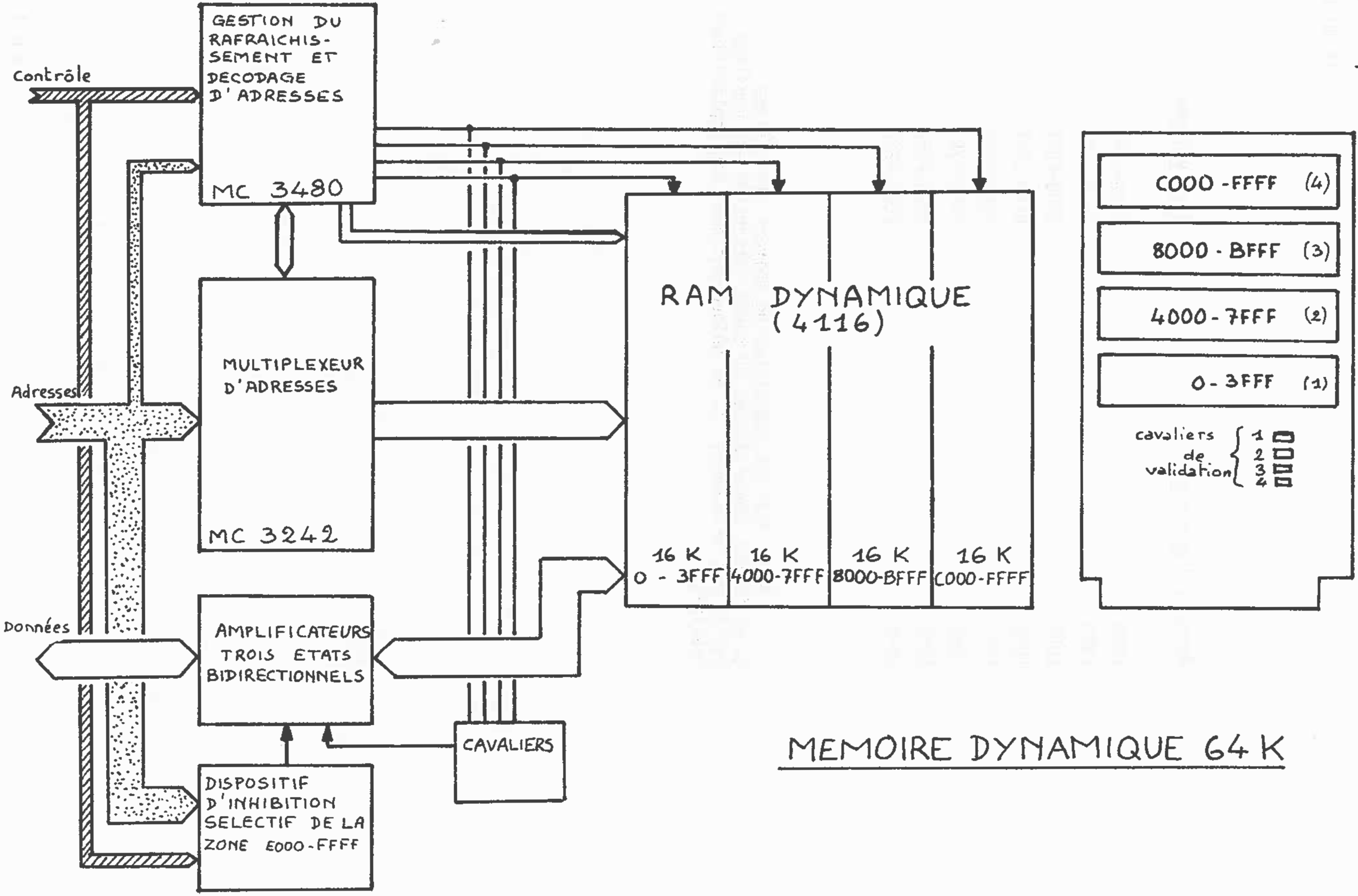
Zone équipée de boitiers	Cavaliers à installer pour validation
0 - 3FFF	1
4000 - 7FFF	2
8000 - BFFF	3
C000 - FFFF	4

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 $\not{0}$ 01 en EBF8 valide la zone E000-E3FF, l'écriture de $\not{0}$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 :

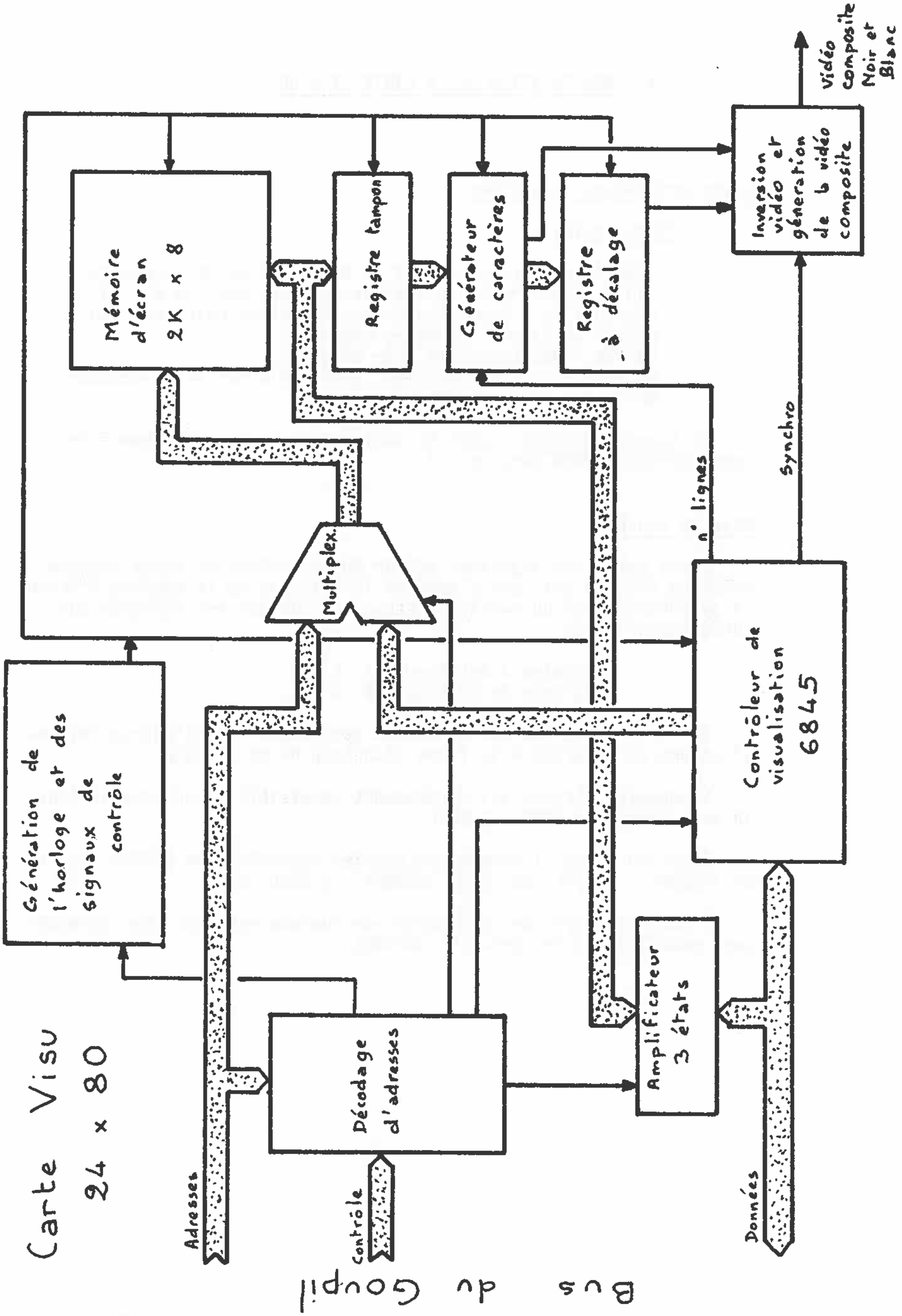
Registre d'adresses : $ E 870
Registre de données : $ E 871

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
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
Décodage d'adresses
Amplificateur 3 états
Adresses
Contrôle
Données
Bus du Goupil

SMT

- 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 : ± Mb̸ AAAA

AAAAb̸NNb̸XXZ

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 :

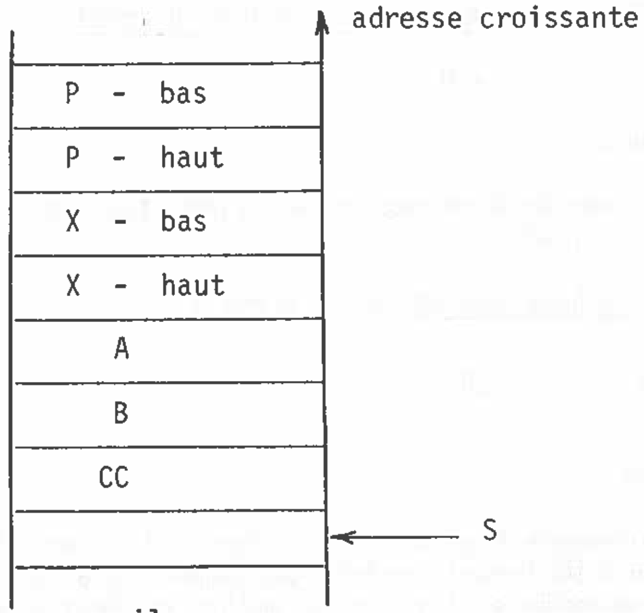
La commande R permet d'afficher le contenu des registres courants du programme utilisateur, selon le format :

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 :



-pile-

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 : $\pm$ 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 : $\pm$ 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 :

± B⌀1000⌀1100

9. COMMANDES D'EXECUTION

9.1. Exécution d'un programme

Syntaxe : ±G⌀AAAA

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 :

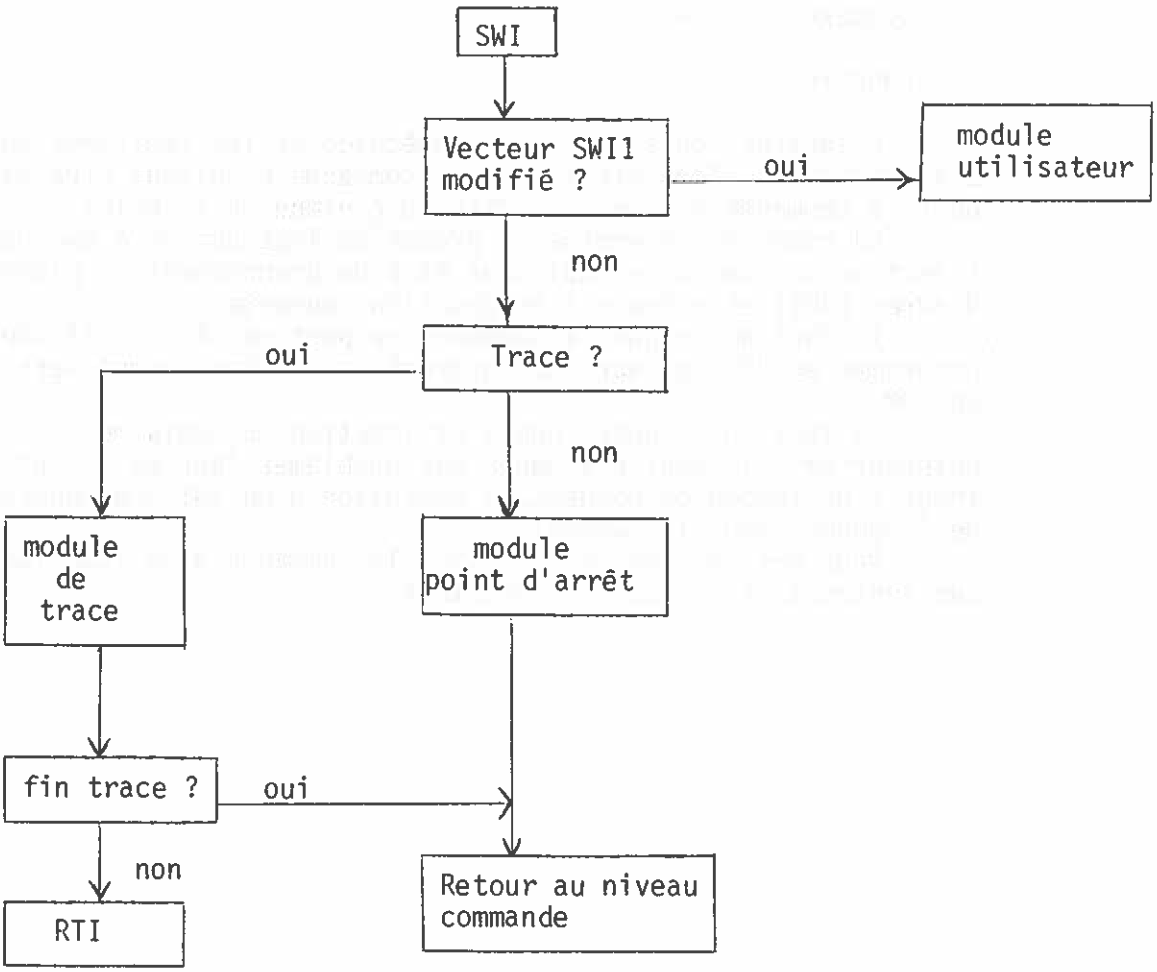
La commande T permet de "tracer" le nombre d'instructions données par NNNN (4 chiffres hexadécimaux) à partir de l'instruction courante en affichant les registres à l'exécution de chaque instruction. Les commentaires donnés pour N concernent aussi la commande T.

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 E40A-E40B (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

  008C        SKIP2   EQU     $8C          SKIP TWO BYTES




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

  F8B0                ORG     $F8B0
                      NAM     SCREEN HANDLER
                      OPT     PAG

              *********************************
              *                               *
              *        SCREEN HANDLER         *
              *                               *
              *********************************


              *
              *    I/O ROUTINES
              *    ------------
              *
              *        INITVS    INITIALIZATIONS
              *        OUTPUT    OUPUT ONE CHARACTER
              *        BIPBIP    OUTPUT BELL
              *
              *
              * EQUATES
              *
  E810        VIAVIS  EQU    $E810
```

```
                   *
                   *  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

                       *
                       * OUTPUT ONE CHARACTER IN A REGISTER
                       *
 F8EF FF E4 C3  OUTPUT  STX    SAVEX
 F8F2 36                PSH A
 F8F3 37                PSH B
 F8F4 4D                TST A
 F8F5 27 59             BEQ    OUTPT5
 F8F7 81 9F             CMP A  £$9F     FUNCTIONS ?
 F8F9 22 55             BHI    OUTPT5
 F8FB 81 1F             CMP A  £$1F     CONTROL ?
 F8FD 22 35             BHI    OUTPT3
 F8FF 81 07             CMP A  £7
 F901 26 04             BNE    OUTPT1
 F903 8D 5C             BSR    BIPBIP
 F905 20 49             BRA    OUTPT5
                * CONTROL CHARACTERS
 F907 CE F8 B0  OUTPT1  LDX    £TABVIS
 F90A FF E4 C0          STX    PTTAB
 F90D BB E4 C1          ADD A  PTTAB+1
 F910 24 03             BCC    OUTPTS
 F912 7C E4 C0          INC    PTTAB
 F915 B7 E4 C1  OUTPTS  STA A  PTTAB+1
 F918 FE E4 C0          LDX    PTTAB
 F91B A6 00             LDA A  0,X
 F91D 81 0F             CMP A  £$F
 F91F 27 2F             BEQ    OUTPT5
 F921 B7 E8 10  OUTPT2  STA A  VIAVIS+ORB
 F924 8D 30             BSR    STROBE
 F926 84 07             AND A  £7
 F928 26 05             BNE    OUTPT6
                * CLEAR SCREEN
 F92A CE 40 73          LDX    £$4073
 F92D 20 1E             BRA    OUTPT4
 F92F CE 04 0C  OUTPT6  LDX    £$40C
 F932 20 19             BRA    OUTPT4
                * STANDARD CHARACTERS
 F934 C6 0F     OUTPT3  LDA B  £$F
 F936 F7 E8 10          STA B  VIAVIS+ORB
 F939 B7 E8 11          STA A  VIAVIS+ORA
 F93C 8D 18             BSR    STROBE
 F93E F6 E8 11  OUTPT8  LDA B  VIAVIS+ORA
 F941 59                ROL B
 F942 24 FA             BCC    OUTPT8
 F944 F6 E8 11  OUTPT7  LDA B  VIAVIS+ORA
 F947 59                ROL B
 F948 25 FA             BCS    OUTPT7
 F94A FE E4 C5          LDX    VTEMP
 F94D 09        OUTPT4  DEX
 F94E 26 FD             BNE    OUTPT4
 F950 33        OUTPT5  PUL B
 F951 32                PUL A
 F952 FE E4 C3          LDX    SAVEX
```

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

F955 39                 RTS
                  *
                  *  OUTPUT STROBE
                  *
F956 C6 FC        STROBE  LDA B  £$FC
F958 F7 E8 1C             STA B  VIAVIS+PCR
F95B C6 FE                LDA B  £$FE
F95D F7 E8 1C             STA B  VIAVIS+PCR
F960 39                   RTS
                  *
                  * OUTPUT BELL
                  *
F961 86 7F        BIPBIP  LDA A  £$7F
F963 8D 0D        BIPSN1  BSR    BIPDEL
F965 53                   COM B
F966 F7 E8 61             STA B  VIAMUS+ORA
F969 8D 07                BSR    BIPDEL
F96B F7 E8 61             STA B  VIAMUS+ORA
F96E 4A                   DEC A
F96F 26 F2                BNE    BIPSN1
F971 39                   RTS
F972 C6 60        BIPDEL  LDA B  £$60
F974 5A           BIPSN2  DEC B
F975 26 FD                BNE    BIPSN2
F977 39                   RTS
                          END
```

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

     F980            ORG    $F980
                     NAM    KEYBOARD HANDLER
                     OPT    PAG

               ***********************************
               *                                 *
               *        KEYBOARD HANDLER         *
               *                                 *
               ***********************************


               *
               *    I/O ROUTINES
               *    ------------
               *
               *        INITCV   INITIALIZATIONS
               *        TINPUT   CHECK FOR TYPED CHARACTER
               *        INPUT    INPUT ONE CHARACTER
               *
               *
               * EQUATES
               *
     E840      VIACL   EQU    $E840
     E820      DATA1   EQU    $E820
     E830      DATA2   EQU    $E830
     E821      CMDE1   EQU    $E821
     E831      CMDE2   EQU    $E831
     E860      VIAMUS  EQU    $E860
     F403      GTEXT   EQU    $F403
     EBF8      SETMEM  EQU    $EBF8
     F400      GINIT   EQU    $F400
     0000      ORB     EQU    0
     0001      ORA     EQU    1
     0002      DDRB    EQU    2
     0003      DDRA    EQU    3
     0004      TILL    EQU    4
     000C      PCR     EQU    $C
     000D      IFR     EQU    $D
     000E      IER     EQU    $E
     00A1      SHIFT   EQU    $A1       SHIFT CODE
     00A0      REPT    EQU    $A0       REPEAT CODE
```

```
                      *
                      * KEYBOARD 1 TABLE DEFINITION
                      *
       F980           ADTAB1  EQU   *
       F980 1A                FCB   $1A,$04,$05,$13,$86,$03,$85,$18
       F988 12                FCB   $12,$07,$14,$06,$88,$02,$87,$16
       F990 19                FCB   $19,$0A,$15,$08,$8A,$9D,$89,$0E
       F998 09                FCB   $09,$11,$01,$00,$84,$17,$F1,$A1
       F9A0 09                FCB   $09,$0C,$0F,$0B,$8C,$9B,$8B,$9E
       F9A8 0A                FCB   $0A,$0D,$00,$7F,$08,$00,$1F,$20
       F9B0 10                FCB   $10,$1C,$1D,$0D,$9C,$00,$00,$1E
       F9B8 00                FCB   $00,$A0,$00,$00,$00,$00,$00,$00

       F9C0 1A                FCB   $1A,$04,$05,$13,$86,$03,$85,$18
       F9C8 12                FCB   $12,$07,$14,$06,$88,$02,$87,$16
       F9D0 19                FCB   $19,$0A,$15,$08,$8A,$9D,$89,$0E
       F9D8 09                FCB   $09,$11,$01,$00,$84,$17,$F1,$A1
       F9E0 09                FCB   $09,$0C,$0F,$0B,$8C,$9B,$8B,$9E
       F9E8 0A                FCB   $0A,$0D,$00,$7F,$08,$00,$1F,$20
       F9F0 10                FCB   $10,$1C,$1D,$0D,$9C,$00,$00,$1E
       F9F8 00                FCB   $00,$A0,$00,$00,$00,$00,$00,$00

       FA00 5A                FCB   $5A,$44,$45,$53,$33,$43,$32,$58
       FA08 52                FCB   $52,$47,$54,$46,$35,$42,$34,$56
       FA10 59                FCB   $59,$4A,$55,$48,$37,$3F,$36,$4E
       FA18 09                FCB   $09,$51,$41,$00,$31,$57,$F1,$A1
       FA20 49                FCB   $49,$4C,$4F,$4B,$39,$2F,$38,$2E
       FA28 0A                FCB   $0A,$0D,$00,$7F,$08,$00,$5F,$20
       FA30 50                FCB   $50,$25,$7E,$4D,$B1,$00,$30,$2B
       FA38 00                FCB   $00,$A0,$00,$00,$00,$00,$00,$00

       FA40 7A                FCB   $7A,$64,$65,$73,$22,$63,$7B,$78
       FA48 72                FCB   $72,$67,$74,$66,$28,$62,$27,$76
       FA50 79                FCB   $79,$6A,$75,$68,$7D,$2C,$83,$6E
       FA58 09                FCB   $09,$71,$61,$00,$26,$77,$F1,$A1
       FA60 69                FCB   $69,$6C,$6F,$6B,$82,$3A,$21,$3B
       FA68 0A                FCB   $0A,$0D,$00,$7F,$08,$00,$2D,$20
       FA70 70                FCB   $70,$7C,$5E,$6D,$29,$00,$40,$3D
       FA78 00                FCB   $00,$A0,$00,$00,$00,$00,$00,$00
                      *
                      * KEYBOARD 2 TABLE DEFINITION
                      *
       FA80           ADTAB2  EQU   *
       FA80 24                FCB   $24,$F2,$0B,$F7,$0C,$00,$0A,$F6
       FA88 00                FCB   $00,$00,$00,$F5,$08,$00,$00,$00
       FA90 5C                FCB   $5C,$80,$00,$00,$00,$F4,$3D,$00
       FA98 3C                FCB   $3C,$00,$31,$34,$30,$F3,$2E,$37
       FAA0 25                FCB   $25,$3E,$2D,$2A,$2B,$00,$00,$2F
       FAA8 5B                FCB   $5B,$2F,$00,$F9,$1D,$00,$00,$F8
       FAB0 2A                FCB   $2A,$5D,$33,$36,$00,$00,$00,$39
       FAB8 23                FCB   $23,$1B,$32,$35,$00,$00,$2C,$38
```

```
                    * 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 E8 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
```

```
FB1D 5A                      DEC B
FB1E 26 EF                   BNE    STAB1
FB20 39                      RTS
```



```
                     *
                     * GET THE CHARACTER TYPED ON KEYBOARD
                     *
FB21 B6 E4 C2        INPUT   LDA A  SAVEA     GET CHARACTER
FB24 7F E4 13                CLR    FLGIN     CLEAR FLAG
FB27 39                      RTS              RETURN
                     *
                     * REPEAT PROCESSING
                     *
FB28 C6 47           REPTA   LDA B  £$47      READ FIFO SENSOR RAM AT ROW 7
FB2A F7 E8 21                STA B  CMDE1
FB2D CE 03 00                LDX    £$300     DELAY AFTER COMMAND
FB30 09              REPTA4  DEX
FB31 26 FD                   BNE    REPTA4
FB33 F6 E8 20                LDA B  DATA1     GET DATA
FB36 C4 02                   AND B  £2        CHECK FOR REPEAT CODE
FB38 27 0B                   BEQ    REPTA2    NO MORE REPEAT
FB3A CE 20 00                LDX    £$2000    DELAY
FB3D 09              REPTA1  DEX
FB3E 26 FD                   BNE    REPTA1
FB40 73 E4 13                COM    FLGIN     SET INPUT FLAG
FB43 20 08                   BRA    REPTA3
FB45 C6 02           REPTA2  LDA B  £2        SET ENCODED SCAN KEYBOARD
FB47 F7 E8 21                STA B  CMDE1
FB4A 7F E4 12                CLR    FLGRPT    CLEAR REPEAT FLAG
FB4D C6 C1           REPTA3  LDA B  £$C1      CLEAR INTEL 8279
FB4F F7 E8 21                STA B  CMDE1
FB52 20 11                   BRA    TINPT1    RETURN
                     *
                     * CHECK IF A CHARACTER HAS BEEN TYPED
                     *
FB54 7D E4 13        TINPUT  TST    FLGIN     CHARACTER ALREADY PRESENT
FB57 26 14                   BNE    TINPT2    YES RETURN
FB59 FF E4 C3                STX    SAVEX     SAVE X
FB5C 36                      PSH A            SAVE A
FB5D 37                      PSH B            SAVE B
FB5E 7D E4 12                TST    FLGRPT    REPEAT ?
FB61 26 C5                   BNE    REPTA     YES GO PROCESS REPEAT
FB63 8D 09                   BSR    KEYB      CHECK AND READ KEYBOARD
FB65 FE E4 C3        TINPT1  LDX    SAVEX     RESTORE X
FB68 7D E4 13                TST    FLGIN     SET Z
FB6B 33                      PUL B            RESTORE B
FB6C 32                      PUL A            RESTORE A
FB6D 39              TINPT2  RTS              RETURN
                     *
                     * CHECK AND READ KEYBOARD
                     *
FB6E B6 E8 21        KEYB    LDA A  CMDE1     CHECK KEYBOARD 1
FB71 84 0F                   AND A  £$F
FB73 26 11                   BNE    KEYB1     WE HAVE CHAR.
FB75 B6 E8 31                LDA A  CMDE2     CHECK KEYBOARD 2
FB78 84 0F                   AND A  £$F
FB7A 27 38                   BEQ    KEYB4     NOTHING, RETURN
```

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

FB7C B6 E8 30           LDA A  DATA2      GET CHAR.
FB7F CE FA 80           LDX    £ADTAB2    ADDRESS OF TABLE 2
FB82 84 3F              AND A  £$3F       ONLY 6 BITS
FB84 20 0D              BRA    KEYB2
FB86 B6 E8 20  KEYB1    LDA A  DATA1      GET CHAR.
FB89 CE F9 80           LDX    £ADTAB1    ADDRESS OF TABLE 1
FB8C 7D E4 11           TST    FLGSHF     CHECK SHIFT LOCK
FB8F 26 02              BNE    KEYB2
FB91 84 BF              AND A  £$BF       IF LOCKED,MASK THE BIT
FB93 FF E4 C0  KEYB2    STX    PTTAB
FB96 BB E4 C1           ADD A  PTTAB+1    ADDRESS OF CHAR. IN THE TABLE
FB99 24 03              BCC    KEYB3
FB9B 7C E4 C0           INC    PTTAB      ADD CARRY IF ANY
FB9E B7 E4 C1  KEYB3    STA A  PTTAB+1
FBA1 FE E4 C0           LDX    PTTAB      GET ADDRESS OF CHAR.
FBA4 A6 00              LDA A  0,X        GET CODE
FBA6 B7 E4 C2           STA A  SAVEA      SAVE IT
FBA9 81 A1              CMP A  £SHIFT     SHIFT KEY ?
FBAB 27 08              BEQ    GSHIFT     YES GO PROCESS IT
FBAD 81 A0              CMP A  £REPT      REPEAT KEY ?
FBAF 27 15              BEQ    GREPT      IF YES GO PROCESS IT
FBB1 73 E4 13           COM    FLGIN      SET INPUT FLAG
FBB4 39        KEYB4    RTS               RETURN
               *
               * SHIFT LOCK PROCESSING
               *
FBB5 7D E4 11  GSHIFT   TST    FLGSHF     CHECK FLAG
FBB8 27 03              BEQ    GSHFT1     SKIP IF LOCKED
FBBA 86 A0              LDA A  £$A0       LIGHT ON
FBBC 8C                 FCB    SKIP2
FBBD 86 A3     GSHFT1   LDA A  £$A3       LIGHT OFF
FBBF B7 E8 21           STA A  CMDE1
FBC2 73 E4 11           COM    FLGSHF     SWITCH THE FLAG
FBC5 39                 RTS               RETURN
               *
               * FIRST TIME REPEAT HAS BEEN TYPED
               *
FBC6 8D A6     GREPT    BSR    KEYB       CHECK AND READ KEYBOARD
FBC8 7D E4 13           TST    FLGIN      CHAR. TYPED ?
FBCB 27 F9              BEQ    GREPT      NO,WAIT
FBCD C6 04              LDA B  £4         YES SET ENCODED SCAN SENSOR MATRIX
FBCF F7 E8 21           STA B  CMDE1
FBD2 86 FF              LDA A  £$FF       SET REPEAT FLAG
FBD4 B7 E4 12           STA A  FLGRPT
FBD7 39                 RTS               RETURN
               *
               * RAM RESERVED TO VARIABLE INFORMATIONS
               *
E411                    ORG    $E411      INITIALIZED TO ZERO AT START
E411           FLGSHF   RMB    1
E412           FLGRPT   RMB    1
E413           FLGIN    RMB    1
E4C0                    ORG    $E4C0
E4C0           PTTAB    RMB    2

KEYBOARD HANDLER                    22-10-81  TSC ASSEMBLER  PAGE   12

E4C2           SAVEA    RMB    1
E4C3           SAVEX    RMB    2
E4C5           VTEMP    RMB    2

                        END
```

```
      FBD8                      ORG    $FBD8
                                NAM    DISK BOOTSTRAP
                                OPT    PAG

                         ********************************
                         *                              *
                         * -S- COMMAND                  *
                         *   DISK BOOTSTRAP (5")        *
                         *                              *
                         ********************************

                         * EQUATES FOR WD 1791

      E8E0               DRVREG EQU    $E8E0      DRIVE REGISTER
      E8F0               COMREG EQU    $E8F0      COMMAND REGISTER
      E8F2               SECREG EQU    $E8F2      SECTOR REGISTER
      E8F3               DATREG EQU    $E8F3      DATA REGISTER
      A100               LOADER EQU    $A100

                         * PROGRAM STARTS HERE

      FBD8 B6 E8 F0  DKBOOT LDA A  COMREG     TURN MOTOR ON
      FBDB 7F E8 E0         CLR    DRVREG     SELECT DRIVE £0
      FBDE CE 00 00         LDX    £0
      FBE1 08        OVR    INX               DELAY FOR MOTOR SPEEDUP
      FBE2 09               DEX
      FBE3 09               DEX
      FBE4 26 FB            BNE    OVR
      FBE6 C6 F4            LDA B  £$F4       DO RESTORE COMMAND
      FBE8 F7 E8 F0         STA B  COMREG
      FBEB 8D 2F            BSR    DELAY
      FBED F6 E8 F0  LOOP1  LDA B  COMREG     CHECK WD STATUS
      FBF0 53               COM B
      FBF1 C5 01            BIT B  £1         WAIT TILL NOT BUSY
      FBF3 26 FB            BNE    LOOP1
      FBF5 86 FE            LDA A  £$FE       SETUP FOR SECTOR £1
      FBF7 B7 E8 F2         STA A  SECREG
      FBFA 8D 20            BSR    DELAY
      FBFC 86 73            LDA A  £$73       SETUP READ COMMAND
      FBFE B7 E8 F0         STA A  COMREG
      FC01 8D 19            BSR    DELAY
      FC03 CE A1 00         LDX    £LOADER    ADDRESS OF LOADER
      FC06 C5 02     LOOP2  BIT B  £2         DTA PRESENT ?
      FC08 27 07            BEQ    LOOP3      SKIP IF NOT
      FC0A B6 E8 F3         LDA A  DATREG     GET A BYTE
      FC0D 43               COM A
      FC0E A7 00            STA A  0,X        PUT IN MEMORY
      FC10 08               INX               BUMP POINTER
      FC11 F6 E8 F0  LOOP3  LDA B  COMREG     CHECK WD STATUS
      FC14 53               COM B
      FC15 C5 01            BIT B  £1         IS WD BUSY ?
      FC17 26 ED            BNE    LOOP2      LOOP IF SO
      FC19 7E A1 00         JMP    LOADER     JUMP TO FLEX LOADER
```



```

      FC1C 8D 00     DELAY  BSR    DEL1
      FC1E 8D 00     DEL1   BSR    RTN
      FC20 39        RTN    RTS

                            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    $B        VIA AUXILIARY CONTROL REGISTER
      0004            TILL    EQU    4         VIA T1 LOW ORDER LATCH
                      *
                      * INITIALIZE ACIA (CPU)
                      *    A:=ACIA COMMAND
                      *    X:=CLOCK VIA
                      *
      FC25 C6 03      INICIA  LDA B  £3        MASTER RESET CODE
      FC27 F7 E8 0C           STA B  ACIAC     RESET ACIA
      FC2A B7 E8 0C           STA A  ACIAC     INITIALIZE ACIA
      FC2D 86 C0              LDA A  £$C0      INZ CLOCK FOR ACIA
      FC2F B7 E8 1B           STA A  VIAC+ACR
      FC32 FF E8 14           STX    VIAC+TILL 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 08      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

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

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

```
                ********************************
                *                              *
                *    PARALLEL PRINTER HANDLER  *
                *                              *
                ********************************


                *
                * OUTPUT ONE CHARACTER TO PRINTER
                *
FC8F 37         POUT   PSH B          SAVE B
FC90 F6 E8 4D   CPOUT  LDA B  VIACL+IFR WAIT FOR READY
FC93 C5 10             BIT B  £$10
FC95 27 F9             BEQ    CPOUT
FC97 B7 E8 40          STA A  VIACL+ORB OUTPUT CHAR.
FC9A C6 C3             LDA B  £$C3
FC9C F7 E8 4C          STA B  VIACL+PCR HANDSHAKE
FC9F C6 E3             LDA B  £$E3
FCA1 F7 E8 4C          STA B  VIACL+PCR
FCA4 33                PUL B          RESTORE B
FCA5 39                RTS            RETURN
```

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

```
                ***********************
                *                     *
                *      MONITOR        *
                *                     *
                ***********************


                *
                * EQUATES
                *
FCA6            ROM    EQU   $FCA6
E400            RAM    EQU   $E400
003F            SWI    EQU   $3F      SWI OP CODE
0030            SZRCL  EQU   SP-NIO-2 SIZE OF RAM TO CLEAR
000D            CR     EQU   $D       CARRIAGE RETURN
000A            LF     EQU   $A       LINE FEED
002B            PRPT   EQU   '+       PROMPT CHARACTER
002E            DOWN   EQU   '.       MEMORY INCREMENT CHAR.
002D            UP     EQU   '-       MEMORY DECREMENT CHAR.
                *
EC00            K7MOD  EQU   $EC00    K7/MODEM HANDLER ADDRESS
8000            BASIC  EQU   $8000    ROM BASIC EXECUTION ADDRESS
FB54            CHKCHR EQU   TINPUT
```

```
FCA6                  ORG    ROM
                *
                * JUMP TABLE TO ROUTINES PERFORMING GPMON FUNCTIONS
                *
FCA6            FCTABL EQU    *
FCA6 4D                FCC    /M/ -M- MEMORY CHANGE
FCA7 FD ED             FDB    CHANGE
FCA9 47                FCC    /G/ -G- GOTO ENTERED ADDRESS
FCAA FE 37             FDB    GOTO
FCAC 52                FCC    /R/ -R- PRINT STACK
FCAD FF 1E             FDB    PSTAK
FCAF 56                FCC    /V/ -V- SET A BREAKPOINT
FCB0 FE 2A             FDB    SETBRK
FCB2 55                FCC    /U/ -U- RESET A BREAKPOINT
FCB3 FE 1E             FDB    RSTBRK
FCB5 57                FCC    /W/ -W- DELETE ALL BREAKPOINTS
FCB6 FE 26             FDB    DELBRK
FCB8 43                FCC    /C/ -C- CONTINUE
FCB9 FE 43             FDB    CONT
FCBB 4E                FCC    /N/ -N- NEXT (TRACE 1 INSTRUCTION)
FCBC FE 4B             FDB    NEXT
FCBE 54                FCC    /T/ -T- TRACE N INSTRUCTIONS
FCBF FE 4F             FDB    TRACE
FCC1 42                FCC    /B/ -B- PRINT ALL BREAKS
FCC2 FE 23             FDB    PNTBRK
FCC4 44                FCC    /D/ -D- MEMORY DUMP
FCC5 FE 65             FDB    DUMP
FCC7 F3                FCB    $F3        -BOOT- EXECUTE DISK BOOTSTRAP
FCC8 FB DB             FDB    DKBOOT
FCCA 4B                FCC    /K/ -K- LINK TO K7/MODEM HANDLER
FCCB EC 00             FDB    K7MOD
FCCD 4F                FCC    /O/ -O- SWITCH OUTPUT TO PRINTER
FCCE FE 8C             FDB    SWITCH
FCD0 F1                FCB    $F1        -BASIC- EXECUTE ROM BASIC
FCD1 80 00             FDB    BASIC
FCD3 F2                FCB    $F2        -COM- TERMINAL MODE
FCD4 FC 53             FDB    COM
FCD6 F4                FCB    $F4        -GR- GRAPHIC MODE
FCD7 FA C0             FDB    TEXT
FCD9            FCTBEN EQU    *
```

```
                   *
                   * 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 B8                 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
```

```
                      *
                      * MAIN COMMAND/CONTROL LOOP
                      *
FD12                  CONTRL  EQU    *
FD12 BE E4 34                 LDS    SP        RESTORE STACK POINTER
FD15 CE FF 06                 LDX    £SWI1S    RESTORE SWI VECTOR
FD18 FF E4 36                 STX    SWI1
FD1B BD FD D4                 JSR    PROMPT    PRINT PROMPT
FD1E 7F E4 08                 CLR    OUTSW     MAKE SURE INPUT IS ECHOED
FD21 BD FD 98                 JSR    INCH      READ COMMAND CHARACTER
FD24 16                       TAB              SAVE IT IN B
FD25 BD FD B3                 JSR    OUTS      PRINT SPACE AFTER COMMAND
                      *
                      * B REGISTER HOLDS CHARACTER INPUT BY USER
                      * USE JUMP TABLE TO GO TO APPROPRIATE ROUTINE
                      *
FD28 CE FC A6                 LDX    £FCTABL   X:= ADDRESS OF JUMP TABLE
FD2B E1 00            NXTCHR  CMP B  0,X       DOES INPUT CHAR. MATCH ?
FD2D 27 0F                    BEQ    GOODCH    YES,GO TO APPROPRIATE ROUTINE
FD2F 08                       INX              NO UPDATE INDEX
FD30 08                       INX
FD31 08                       INX
FD32 8C FC D9                 CPX    £FCTBEN   END OF TABLE REACHED ?
FD35 26 F4                    BNE    NXTCHR    NO,TRY NEXT COMMAND
FD37 86 3F            WHAT    LDA A  £'?       NO MATCH PRINT "?"
FD39 BD FD 82                 JSR    OUTCH
FD3C 20 D4                    BRA    CONTRL    REPROMPT USER
                      *
FD3E EE 01            GOODCH  LDX    1,X       GET ADDRESS FROM J.T.
FD40 6E 00                    JMP    0,X       GO TO APPROPRIATE ROUTINE
```

```
FD42                  ORG    $FD42
                *
                * BASIC ROUTINES
                *

                *
                * INPUT HEX CHARACTER
                *
FD42 8D 54      INHEX   BSR    INCH
FD44 80 30      INHEX2  SUB A  £$30
FD46 2B CA              BMI    CONTRL    NOT HEX
FD48 81 09              CMP A  £$9
FD4A 2F 0A              BLE    IN1HG
FD4C 81 11              CMP A  £$11
FD4E 2B C2              BMI    CONTRL    NOT HEX
FD50 81 16              CMP A  £$16
FD52 2E BE              BGT    CONTRL    NOT HEX
FD54 80 07              SUB A  £7
FD56 39         IN1HG   RTS
                *
                * BUILD ADDRESS
                *
FD57 8D 10      BADDR   BSR    BYTE      READ 2 CHAR.
FD59 B7 E4 0A           STA A  XHI
FD5C 8D 0B              BSR    BYTE      READ 2 CHAR.
FD5E B7 E4 0B           STA A  XLOW
FD61 FE E4 0A           LDX    XHI       X:= ADDRESS WE BUILT
FD64 39                 RTS
FD65 8D F0      BADDRS  BSR    BADDR     BUILD ADDRESS
FD67 20 4A              BRA    OUTS      PRINT SPACE AND RETURN
                *
                * INPUT BYTE
                *
FD69 8D D7      BYTE    BSR    INHEX     GET HEX CHAR.
FD6B 48         BYTE2   ASL A
FD6C 48                 ASL A
FD6D 48                 ASL A
FD6E 48                 ASL A
FD6F 16                 TAB
FD70 8D D0              BSR    INHEX     GET HEX CHAR.
FD72 1B                 ABA
FD73 39                 RTS
                *
                * OUT HEXA BCD DIGIT
                *
FD74 44         OUTHL   LSR A            OUT HEX LEFT BCD DIGIT
FD75 44                 LSR A
FD76 44                 LSR A
FD77 44                 LSR A
FD78 84 0F      OUTHR   AND A  £$F       OUT HEX RIGHT BCD DIGIT
FD7A 8B 30              ADD A  £$30
FD7C 81 39              CMP A  £$39
FD7E 23 02              BLS    OUTCH
```

```
FD80 8B 07              ADD A  £7
                 *
                 * OUTPUT ONE CHARACTER
                 *
FD82 BD F8 EF    OUTCH  JSR    OUTPUT    OUTPUT CHAR. TO SCREEN
FD85 7D E4 09           TST    PRTFLG    TO PRINTER ?
FD88 26 01              BNE    POUTS     IF YES PRINT CHAR.
FD8A 39                 RTS              RETURN
FD8B 7E FC 8F    POUTS  JMP    POUT
                 *
                 * PRINT DATA POINTED BY X-REG
                 *
FD8E 8D F2       PDATA2 BSR    OUTCH     OUTPUT DATA
FD90 08          PDATA3 INX
FD91 A6 00       PDATA1 LDA A  X
FD93 81 04              CMP A  £4        STOP IF EOT
FD95 26 F7              BNE    PDATA2
FD97 39                 RTS              RETURN
                 *
                 * INPUT ONE CHARACTER
                 *
FD98 BD FB 54    INCH   JSR    TINPUT    CHECK FOR TYPED CHAR
FD9B 27 FB              BEQ    INCH      NOT READY
FD9D BD FB 21           JSR    INPUT     INPUT CHAR.
FDA0 7D E4 08           TST    OUTSW     SHOULD INPUT BE ECHOED?
FDA3 27 DD              BEQ    OUTCH     IF SO ,OUTPUT CHAR.
FDA5 39                 RTS
                 *
                 * OUTPUT BYTE
                 *
FDA6 A6 00       OUT2H  LDA A  0,X       OUTPUT 2 HEX CHAR.
FDA8 08                 INX
FDA9 36          OUT2HA PSH A
FDAA 8D C8              BSR    OUTHL     OUT LEFT HEX CHAR.
FDAC 32                 PUL A
FDAD 20 C9              BRA    OUTHR     OUTPUT RIGHT HEX CHAR AND RTS
                 *
                 * OUTPUT ADDRESS
                 *
FDAF 8D F5       OUT4HS BSR    OUT2H     OUTPUT 4 HEX CHAR.+SPACE
FDB1 8D F3       OUT2HS BSR    OUT2H     OUTPUT 2 HEX CHAR.+SPACE
FDB3 86 20       OUTS   LDA A  £$20      OUTPUT SPACE
FDB5 20 CB              BRA    OUTCH     (BSR+RTS)
                 *
                 * PRINT STACK CONTENTS
                 *
FDB7 8D 13       PRINT  BSR    PCRLF     OUTPUT CR/LF
FDB9 FE E4 34           LDX    SP        PRINT OUT STACK
FDBC 08                 INX
FDBD 8D F2              BSR    OUT2HS    CC-REG
FDBF 8D F0              BSR    OUT2HS    ACC-B
FDC1 8D EE              BSR    OUT2HS    ACC-A
FDC3 8D EA              BSR    OUT4HS    X-REG
FDC5 8D E8              BSR    OUT4HS    P-COUNTER
```

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

FDC7 CE E4 34             LDX    £SP
FDCA 20 E3                BRA    OUT4HS    PRINT SP AND RETURN
                   *
                   * OUTPUT CR/LF
                   *
FDCC 86 0D         PCRLF  LDA A  £CR       OUTPUT CR
FDCE 8D B2                BSR    OUTCH
FDD0 86 0A                LDA A  £LF       OUTPUT LF
FDD2 20 AE                BRA    OUTCH     AND RETURN
                   *
                   * OUTPUT PROMPT
                   *
FDD4 8D F6         PROMPT BSR    PCRLF     DO CR/LF
FDD6 86 2B                LDA A  £PRPT
FDD8 20 A8                BRA    OUTCH     OUTPUT PROMPT AND RETURN
                   *
                   * ERROR : VALUE IN RAM DO NOT MATCH
                   *
FDDA 16            ERROR  TAB              SAVE A
FDDB 8D EF                BSR    PCRLF
FDDD CE E4 0A             LDX    £XHI
FDE0 8D CD                BSR    OUT4HS    PRINT ADDRESS
FDE2 FE E4 0A             LDX    XHI
FDE5 8D CA                BSR    OUT2HS    PRINT CONTENT
FDE7 17                   TBA              RESTORE A
FDE8 8D BF                BSR    OUT2HA    PRINT VALUE WE WANT TO STORE
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
```

```
FE2E 8D 63          BSR    BRKSUB
FE30 86 05          LDA A  £BKFRCD    SET ONE BREAKPOINT FLAG
FE32 8D 5F   SETB2  BSR    BRKSUB     BREAK HANDLING SUBROUTINE
FE34 7E FD 12 CTRLJ JMP    CONTRL     RETURN TO COMMAND LEVEL
             *
             * -G- COMMAND
             *   GO TO ENTERED ADDRESS
             *
FE37 8D 29   GOTO   BSR    BADDRJ     GET ADDR. FROM USER IN XHI,XLOW
FE39 FE E4 34       LDX    SP         GET VALUE SAVED WHEN ENTERING GPMON
FE3C A7 07          STA A  7,X        A-REG CONTAINS XLOW
FE3E B6 E4 0A       LDA A  XHI
FE41 A7 06          STA A  6,X        PLACE ADDRESS ON STACK
             *
             * -C- COMMAND
             *   CONTINUE EXECUTION
             *
FE43 7C E4 10 CONT  INC    BRKTRC     TRACE 1 TO RESTORE SWI'S
FE46 CE 00 01       LDX    £1
FE49 20 09          BRA    TRACE3
             *
             * -N- COMMAND
             *   SINGLE INSTRUCTION TRACE REQUESTED
             *
FE4B CE 00 01 NEXT  LDX    £1         £INSTRUCTION TO TRACE
FE4E 8C             FCB    SKIP2      SKIP 2 BYTES
             *
             * -T- COMMAND
             *   MULTIPLE INSTRUCTION TRACE
             *
FE4F 8D 11   TRACE  BSR    BADDRJ     GET N. OF INSTRUCTION TO TRACE
FE51 7F E4 10 TRACE1 CLR   BRKTRC     CLEAR FLAG INDICATING TRACE IS DUE TO BREA
FE54 FF E4 06 TRACE3 STX   NTRACE     SAVE £ INST'S TO TRACE
FE57 FE E4 34       LDX    SP         X:=STACK POINTER
FE5A EE 06          LDX    6,X        X:=ADDR OF INSTR TO BE EXECUTED
FE5C BD FF 9A       JSR    SAVTRC     SAVE ADDR/OPCODE FOR TRACE
FE5F 7E FF 47       JMP    CONTRC     GOTO CONTINUE TRACE
FE62 7E FD 57 BADDRJ JMP   BADDR
             *
             * -D- COMMAND
             *   MEMORY DUMP
             *
FE65 BD FD 65 DUMP  JSR    BADDRS     GET START ADDRESS
FE68 FF E4 0C       STX    XSAVE      SAVE IT
FE6B 8D F5          BSR    BADDRJ     GET END ADDRESS IN XHI,XLOW
FE6D BD FD CC DUMP1 JSR    PCRLF      OUTPUT CR,LF
FE70 C6 10          LDA B  £16        16 BYTES PER LINE
FE72 CE E4 0C       LDX    £XSAVE     PRINT ADDRESS
FE75 BD FD AF       JSR    OUT4HS
FE78 FE E4 0C DUMP2 LDX    XSAVE      PRINT BYTE
FE7B BD FD B1       JSR    OUT2HS
FE7E FF E4 0C       STX    XSAVE      SAVE NEW ADDRESS
FE81 09             DEX
FE82 BC E4 0A       CPX    XHI        FINISHED ?
```

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

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

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

                  *
                  * 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
 0082             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 02               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
```

```
GPMON V1.3                          22-10-81  TSC ASSEMBLER  PAGE   29

FEB7 24 0D              BCC    BRKP3
FEB9 A6 00              LDA A  X          SAVE USER CODE
FEBB 36                 PSH A
FEBC 86 3F              LDA A  £SWI
FEBE A7 00              STA A  X          WRITE SWI
FEC0 32                 PUL A
FEC1 FE E4 0C           LDX    XSAVE      ADDR OF BREAK IN BREAK TABLE
FEC4 A7 02              STA A  2,X        SAVE USER CODE IN BREAK TABLE
                *
                * RECORD BREAKPOINT IN TABLE
                *
FEC6 FE E4 0C   BRKP3   LDX    XSAVE      GET ADDRESS OF BREAK IN TABLE
FEC9 54                 LSR B             NEXT FUNCTION
FECA 24 0E              BCC    BRKP4
FECC B6 E4 0A           LDA A  XHI        GET HI-BYTE OF NEW ADDRESS
FECF A7 00              STA A  0,X        WRITE IN TABLE
FED1 B6 E4 0B           LDA A  XLOW       GET LOW BYTE
FED4 A7 01              STA A  1,X
FED6 32                 PUL A             GET FUNCTION NUMBER FROM STACK
FED7 84 FB              AND A  £%11111011 DO NOT PLACE ADDRESS MORE THAN ONCE
FED9 36                 PSH A             CONTINUE TO TAKE OUT BREAKPOINTS
                *
                * PURGE BREAKPOINT WHOSE ADDRESS IS IN XHI,XLOW
                *
FEDA 54         BRKP4   LSR B
FEDB 24 0E              BCC    BRKP5
FEDD A6 00              LDA A  X          GET HI-BYTE FROM TABLE
FEDF B1 E4 0A           CMP A  XHI        COMPARE
FEE2 26 07              BNE    BRKP5      NOT THIS BREAK TO BE PURGED
FEE4 A6 01              LDA A  1,X        GET LOW BYTE
FEE6 B1 E4 0B           CMP A  XLOW
FEE9 27 03              BEQ    BRKP52     THIS BREAK IS TO BE PURGED
                *
                * PURGE ALL BREAKPOINTS
                *
FEEB 54         BRKP5   LSR B
FEEC 24 04              BCC    BRKP6
FEEE 6F 00      BRKP52  CLR    0,X        CLEAR BREAKPOINT HIGH
FEF0 6F 01              CLR    1,X        CLEAR LOW BYTE
                *
                * PRINT BREAKPOINT
                *
FEF2 54         BRKP6   LSR B
FEF3 24 04              BCC    BRKPE
FEF5 BD FD AF           JSR    OUT4HS     PRINT BREAKPOINT ADDRESS
FEF8 8C                 FCB    SKIP2      SKIP 2 BYTES
                *
                * UPDATE LOOP INDEX AND LOOP IF APPROPRIATE
                *
FEF9 08         BRKPE   INX
FEFA 08                 INX
FEFB 08         BRKPE1  INX               X POINTS TO NEXT ADDR IN TABLE
FEFC 8C E4 34           CPX    £BRKINS
FEFF 26 96              BNE    BRKP0

GPMON V1.3                          22-10-81  TSC ASSEMBLER  PAGE   30

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

```
                    *
                    * SWI SOFTWARE INTERRUPT PROCESSING
                    *
FF06                SWI1S   EQU    *
FF06 BF E4 34               STS    SP          SAVE USER'S SP
FF09 86 01                  LDA A  £BKFRMV
FF0B BD FE 93               JSR    BRKSUB      GO TAKE OUT ALL BREAKS
                    * DECREMENT P-COUNTER
FF0E 30                     TSX                X:=STACK POINTER-1
FF0F 6D 06                  TST    6,X         IF LOWER BYTE=0 : BORROW
FF11 26 02                  BNE    SWI1S1      BRANCH IF BORROW NOT REQUIRED
FF13 6A 05                  DEC    5,X         DECREMENT UPPER BYTE
FF15 6A 06          SWI1S1  DEC    6,X         DECREMENT LOWER BYTE
                    * TEST FOR ADDRESS TRACE OR BREAK
FF17 EE 05                  LDX    5,X         X:=P-COUNTER
FF19 BC E4 04               CPX    TRCADR      IS SWI FOR TRACE ?
FF1C 27 06                  BEQ    TRCINH      YES,GO TO TRACE INT HANDLER
                    *
                    * BREAK INTERRUPT HANDLER
                    *
                    *
                    * -R- COMMAND
                    *
FF1E                PSTAK   EQU    *
FF1E BD FD B7               JSR    PRINT       STOP AND SHOW REGISTER TO USER
FF21 7E FD 12       CNTRL3  JMP    CONTRL      RETURN
                    *
                    * TRACE INTERRUPT HANDLER
                    * P-COUNTER HAS BEEN DECREMENTED TO POINT AT SWI
                    * TRCINS HOLDS OP CODE REPLACED BY SWI
                    * X HOLDS ADDRESS OF WHERE TRACE SWI IS
                    *
FF24 B6 E4 0E       TRCINH  LDA A  TRCINS      GET OPCODE OF TRACED INSTR.
FF27 A7 00                  STA A  0,X         RESTORE TO USER'CODE
FF29 7D E4 10               TST    BRKTRC      IS PROCESSING TO BE IMMEDIATELY CONTINUED ?
FF2C 27 0D                  BEQ    NBKTRC      BRANCH IF NOT
                    *
                    * PROCESSING IS TO "CONTINUE"
                    *
FF2E 7F E4 10               CLR    BRKTRC      RESET CONTINUE FLAG
FF31 BD FE 91               JSR    INSBRK      INSERT BREAKS IN USER CODE
FF34 7F E4 04               CLR    TRCADR      NO MORE TRACE,SO CLEAR ADDRESS
FF37 7F E4 05               CLR    TRCADR+1
FF3A 3B                     RTI                CONTINUE
                    *
                    * TRACE IS DUE TO N OR T TRACE COMMAND
                    *
FF3B BD FD B7       NBKTRC  JSR    PRINT       PRINT STACK
FF3E FE E4 06               LDX    NTRACE      GET £ OF INSTRUCTIONS TO TRACE
FF41 09                     DEX                DECREMENT COUNT
FF42 FF E4 06               STX    NTRACE      AND RESTORE
FF45 27 DA                  BEQ    CNTRL3      BRANCH IF ALL TRACES DONE
                    *
```

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

                         * TRACE NOT DONE - TRACE NEXT INSTRUCTION
                         *
 FF47 B6 E4 0E   CONTRC  LDA A  TRCINS     GET CURRENT INSTRUCTION
 FF4A B7 E4 38           STA A  BRINS      SAVE IN CASE IT'S A BRANCH
 FF4D 8D 6F              BSR    OPCBYT     GO GET £ OF BYTES/TYPE
 FF4F 4D                 TST A             CHECK FOR BRANCH
 FF50 2A 0A              BPL    CKOBRA     CHECK FOR OTHER THAN BRANCH
                         *
                         * RELATIVE BRANCH TYPE INSTRUCTION
                         * DETERMINE WHERE TO PUT SWI
                         * S HOLDS POINTER TO USER STACK AFTER SWI
                         *
 FF52 32                 PUL A             GET CONDITION CODE
 FF53 34                 DES               UPDATE STACK PTR AFTER PULL
 FF54 9A 10              ORA A  %00010000 MAKE INT'S INHIBITED
 FF56 06                 TAP               RESTORE USER'S C CODE REG
 FF57 7E E4 38           JMP    BRINS      GO SEE HOW RELATIVE BRANCH FARES
                         *
                         * BRANCH WAS NOGO - PUT SWI AT NEXT INSTRUCTION
                         *
 FF5A 86 02      BRNOGO  LDA A  £2         A:=£ OF BYTES AFTER CURRENT INSTR.
                         *
                         * INSTRUCTION TO BE TRACED IS NOT A BRANCH
                         *
 FF5C FE E4 04   CKOBRA  LDX    TRCADR     X:=TRACE ADDRESS
 FF5F E6 00              LDA B  0,X        GET INSTR. TO BE TRACED
 FF61 C1 6E              CMP B  £$6E       IS IT A JUMP,INDEXED
 FF63 27 44              BEQ    JMPIDX     YES GO SIMULATE JUMP IDXED
 FF65 C1 7E              CMP B  £$7E       JUMP EXTENDED ?
 FF67 27 51              BEQ    JMPEXT
 FF69 C1 AD              CMP B  £$AD       JSR,INDEXED ?
 FF6B 27 3C              BEQ    JMPIDX     JUMP IDXED IS SAME AS TRANSFERT OF CONTROL
 FF6D C1 BD              CMP B  £$BD       JSR,EXTENDED ?
 FF6F 27 49              BEQ    JMPEXT
 FF71 C1 3B              CMP B  £$3B       RTI ?
 FF73 27 3B              BEQ    RTISIM
 FF75 C1 39              CMP B  £$39       RTS ?
 FF77 27 3C              BEQ    RTSSIM
 FF79 C1 3F              CMP B  £$3F       SWI ? IF SWI USER DON'T INCREMENT ADDR.
 FF7B 27 13              BEQ    BRG2
 FF7D C1 8D              CMP B  £$8D       BSR ? IF YES BRANCH PROCESSING
                         *
                         * NOT A BRANCH,JUMP,RTI,RTS
                         * A REGISTER HOLDS £ OF BYTES IN INSTRUCTION
                         *
 FF7F 26 0D              BNE    BRG1       PUT IN NEW SWI AND TRACE NEXT INSTR.
                         *
                         * BRANCH WAS GO,PUT SWI AT ADDRESS BEING JUMP TO
                         *
 FF81 FE E4 04   BRGO    LDX    TRCADR     X:=TRACE ADDRESS
 FF84 A6 01              LDA A  1,X        GET BRANCH OFFSET
 FF86 08                 INX               OFFSET IS RELATIVE TO
 FF87 08                 INX               INSTR FOLLOWING BRANCH
 FF88 2A 04              BPL    BRG1       BRANCH IS OFFSET POSITIVE
```

```
                      * X NEEDS TO BE DECREMENTED (OFFSET NEGATIVE)
FF8A 09         BRGODC  DEX                 DECREMENT ADDRESS
FF8B 4C                 INC A               INCREMENT COUNTER
FF8C 26 FC              BNE     BRGODC
                      * CONTENT OF A IS 0
FF8E 8D 15      BRG1    BSR     INCX        INCREMENT X BY AMOUNT IN A-REG
FF90 8D 08      BRG2    BSR     SAVTRC      SAVE ADDR/OPC OF NEXT INSTR TO STOP ON
FF92 86 3F              LDA A   £SWI        REPLACE INSTR. WITH SWI
FF94 A7 00              STA A   X
FF96 BE E4 34   RTIPC   LDS     SP          GET ORIGINAL STACK POINTER
FF99 3B                 RTI                 TRACE ANOTHER INSTR.
                      *
                      * SUBROUTINE TO SAVE ADDR IN X INTO TRCADR AND OPC INTO TRCINS
                      *
FF9A            SAVTRC  EQU     *
FF9A FF E4 04           STX     TRCADR
FF9D A6 00              LDA A   X
FF9F B7 E4 0E           STA A   TRCINS
FFA2 39                 RTS
                      *
                      * SUBROUTINE TO INCREMENT X BY CONTENT OF A
                      *
FFA3 08         INXLP   INX                 INCREMENT X
FFA4 4A                 DEC A               DECREMENT COUNT
FFA5 4D         INCX    TST A
FFA6 26 FB              BNE     INXLP       IF COUNT NOT YET 0 LOOP
FFA8 39                 RTS                 RETURN
                      *
                      * JUMP,JSR INDEXED SIMULATION
                      *
FFA9 A6 01      JMPIDX  LDA A   1,X         A:=ADDR OFFSET
FFAB 30                 TSX
FFAC EE 03              LDX     3,X         GET TARGET'S X REG
FFAE 20 DE              BRA     BRG1        UPDATEX TRACE NEXT INSTR
                      *
                      * RTI ENCOUNTERED
                      *
FFB0 30         RTISIM  TSX
FFB1 EE 0C              LDX     12,X        GET P-COUNTER FROM STACK
FFB3 20 DB              BRA     BRG2        GO TRACE NEXT INSTR.
                      *
                      * RTS ENCOUNTERED
                      *
FFB5 30         RTSSIM  TSX
FFB6 EE 07              LDX     7,X         GET RETURN P-REG FROM STACK
FFB8 20 D6              BRA     BRG2
                      *
                      * JUMP,JSR EXTENDED
                      *
FFBA EE 01      JMPEXT  LDX     1,X
FFBC 20 D2              BRA     BRG2
```

```
                   *
                   * OPBCYT
                   *
                   * THIS ROUTINE DETERMINES THE £ OF BYTES IN AN INSTRUCTION
                   * GIVEN ITS OP CODE
                   *
                   * INPUT : A HOLDS THE OP CODE
                   *
                   * OUTPUT : X HOLDS INDEX OF TABLE ELEMENT
                   * B NOT RESTORED
                   * A HOLDS £ OF BYTES IN INSTRUCTION
                   * EXCEPT FOR BRANCHES IN WHICH CASE A IS NEGATIVE
                   *
FFBE               OPCBYT  EQU    *
FFBE 16                    TAB           B:=OP CODE
FFBF 44                    LSR A
FFC0 44                    LSR A
FFC1 44                    LSR A         PUT 4 UPPER BITS OF OP CODE INTO
FFC2 44                    LSR A         LOWER 4 BITS OF A
                   *
FFC3 CE FF D6              LDX    £OPBTTB  X:=ADDR OF TABLE
FFC6 8D DD                 BSR    INCX     INC X TO POINT TO CORRECT ENTRY
                   *
FFC8 A6 00                 LDA A  0,X      GET TABLE ENTRY
FFCA 26 09                 BNE    OPBTRT   IF NOT 0 THEN NO FURTHER PROCESSING NEEDED
                   *
                   * IF TOP 4 BITS=8 OR C ,THEN THERE ARE TWO CLASSE
                   * OF INSTRUCTIONS : 2 BYTE INSTRUCTIONS AND
                   * CE,8C AND 8E WHICH ARE 3 BYTE INSTRUCTIONS
                   *
FFCC 86 02                 LDA A  £2       £ OF BYTES IN MOST OF 8£ INSTRUCTIONS
FFCE C4 0D                 AND B  £%00001101
FFD0 C1 0C                 CMP B  £%00001100
FFD2 26 01                 BNE    OPBTRT   NO ,RETURN
FFD4 4C                    INC A           £ OF BYTES IN INSTRUCTION :=3
FFD5 39            OPBTRT  RTS             RETURN TO CALLER
                   *
                   * OP CODE TO NUMBER OF BYTES CONVERSION TABLE
                   *
                   *    £ BYTES TOP 4 BITS OF OPCODE
                   *
FFD6               OPBTTB  EQU    *
FFD6 01                    FCB    1        0
FFD7 01                    FCB    1        1
FFD8 82                    FCB    2+%10000000 2  MINUS = BRANCHES
FFD9 01                    FCB    1        3
FFDA 01                    FCB    1        4
FFDB 01                    FCB    1        5
FFDC 02                    FCB    2        6
FFDD 03                    FCB    3        7
FFDE 00                    FCB    0        8 £ BYTES=2 EXCEPT 8C,8E
FFDF 02                    FCB    2        9
FFE0 02                    FCB    2        A
```



```
FFE1 03                    FCB    3        B
FFE2 00                    FCB    0        C £ BYTES=2 EXCEPT CE
FFE3 02                    FCB    2        D
FFE4 02                    FCB    2        E
FFE5 03                    FCB    3        F
```

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

```
                  *
                  * RAM - LOCATIONS DEVOTED TO VARIABLE INFORMATION
                  *
      E400                ORG    RAM        START OF RAM
      0008        NBRBPT  EQU    8          £ OF BREAKPOINTS SUPPORTED
                  *
      E400        IOV     RMB    2          I/O INTERRUPT POINTER
      E402        NIO     RMB    2          NMI INTERRUPT POINTER
                  *
                  * THE FOLLOWING ARE INITIALIZED TO ZERO AT START
                  *
      E404        TRCADR  RMB    2          TRACE ADDRESS
      E406        NTRACE  RMB    2          NO. OF INSTRUCTIONS TO TRACE
      E408        OUTSW   RMB    1          OUTPUT SWITCH (0=ECHO INPUT)
      E409        PRTFLG  RMB    1          PRINT FLAG
      E40A        XHI     RMB    1          X REG HIGH (TEMP)
      E40B        XLOW    RMB    1          X REG LOW
                  *
      E40C        XSAVE   EQU    *
      E40C                RMB    2
                  *
      E40E        TRCINS  RMB    1          OPCODE REPLACED BY SWI IN TRACE MODE
      E40F        BRKSIN  RMB    1          1=BREAKS ARE IN USER PGM
      E410        BRKTRC  RMB    1          1=P-COUNTER IS AT BREAKPOINT AND
                  *              USER WANTS TO CONTINUE - ONE TRACE WILL
                  *              BE DONE AND BREAKPOINTS RESTORED
      E411                RMB    3          RESERVED FOR IO HANDLER
      E414                RMB    8          FREE SPACE
                  *
      E41C        BRKADR  RMB    NBRBPT*3   BREAKPOINT ADDRESS TABLE
      E434        BRKINS  EQU    *
                  *
                  * THE FOLLOWING ARE INITIALIZED AT START
                  *
      E434        SP      RMB    2          USER STACK POINTER
      E436        SWI1    RMB    2          SWI VECTOR
      E438        BRINS   RMB    8          STORAGE FOR CONDITIONNAL BRANCH ROUTINE
      E440        BRANEN  EQU    *          END OF BRANCH ROUTINE+1
                  *
                  * STACK
                  *
      E440        STKEND  RMB    128        RAM USED FOR STACK
      E4BF        STACK   EQU    *-1        START OF STACK AREA
                  *
                  *
                  *
                          END    START
                          END
```

NO ERROR(S) DETECTED

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	FFB1	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	FBB5
GTEXT	F403	IER	000E	IFR	000D	IN1HG	FD56	INACIA	FC43
INCH	FD9B	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	002B	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	SFE1	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				

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

```
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

F415                   ORG    $F415
                *
                *
                * GTEXT
                * ARGUMENTS D APPEL = AUCUN
                * REGISTRE MODIFIE :A
                *
F415 86 80      GTEXT  LDA A  £$80
F417 B7 E4 D3          STA A  DIRECT
F41A 86 C0             LDA A  £$C0
F41C B7 E7 FC          STA A  GCONTR
F41F 39                RTS
                *
                * GINIT
                * ARGUMENTS D'APPEL=AUCUN
                * REGISTRES MODIFIES=A,B
                *
F420 FF E4 CA   GINIT  STX    GDELTA
F423 CE F4 5A          LDX    £GCRTTB   INITIALISATION 6845
F426 5F                CLR B
F427 F7 E7 FE   GBCRT  STA B  GCRTC
F42A A6 00             LDA A  0,X
F42C B7 E7 FF          STA A  GCRTC+1
F42F 08                INX
F430 5C                INC B
F431 C1 10             CMP B  £16
F433 26 F2             BNE    GBCRT
F435 B6 E7 FD          LDA A  GCONTR+1  DEFINITION DU FOND COLORE
F438 88 40             EOR A  £$40
F43A B7 E7 FD          STA A  GCONTR+1
F43D 86 14             LDA A  £20       ATTENTE 20MS
F43F 5F                CLR B
F440 5A         GBWAIT DEC B
F441 26 FD             BNE    GBWAIT
F443 4A                DEC A
F444 26 FA             BNE    GBWAIT
F446 B6 E7 FD          LDA A  GCONTR+1
F449 88 40             EOR A  £$40
F44B B7 E7 FD          STA A  GCONTR+1
F44E FE E4 CA          LDX    GDELTA
                *
                * GRAPH
                * ARGUMENT D APPEL = AUCUN
                * REGISTRE MODIFIE : A
                *
F451 7F E4 D3   GRAPH  CLR    DIRECT
F454 86 40             LDA A  £$40
F456 B7 E7 FC          STA A  GCONTR
F459 39                RTS
                *TABLE D'INITIALISATION DU 6845
F45A 55         GCRTTB FCB    $55,64,69,6,$4D,0,$40,$44
F462 00                FCB    0,3,0,0,0,0,0,0
                *
```

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

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

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

                        *
                        * GPLOT
                        *
                        * ARGUMENTS D'APPEL=X-)ZONE DE PARAMETRES
                        * SOUS-PROGRAMMES APPELES=GMOVE,GTRACE
                        * REGISTRES MODIFIES=A,B
                        *
     F475 A6 00         GPLOT   LDA A  0,X
     F477 81 00                 CMP A  £0
     F479 26 06                 BNE    GCMD1
     F47B A6 01         GCMD0   LDA A  1,X
     F47D E6 02                 LDA B  2,X
     F47F 20 1B                 BRA    GMOVE
     F481 81 01         GCMD1   CMP A  £1
     F483 26 06                 BNE    GCMD2
     F485 A6 01                 LDA A  1,X
     F487 E6 02                 LDA B  2,X
     F489 20 1E                 BRA    GTRACE
     F48B 81 02         GCMD2   CMP A  £2
     F48D 26 0C                 BNE    GCMDE
     F48F A6 01                 LDA A  1,X
     F491 E6 02                 LDA B  2,X
     F493 8D 07                 BSR    GMOVE
     F495 A6 03                 LDA A  3,X
     F497 E6 04                 LDA B  4,X
     F499 20 0E                 BRA    GTRACE
     F49B 39            GCMDE   RTS
```

```
                       *
                       * GMOVE   POSITIONNEMENT SUR UN POINT
                       *         ET ECRITURE
                       *
                       * ARGUMENTS D'APPEL=(A)ABCISSE,(B)ORDONNEE
                       * REGISTRES MODIFIES=A
                       *
F49C B7 E7 FB  GMOVE   STA A  GCURX
F49F F7 E7 F9          STA B  GCURY
F4A2 B6 E7 FD          LDA A  GCONTR+1
F4A5 B7 E7 FC          STA A  GCONTR
F4A8 39                RTS
                       *
                       * GTRACE   TRACE D'UNE LIGNE DROITE
                       *
                       * ARGUMENTS D'APPEL=(A)ABSCISSE,(B)ORDONNEE
                       * SOUS-PROGRAMMES APPELES=GCINC,GAINC
                       * REGISTRES MODIFIES=A,B
                       *
F4A9 B0 E7 FB  GTRACE  SUB A  GCURX    CALCUL DU DEPLACEMENT
F4AC 22 07             BHI    GSUPX    SUIVANT X ET SON SIGNE
F4AE 26 0C             BNE    GINFX
F4B0 B7 E4 CE          STA A  GDX
F4B3 20 0D             BRA    GMVY
F4B5 B7 E4 CE  GSUPX   STA A  GDX
F4B8 86 01             LDA A  £1
F4BA 20 06             BRA    GMVY
F4BC 40        GINFX   NEG A
F4BD B7 E4 CE          STA A  GDX
F4C0 86 FF             LDA A  £-1
F4C2 B7 E4 D0  GMVY    STA A  GSX
F4C5 F0 E7 F9          SUB B  GCURY    CALCUL DU DEPLACEMENT
F4C8 22 07             BHI    GSUPY    SUIVANT Y ET SON SIGNE
F4CA 26 0C             BNE    GINFY
F4CC F7 E4 CF          STA B  GDY
F4CF 20 0D             BRA    GCMPD
F4D1 F7 E4 CF  GSUPY   STA B  GDY
F4D4 C6 01             LDA B  £1
F4D6 20 06             BRA    GCMPD
F4D8 50        GINFY   NEG B
F4D9 F7 E4 CF          STA B  GDY
F4DC C6 FF             LDA B  £-1
F4DE F7 E4 D1  GCMPD   STA B  GSY
F4E1 B6 E4 CE          LDA A  GDX      COMPARAISON DES DEPLACEMENTS
F4E4 B1 E4 CF          CMP A  GDY
F4E7 22 21             BHI    GPASX
F4E9 26 54             BNE    GPASY
F4EB 4D                TST A
F4EC 27 1B             BEQ    GPRET    ILS SONT NULS:RIEN A FAIRE
F4EE F6 E7 FB  GBPLT1  LDA B  GCURX    ILS SONT EGAUX:INCREMENTS
F4F1 FB E4 D0          ADD B  GSX      ENTIER GSX ET GSY
F4F4 F7 E7 FB          STA B  GCURX
F4F7 F6 E7 F9          LDA B  GCURY
```

```
F4FA FB E4 D1               ADD B  GSY
F4FD F7 E7 F9               STA B  GCURY
F500 F6 E7 FD               LDA B  GCONTR+1
F503 F7 E7 FC               STA B  GCONTR
F506 4A                     DEC A
F507 26 E5                  BNE    GBPLT1
F509 39            GPRET    RTS
F50A B6 E4 CE      GPASX    LDA A  GDX        DEPLACEMENT PLUS GRAND
F50D F6 E4 CF               LDA B  GDY        SUIVANT X:INCREMENT ENTIER POUR X
F510 27 DC                  BEQ    GBPLT1     CALCUL DE L'INCREMENT POUR Y
F512 F7 E4 CA               STA B  GDELTA
F515 F6 E7 F9               LDA B  GCURY
F518 F7 E4 C7               STA B  GCUR
F51B F6 E4 D1               LDA B  GSY
F51E F7 E4 CD               STA B  GSD
F521 8D 50                  BSR    GCINC
F523 BD F5 B6      GBPLT2   JSR    GAINC      TRACE
F526 F6 E7 FB               LDA B  GCURX
F529 FB E4 D0               ADD B  GSX
F52C F7 E7 FB               STA B  GCURX
F52F F6 E4 C7               LDA B  GCUR
F532 F7 E7 F9               STA B  GCURY
F535 F6 E7 FD               LDA B  GCONTR+1
F538 F7 E7 FC               STA B  GCONTR
F53B 4A                     DEC A
F53C 26 E5                  BNE    GBPLT2
F53E 39                     RTS
F53F B6 E4 CF      GPASY    LDA A  GDY        DEPLACEMENT PLUS GRAND
F542 F6 E4 CE               LDA B  GDX        SUIVANT Y:INCREMENT ENTIER POUR Y
F545 27 A7                  BEQ    GBPLT1     CALCUL DE L'INCREMENT POUR X
F547 F7 E4 CA               STA B  GDELTA
F54A F6 E7 FB               LDA B  GCURX
F54D F7 E4 C7               STA B  GCUR
F550 F6 E4 D0               LDA B  GSX
F553 F7 E4 CD               STA B  GSD
F556 8D 1B                  BSR    GCINC
F558 8D 5C         GBPLT3   BSR    GAINC      TRACE
F55A F6 E4 C7               LDA B  GCUR
F55D F7 E7 FB               STA B  GCURX
F560 F6 E7 F9               LDA B  GCURY
F563 FB E4 D1               ADD B  GSY
F566 F7 E7 F9               STA B  GCURY
F569 F6 E7 FD               LDA B  GCONTR+1
F56C F7 E7 FC               STA B  GCONTR
F56F 4A                     DEC A
F570 26 E6                  BNE    GBPLT3
F572 39                     RTS
```

```
                      *
                      * GCINC   CALCUL DES INCREMENTS FRACTIONNAIRES
                      *
                      * ARGUMENTS D'APPEL=(A)NOMBRE DE PAS
                      *                   (GDELTA)DEPLACEMENT
                      *                   (GSD)SIGNE
                      * REGISTRES MODIFIES=B
                      *
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 C8                 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
```

```
                    NAM    MUSIBA4 * 12.FEV.81
                    OPT    PAG
E4D4                ORG    $E4D4
              *
              * RESERVATIONS MEMOIRE RAM
              *
E4D4          POKTIM  RMB   1
E4D5          POKCOR  RMB   1
E4D6          TEMPO   RMB   1
E4D7          COMMUN  RMB   2
E4D9          EXPLOR  RMB   2
E4DB          DELPUL  RMB   1
E4DC          DELREP  RMB   1
E4DD          DEL2    RMB   2
E4DF          DEL8L   RMB   2
E4E1          DEREPB  RMB   1
E4E2          NBPER   RMB   1
E4E3          NBPERB  RMB   1
E4E4          NPH     RMB   1
E4E5          NPB     RMB   1
E4E6          FRAC    RMB   1
E4E7          STROBE  RMB   1
E4E8          NIVEAU  RMB   1
E4E9          NIVB    RMB   1
E4EA          DECLOG  RMB   1
E4EB          DECPAS  RMB   1
E4EC          SILENC  RMB   1
E4ED          TIMJEU  RMB   1
E4EE          GARX    RMB   2
E4F0          BY1     RMB   1
E4F1          BY2     RMB   1
E4F2          BY3     RMB   1
E4F3          BY4     RMB   1
E4F4          BY5     RMB   1
              ************
E861          LESON   EQU   $E861
              **********************
F5D3                  ORG   $F5D3
F5D3 46       TABNOT  FCB   70,67,63,60,57,54,51,48,45,43,41,39,37
F5E0 23               FCB   35,33,31,29,27,26,24,23,21,20,19,18
F5EC 11               FCB   17,16,15,14,13,12,11,10,9,9,8,7
F5F8 07               FCB   7,6,5,5,4,4,3,3,3,2,2,1,0
F605 12               FCB   18,16,15,14,13,12,12,11,10,9,9,9,7
F612 07               FCB   7,6,6,5,5,5,4,4,4,3,3,3
F61E 02               FCB   2,2,2,2,1,1,1,1,1,1,1,1
F62A 00               FCB   0,0,0,0,0,0,0,0,0,0,0,0,0
F637 3B               FCB   59,239,238,213,213,235,23
F63E 56               FCB   86,121,221,34,119,218
F644 4C               FCB   76,204,88,240,147,16
F64A C8               FCB   200,89,35,198,112,34
F650 DB               FCB   219,155,97,45,254,213
F656 B1               FCB   177,146,119,48,29,14
F65C D3               FCB   211,202,197,147,148,104
```

```
F662 6E                 FCB    110,70,32,45,12,28,46
F669 10                 FCB    16,16,17,18,19,20,21
F670 17                 FCB    23,24,25,27,28,30
F676 20                 FCB    32,34,36,38,40,43
F67C 2D                 FCB    45,48,51,54,57,60
F682 3F                 FCB    63,66,72,76,80,85
F688 5A                 FCB    90,96,101,107,114,120
F68E 7F                 FCB    127,135,143,152,161,170
F694 B4                 FCB    180,191,202,214,227,241,255
F69B 44        DOASI    FCC    /DO33RE33MIFA33SO33LA33SI/
F6B3 51        TRADUR   FCC    /QTDCNBR/
               *******************************
F6BA FF        TABTEM   FCB    255,220,180,136,105,86,73,63
F6C2 38                 FCB    56,50,45,41,38,35,33,31
               **********************
F6CA F7 E4 E6  MUL2P1   STA B  FRAC
F6CD FE E4 E2           LDX    NBPER
F6D0 FF E4 E4           STX    NPH
F6D3 4F                 CLR A
F6D4 B7 E4 E2           STA A  NBPER
F6D7 B7 E4 E3           STA A  NBPERB
F6DA 86 80              LDA A  £$80
F6DC B7 E4 E7           STA A  STROBE
F6DF B6 E4 E7  MUL1     LDA A  STROBE
F6E2 B4 E4 E6           AND A  FRAC
F6E5 26 04              BNE    MUL2
F6E7 8D 16              BSR    DEC1M
F6E9 20 0E              BRA    MUL3
F6EB 8D 12     MUL2     BSR    DEC1M
F6ED FB E4 E3           ADD B  NBPERB
F6F0 B9 E4 E2           ADC A  NBPER
F6F3 B7 E4 E2           STA A  NBPER
F6F6 F7 E4 E3           STA B  NBPERB
F6F9 74 E4 E7  MUL3     LSR    STROBE
F6FC 26 E1              BNE    MUL1
F6FE 39                 RTS
F6FF B6 E4 E4  DEC1M    LDA A  NPH
F702 F6 E4 E5           LDA B  NPB
F705 54        DECUN    LSR B
F706 44                 LSR A
F707 24 02              BCC    FDUN
F709 CB 80              ADD B  £$80
F70B B7 E4 E4  FDUN     STA A  NPH
F70E F7 E4 E5           STA B  NPB
F711 39                 RTS
F712 FF E4 D7  PLUSA    STX    COMMUN
F715 F6 E4 D7           LDA B  COMMUN
F718 BB E4 D8           ADD A  COMMUN+1
F71B C9 00              ADC B  £0
F71D F7 E4 D7           STA B  COMMUN
F720 B7 E4 D8           STA A  COMMUN+1
F723 FE E4 D7           LDX    COMMUN
F726 39                 RTS
               ************
```

```
F727 A6 00    SPNOTE  LDA A  X
F729 84 DF            AND A  £$DF
F72B B7 E4 F0         STA A  BY1
F72E A6 01            LDA A  1,X
F730 84 DF            AND A  £$DF
F732 B7 E4 F1         STA A  BY2
F735 A6 02            LDA A  2,X
F737 B7 E4 F2         STA A  BY3
F73A A6 03            LDA A  3,X
F73C 84 DF            AND A  £$DF
F73E B7 E4 F3         STA A  BY4
F741 A6 04            LDA A  4,X
F743 B7 E4 F4         STA A  BY5
F746 B6 E4 D6         LDA A  TEMPO
F749 26 05            BNE    OKTEM
F74B 86 8A            LDA A  £138
F74D B7 E4 D6         STA A  TEMPO
F750 8D 13    OKTEM   BSR    PREPAR
F752 BD F8 34         JSR    LANOTE
F755 39               RTS
              ************
F756 44       SPTEMP  LSR A
F757 44               LSR A
F758 44               LSR A
F759 44               LSR A
F75A CE F6 8A         LDX    £TABTEM
F75D 8D B3            BSR    PLUSA
F75F A6 00            LDA A  X
F761 B7 E4 D6         STA A  TEMPO
F764 39               RTS
              **************
F765 B6 E4 D4 PREPAR  LDA A  POKTIM
F768 26 03            BNE    OKTIM
F76A 73 E4 D4         COM    POKTIM
F76D 86 02    OKTIM   LDA A  £2
F76F CE F6 98         LDX    £DOASI
F772 C6 FF            LDA B  £$FF
F774 F7 E4 EC         STA B  SILENC
F777 FF E4 EE RECNOT  STX    GARX
F77A EE 00            LDX    X
F77C BC E4 F0         CPX    BY1
F77F 27 11            BEQ    OKMOT
F781 4C               INC A
F782 FE E4 EE         LDX    GARX
F785 08               INX
F786 08               INX
F787 8C F6 B3         CPX    £DOASI+24
F78A 26 EB            BNE    RECNOT
F78C 4F               CLR A
F78D 7F E4 EC         CLR    SILENC
F790 20 11            BRA    PAUSE
F792 F6 E4 F2 OKMOT   LDA B  BY3
F795 C0 02            SUB B  £2
F797 2E 02            BGT    EVIT1
```

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

F799 C6 01                 LDA B  £1
F79B 58          EVIT1     ASL B
F79C 58                    ASL B
F79D 1B                    ABA
F79E 58                    ASL B
F79F 1B                    ABA
F7A0 BB E4 F4              ADD A  BY5
F7A3 CE F5 D3    PAUSE     LDX    £TABNOT
F7A6 BD F7 12              JSR    PLUSA
F7A9 A6 00                 LDA A  X
F7AB B7 E4 DB              STA A  DELPUL
F7AE A6 32                 LDA A  50,X
F7B0 B7 E4 DC              STA A  DELREP
F7B3 A6 64                 LDA A  100,X
F7B5 B7 E4 E1              STA A  DEREPB
F7B8 A6 96                 LDA A  150,X
F7BA B7 E4 E2              STA A  NBPER
F7BD 7F E4 E3              CLR    NBPER+1
F7C0 44                    LSR A
F7C1 44                    LSR A
F7C2 44                    LSR A
F7C3 44                    LSR A
F7C4 B7 E4 EA              STA A  DECLOG
F7C7 B7 E4 EB              STA A  DECPAS
F7CA B6 E4 E1              LDA A  DEREPB
F7CD 84 06                 AND A  £6
F7CF CE F8 2D              LDX    £DELAI8
F7D2 4D          TESSAU    TST A
F7D3 27 05                 BEQ    FINSAU
F7D5 09                    DEX
F7D6 4A                    DEC A
F7D7 4A                    DEC A
F7D8 20 F8                 BRA    TESSAU
F7DA FF E4 DD    FINSAU    STX    DEL2
F7DD B6 E4 DC              LDA A  DELREP
F7E0 F6 E4 E1              LDA B  DEREPB
F7E3 BD F7 05              JSR    DECUN
F7E6 BD F7 05              JSR    DECUN
F7E9 BD F7 05              JSR    DECUN
F7EC B7 E4 DF              STA A  DEL8L
F7EF F7 E4 E0              STA B  DEL8L+1
F7F2 B6 E4 E1              LDA A  DEREPB
F7F5 4F                    CLR A
F7F6 CE F6 B3              LDX    £TRADUR
F7F9 E6 00       RECDUR    LDA B  X
F7FB F1 E4 F3              CMP B  BY4
F7FE 27 07                 BEQ    OKDUR
F800 4C                    INC A
F801 08                    INX
F802 8C F6 BA              CPX    £TRADUR+7
F805 26 F2                 BNE    RECDUR
F807 C6 01       OKDUR     LDA B  £1
F809 4D          CALDUR    TST A
F80A 27 04                 BEQ    CALPER
```

```
F80C 58                ASL B
F80D 4A                DEC A
F80E 20 F9             BRA    CALDUR
F810 BD F6 CA  CALPER  JSR    MUL2P1
F813 FE E4 DE          LDA B  TEMPO
F816 BD F6 CA          JSR    MUL2P1
F819 86 3F             LDA A  £63
F81B B4 E4 EC          AND A  SILENC
F81E B7 E4 E8          STA A  NIVEAU
F821 7F E4 E9          CLR    NIVB
F824 39                RTS
               ***********
F825 FE E4 DD  DELMUL  LDX    DEL2
F828 6E 00             JMP    X
F82A 01                NOP
F82B 01                NOP
F82C 01                NOP
F82D FE E4 DF  DELAIB  LDX    DEL8L
F830 09        PLUS8   DEX
F831 26 FD             BNE    PLUS8
F833 39                RTS
               **********
F834 C6 08     LANOTE  LDA B  £8
F836 B6 E4 D4          LDA A  POKTIM
F839 B7 E4 ED          STA A  TIMJEU
F83C 86 40     LES8IM  LDA A  £64
F83E 78 E4 ED          ASL    TIMJEU
F841 24 05             BCC    BITABS
F843 BB E4 E8          ADD A  NIVEAU
F846 20 05             BRA    OUTSON
F848 B0 E4 E8  BITABS  SUB A  NIVEAU
F84B 20 00             BRA    OUTSON
F84D B7 E8 61  OUTSON  STA A  LESON
F850 B6 E4 DB          LDA A  DELPUL
F853 27 03             BEQ    PASDE8
F855 4A        ET8     DEC A
F856 26 FD             BNE    ET8
F858 5A        PASDE8  DEC B
F859 26 E1             BNE    LES8IM
F85B C6 40             LDA B  £64
F85D F7 E8 61          STA B  LESON
F860 8D C3             BSR    DELMUL
F862 B6 E4 D5          LDA A  POKCOR
F865 27 20             BEQ    RAF1
F867 7A E4 EB          DEC    DECPAS
F86A 26 2C             BNE    RAF2
F86C B6 E4 EA          LDA A  DECLOG
F86F B7 E4 EB          STA A  DECPAS
F872 B6 E4 E8          LDA A  NIVEAU
F875 F6 E4 E9          LDA B  NIVB
F878 F0 E4 E8          SUB B  NIVEAU
F87B 82 00             SBC A  £0
F87D F0 E4 E8          SUB B  NIVEAU
F880 82 00             SBC A  £0
```

```
F882 F0 E4 E8          SUB B  NIVEAU
F885 82 00             SBC A  £0
F887 F0 E4 E8          SUB B  NIVEAU
F88A 82 00             SBC A  £0
F88C B7 E4 E8          STA A  NIVEAU
F88F F7 E4 E9          STA B  NIVB
F892 20 09             BRA    TESTAM
F894 86 09    RAF1     LDA A  £9
F896 20 02             BRA    COMPEN
F898 86 07    RAF2     LDA A  £7
F89A 4A       COMPEN   DEC A
F89B 26 FD             BNE    COMPEN
F89D FE E4 E2 TESTAM   LDX    NBPER
F8A0 09                DEX
F8A1 FF E4 E2          STX    NBPER
F8A4 26 8E             BNE    LANOTE
F8A6 39                RTS
              ********************
                       END
                       END

NO ERROR(S) DETECTED
```


SYMBOL TABLE:

```
ADVISU E7F8   BITABS F848   BY1    E4F0   BY2    E4F1   BY3    E4F2
BY4    E4F3   BY5    E4F4   CALDUR F809   CALPER F810   COMMUN E4D7
COMPEN F89A   DEC1M  F6FF   DECLOG E4EA   DECPAS E4EB   DECUN  F705
DEL2   E4DD   DEL8L  E4DF   DELAI8 F82D   DELMUL F825   DELPUL E4DB
DELREP E4DC   DEREPB E4E1   DIRECT E4D3   DOASI  F69B   ET8    F855
EVIT1  F798   EXPLOR E4D9   FOUN   F70B   FINSAU F7DA   FRAC   E4E6
GAINC  F5B6   GARX   E4EE   GBCRT  F427   GBDIV  F585   GBPLT1 F4EE
GBPLT2 F523   GBPLT3 F55B   GBWAIT F440   GCINC  F573   GCMD0  F478
GCMD1  F481   GCMD2  F48B   GCMDE  F49B   GCMPD  F4DE   GCMPT  E4D2
GCOLOR F46A   GCONTR E7FC   GCRTC  E7FE   GCRTTB F45A   GCUR   E4C7
GCURX  E7FB   GCURY  E7F9   GDELTA E4CA   GDSUB  F591   GDX    E4CE
GDY    E4CF   GFBDIV F595   GINFX  F4BC   GINFY  F4D2   GINIT  F420
GMOVE  F49C   GMVY   F4C2   GPASX  F50A   GPASY  F53F   GPLOT  F475
GPRET  F509   GRAPH  F451   GRCINC F5B4   GSD    E4CD   GSUPX  F4B5
GSUPY  F4D1   GSX    E4D0   GSY    E4D1   GTEXT  F415   GTRACE F4A9
LANOTE F834   LESBIM F83C   LESON  E861   MUL1   F6DF   MUL2   F6EB
MUL2P1 F6CA   MUL3   F6F9   NBPER  E4E2   NBPERB E4E3   NIVB   E4E9
NIVEAU E4E8   NPB    E4E5   NPH    E4E4   OKDUR  F807   OKMOT  F792
OKTEM  F750   OKTIM  F76D   OUTSON F840   PASDEB FB58   PAUSE  F7A3
PLUS8  F830   PLUSA  F712   POKCOR E4D5   POKTIM E4D4   PREPAR F765
RAF1   F894   RAF2   F898   RECDUR F7F9   RECNOT F777   SILENC E4EC
SPNOTE F727   SPTEMP F756   STROBE E4E7   TABNOT F5D3   TABTEM F6BA
TEMPO  E4D6   TESSAU F7D2   TESTAM F89D   TIMJEU E4ED   TRADUR F6B3
```

```
               NAM    V1MON V1.4
               OPT    PAG

       ************************************
       *                                  *
       *     V1MON                        *
       *     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
```

```
F400                 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

                    *
                    * TABLE DE BRANCHEMENTS
                    *
F400 7E F4 1A              JMP   GINIT     GR
F403 7E F4 0F              JMP   GTEXT     TEXT
F406 7E F4 64              JMP   GCOLOR    COLOR
F409 7E F4 6F              JMP   GPLOT     PLOT
F40C 7E F4 4B              JMP   GRAPH     GRAF
                    *
                    *
                    * GTEXT
                    * ARGUMENTS D APPEL = AUCUN
                    * REGISTRE MODIFIE :A
                    *
F40F 86 80          GTEXT  LDA A £$80
F411 B7 E4 D3              STA A DIRECT
F414 86 C0                 LDA A £$C0
F416 B7 E7 FC              STA A GCONTR
F419 39                    RTS
                    *
                    * GINIT
                    * ARGUMENTS D'APPEL=AUCUN
                    * REGISTRES MODIFIES=A,B
                    *
F41A FF E4 CA       GINIT  STX   GDELTA
F41D CE F4 54              LDX   £GCRTTB    INITIALISATION 6845
F420 5F                    CLR B
F421 F7 E7 FE       GBCRT  STA B GCRTC
F424 A6 00                 LDA A 0,X
F426 B7 E7 FF              STA A GCRTC+1
F429 08                    INX
F42A 5C                    INC B
F42B C1 10                 CMP B £16
F42D 26 F2                 BNE   GBCRT
F42F B6 E7 FD              LDA A GCONTR+1  DEFINITION DU FOND COLORE
F432 88 40                 EOR A £$40
F434 B7 E7 FD              STA A GCONTR+1
F437 86 14                 LDA A £20       ATTENTE 20MS
F439 5F                    CLR B
F43A 5A             GBWAIT DEC B
F43B 26 FD                 BNE   GBWAIT
F43D 4A                    DEC A
F43E 26 FA                 BNE   GBWAIT
F440 B6 E7 FD              LDA A GCONTR+1
F443 88 40                 EOR A £$40
F445 B7 E7 FD              STA A GCONTR+1
F448 FE E4 CA              LDX   GDELTA
                    *
                    * GRAPH
                    * ARGUMENT D APPEL = AUCUN
                    * REGISTRE MODIFIE : A
                    *
F44B 7F E4 D3       GRAPH  CLR   DIRECT
F44E 86 40                 LDA A £$40
F450 B7 E7 FC              STA A GCONTR
```

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

F453 39                 RTS
                *TABLE D'INITIALISATION DU 6845
F454 55         GCRTTB  FCB     $55,64,69,6,$40,0,$40,$44
F45C 00                 FCB     0,3,0,0,0,0,0,0
                *
                * GCOLOR
                *
                * ARGUMENTS D'APPEL=(A)CODE DE COULEUR
                * REGISTRES MODIFIES=A
                *
F464 84 07      GCOLOR  AND A  £$07
F466 8A 40              ORA A  £$40
F468 BA E4 03           ORA A  DIRECT
F46B B7 E7 FD           STA A  GCONTR+1
F46E 39                 RTS
```

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

                *
                * GPLOT
                *
                * ARGUMENTS D'APPEL=X->ZONE DE PARAMETRES
                * SOUS-PROGRAMMES APPELES=GMOVE,GTRACE
                * REGISTRES MODIFIES=A,B
                *
F46F A6 00      GPLOT   LDA A  0,X
F471 81 00              CMP A  £0
F473 26 06              BNE    GCMD1
F475 A6 01      GCMD0   LDA A  1,X
F477 E6 02              LDA B  2,X
F479 20 1B              BRA    GMOVE
F47B 81 01      GCMD1   CMP A  £1
F47D 26 06              BNE    GCMD2
F47F A6 01              LDA A  1,X
F481 E6 02              LDA B  2,X
F483 20 1E              BRA    GTRACE
F485 81 02      GCMD2   CMP A  £2
F487 26 0C              BNE    GCMDE
F489 A6 01              LDA A  1,X
F48B E6 02              LDA B  2,X
F48D 8D 07              BSR    GMOVE
F48F A6 03              LDA A  3,X
F491 E6 04              LDA B  4,X
F493 20 0E              BRA    GTRACE
F495 39         GCMDE   RTS
```

```
                    *
                    * 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 B6 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 18             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

F4FA F6 E7 FD           LDA B  GCONTR+1
F4FD F7 E7 FC           STA B  GCONTR
F500 4A                 DEC A
F501 26 E5              BNE    GBPLT1
F503 39        GPRET    RTS
F504 B6 E4 CE  GPASX    LDA A  GDX        DEPLACEMENT PLUS GRAND
F507 F6 E4 CF           LDA B  GDY        SUIVANT X:INCREMENT ENTIER POUR X
F50A 27 DC              BEQ    GBPLT1     CALCUL DE L'INCREMENT POUR Y
F50C F7 E4 CA           STA B  GDELTA
F50F F6 E7 F9           LDA B  GCURY
F512 F7 E4 C7           STA B  GCUR
F515 F6 E4 D1           LDA B  GSY
F518 F7 E4 CD           STA B  GSD
F51B 8D 50              BSR    GCINC
F51D BD F5 B0  GBPLT2   JSR    GAINC      TRACE
F520 F6 E7 FB           LDA B  GCURX
F523 FB E4 D0           ADD B  GSX
F526 F7 E7 FB           STA B  GCURX
F529 F6 E4 C7           LDA B  GCUR
F52C F7 E7 F9           STA B  GCURY
F52F F6 E7 FD           LDA B  GCONTR+1
F532 F7 E7 FC           STA B  GCONTR
F535 4A                 DEC A
F536 26 E5              BNE    GBPLT2
F538 39                 RTS
F539 B6 E4 CF  GPASY    LDA A  GDY        DEPLACEMENT PLUS GRAND
F53C F6 E4 CE           LDA B  GDX        SUIVANT Y:INCREMENT ENTIER POUR Y
F53F 27 A7              BEQ    GBPLT1     CALCUL DE L'INCREMENT POUR X
F541 F7 E4 CA           STA B  GDELTA
F544 F6 E7 FB           LDA B  GCURX
F547 F7 E4 C7           STA B  GCUR
F54A F6 E4 D0           LDA B  GSX
F54D F7 E4 CD           STA B  GSD
F550 8D 1B              BSR    GCINC
F552 8D 5C     GBPLT3   BSR    GAINC      TRACE
F554 F6 E4 C7           LDA B  GCUR
F557 F7 E7 FB           STA B  GCURX
F55A F6 E7 F9           LDA B  GCURY
F55D FB E4 D1           ADD B  GSY
F560 F7 E7 F9           STA B  GCURY
F563 F6 E7 FD           LDA B  GCONTR+1
F566 F7 E7 FC           STA B  GCONTR
F569 4A                 DEC A
F56A 26 E6              BNE    GBPLT3
F56C 39                 RTS
```

```
                   *
                   * 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

                  * 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
                         END
```

```
F5CC                  ORG    $F5CC
                      NAM    8 OR 5" DISK BOOTSTRAP


                 *********************************
                 *                               *
                 * -5- COMMAND                   *
                 *   DISK BOOTSTRAP              *
                 *   5 OR 8" DMAF2 - 5" MFB      *
                 *                               *
                 *********************************

                 * EQUATES FOR WD 1795 AND DMA

EB20             STAREG  EQU   $EB20
EB21             TRKREG  EQU   $EB21
EB22             SECREG  EQU   $EB22
EB23             DATREG  EQU   $EB23
EB24             DRVREG  EQU   $EB24
EB00             DMAADD  EQU   $EB00
EB02             DMACON  EQU   $EB02
EB10             DMACOM  EQU   $EB10
EB14             DMAPRI  EQU   $EB14
A100             LOADER  EQU   $A100

                 * EQUATES FOR WD 1791

E8E0             DRVR91  EQU   $E8E0
E8F0             COMR91  EQU   $E8F0
E8F2             SECR91  EQU   $E8F2
E8F3             DATR91  EQU   $E8F3

                 * PROGRAM STARTS HERE

F5CC 86 01       DKBOOT  LDA A  £$01
F5CE B7 EB 22            STA A  SECREG
F5D1 8D 69               BSR    DEL1
F5D3 81 EB 22            CMP A  SECREG   TEST IF DMAF2 PRESENT
F5D6 26 67               BNE    MFBOOT   IF NOT BOOT ON MFB

F5D8 C6 BF       DMABOT  LDA B  £$BF
F5DA F7 EB 24            STA B  DRVREG   SELECT DRIVE
F5DD 8D 4C               BSR    READY    TEST READY
F5DF 27 0E               BEQ    SBOOT    IF READY CONTINUE
F5E1 C6 7F               LDA B  £$7F     SELECT 5"
F5E3 F7 EB 24            STA B  DRVREG
F5E6 8D 49               BSR    ONESEC   ONE SECOND DELAY
F5E8 8D 41               BSR    READY    TEST READY
F5EA 27 03               BEQ    SBOOT    IF READY CONTINUE
F5EC 7E FD 12            JMP    CONTRL
F5EF 86 09       SBOOT   LDA A  £9
F5F1 B7 EB 20            STA A  STAREG   SEND RESTOR COMMAND
F5F4 8D 44               BSR    DELAY
F5F6 8D 2D               BSR    TEST     WAIT TILL NOT BUSY
F5F8 CE FE FF            LDX    £$FEFF
```

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

F5FB FF EB 02          STX    DMACON    SET BUFFER LENGTH
F5FE CE 5E FF          LDX    £$5EFF
F601 FF EB 00          STX    DMAADD    SET BUFFER ADDRESS
F604 86 FD             LDA A  £$FD
F606 B7 EB 10          STA A  DMACOM
F609 86 FE             LDA A  £$FE
F60B B7 EB 14          STA A  DMAPRI
F60E 86 8C             LDA A  £$8C
F610 B7 EB 20          STA A  STAREG    SEND READ COMMAND
F613 FE EB 02  WAIT    LDX    DMACON
F616 8C FE FF          CPX    £$FEFF
F619 27 F8             BEQ    WAIT      WAIT TIL DMA OP. FINISHED
F61B 86 FF             LDA A  £$FF      RESET DMA
F61D B7 EB 14          STA A  DMAPRI
F620 8D 03             BSR    TEST
F622 7E A1 00  GOLOAD  JMP    LOADER    GOTO LOADER
F625 8D 04     TEST    BSR    READY     GET STATUS
F627 47                ASR A
F628 25 FB             BCS    TEST      WAIT TIL NOT BUSY
F62A 39                RTS
F62B B6 EB 20  READY   LDA A  STAREG    GET WD STATUS
F62E 85 80             BIT A  £$80
F630 39                RTS

F631 CE 00 00  ONESEC  LDX    £0        ONE SECOND DELAY
F634 08        D15     INX
F635 09                DEX
F636 09                DEX
F637 26 FB             BNE    D15
F639 39                RTS

F63A 8D 00     DELAY   BSR    DEL1
F63C 8D 00     DEL1    BSR    RTN
F63E 39        RTN     RTS

               * BOOTSTRAP FOR MFB

F63F 43        MFBOOT  COM A            SET SECTOR ONE
F640 B7 E8 F2          STA A  SECR91    AND START MOTOR
F643 7F E8 E0          CLR    DRVR91    SELECT DRIVE 0
F646 8D E9             BSR    ONESEC    DELAY ONE SECOND
F648 C6 F4             LDA B  £$F4      SEND RESTOR COMMAND
F64A F7 E8 F0          STA B  COMR91
F64D 8D EB             BSR    DELAY
F64F F6 E8 F0  LOOP1   LDA B  COMR91
F652 57                ASR B
F653 24 FA             BCC    LOOP1     WAIT TILL NOT BUSY
F655 86 73             LDA A  £$73      SEND READ COMMAND
F657 B7 E8 F0          STA A  COMR91
F65A 8D DE             BSR    DELAY
F65C CE A1 00          LDX    £LOADER   ADDRESS OF LOADER
F65F F6 E8 F0  LOOP2   LDA B  COMR91    GET WD STATUS
F662 57                ASR B            CHECK BUSY
F663 25 BD             BCS    GOLOAD    IF NOT BUSY GO TO LOADER
F665 57                ASR B            CHECK DATA PRESENT
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

F671                 ORG    $F671
                     NAM    2480 SCREEN HANDLER


            ******************************
            *                            *
            *    24*80 SCREEN HANDLER    *
            *                            *
            ******************************


            *
            *    I/O ROUTINES
            *    ------------
            *
            *       CRTINT     INITIALIZATIONS
            *       OUTPUT     OUTPUT ONE CHARACTER
            *       BELL       OUTPUT BELL
            *
```

```
                    *
                    * 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
```

```
F6D2 26 FD              BNE    BELDE1
F6D4 39                 RTS
                *
                * CLEAR SCREEN
                *
F6D5 8D A5      CLEAR   BSR    CLEAR2      CLEAR REFRESH MEMORY
                *
                * CURSOR HOME
                *
F6D7 7F E5 71   HOME    CLR    LINE        RESET LINE NUMBER
                *
                * LINE RETURN
                *
F6DA 7F E5 72   LINRET  CLR    COLUMN      RESET COLUMN NUMBER
F6DD 20 58              BRA    MOVEA       GOTO MOVE CURSOR
                *
                * MOVE CURSOR
                *
F6DF 7C E5 72   INCCL   INC    COLUMN      UPDATE COLUMN NUMBER
F6E2 B6 E5 72           LDA A  COLUMN
F6E5 81 4F              CMP A  £79         END OF LINE ?
F6E7 22 14              BHI    NEWLIN      IF YES ,NEW LINE
F6E9 FE E5 73           LDX    CURAD       UPDATE CURSOR ADDRESS
F6EC 08                 INX
F6ED FF E5 73   DECCL1  STX    CURAD
F6F0 20 66              BRA    OUTCUR      GOTO OUTPUT CURSOR
F6F2 7A E5 72   DECCL   DEC    COLUMN      DEC COLUMN NUMBER
F6F5 2B 24              BMI    UPLINE      BRANCH IF PREVIOUS LINE
F6F7 FE E5 73           LDX    CURAD       UPDATE CURSOR ADDRESS
F6FA 09                 DEX
F6FB 20 F0              BRA    DECCL1      GOTO OUTPUT NEW CURSOR
F6FD 7F E5 72   NEWLIN  CLR    COLUMN      CLEAR COLUMN NUMBER
F700 7C E5 71   INCLN   INC    LINE
F703 B6 E5 71           LDA A  LINE
F706 81 17              CMP A  £23
F708 23 2D              BLS    MOVEA
F70A 7D E4 18           TST    PAGFG
F70D 26 07              BNE    INCLN1
F70F 7A E5 71           DEC    LINE
F712 8D 5A              BSR    SCROLL
F714 20 21              BRA    MOVEA
F716 7F E5 71   INCLN1  CLR    LINE
F719 20 1C              BRA    MOVEA
                *
F71B 86 4F      UPLINE  LDA A  £79         CURSOR ON END OF LINE
F71D B7 E5 72           STA A  COLUMN
F720 7A E5 71   DECLN   DEC    LINE
F723 2A 12              BPL    MOVEA
F725 7D E4 18           TST    PAGFG
F728 26 08              BNE    DECLN1
F72A 7C E5 71           INC    LINE
F72D BD F7 CB           JSR    SCROLD
```

```
F730 20 05             BRA    MOVEA
F732 86 17     DECLN1  LDA A  £23
F734 B7 E5 71          STA A  LINE
               * CURSOR RELATIVE ADDRESS
F737 F6 E5 72  MOVEA   LDA B  COLUMN     GET COLUMN NUMBER
F73A B6 E5 71          LDA A  LINE       GET LINE NUMBER
F73D 27 0D             BEQ    MOVE2      BRANCH IF ZERO
F73F B7 E5 79          STA A  TEMP1
F742 4F                CLR A
F743 CB 50     MOVE1   ADD B  £80
F745 89 00             ADC A  £0
F747 7A E5 79          DEC    TEMP1      DEC COUNT
F74A 26 F7             BNE    MOVE1      LOOP IF NOT ZERO
               * CURSOR ABSOLUTE ADDRESS
F74C FB E5 76  MOVE2   ADD B  DPAGE+1    ADD PAGE ADDRESS
F74F B9 E5 75          ADC A  DPAGE
F752 B7 E5 73          STA A  CURAD      SAVE CURRENT CURSOR ADDRESS
F755 F7 E5 74          STA B  CURAD+1
               * OUTPUT CURSOR TO CRTC
F758 C6 0E     OUTCUR  LDA B  £14
F75A F7 EB 70          STA B  CRTC
F75D B6 E5 73          LDA A  CURAD
F760 B7 EB 71          STA A  CRTC+1
F763 5C                INC B
F764 F7 EB 70          STA B  CRTC
F767 B6 E5 74          LDA A  CURAD+1
F76A B7 EB 71          STA A  CRTC+1
F76D 39                RTS
               *
               * SCROLL UP
               *
               * UPDATE END ADDRESS AND FILL WITH SPACE
F76E FE E5 77  SCROLL  LDX    EPAGE      X:= END OF PAGE
F771 8D 32             BSR    DELLIN     ERASE NEW LINE
F773 FF E5 77          STX    EPAGE      NEW END OF PAGE
F776 B6 E5 77          LDA A  EPAGE
F779 84 07             AND A  £7
F77B B7 E5 77          STA A  EPAGE
               * UPDATE START ADDRESS
F77E 4F                CLR A
F77F F6 E5 76          LDA B  DPAGE+1
F782 CB 50             ADD B  £80
F784 B9 E5 75          ADC A  DPAGE
F787 84 07             AND A  £7
F789 B7 E5 75          STA A  DPAGE
F78C F7 E5 76          STA B  DPAGE+1
               * UPDATE CRTC START ADDRESS
F78F C6 0C     UCRTC   LDA B  £12
```

```
F791 F7 E8 70          STA B  CRTC
F794 B6 E5 75          LDA A  DPAGE
F797 B7 E8 71          STA A  CRTC+1
F79A 5C                INC B
F79B F7 E8 70          STA B  CRTC
F79E B6 E5 76          LDA A  DPAGE+1
F7A1 B7 E8 71          STA A  CRTC+1

F7A4 39                RTS

              * DELETE LINE ROUTINE
F7A5 C6 50    DELLIN   LDA B  £80        80 CHAR. PER LINE
F7A7 08       DELLI1   INX
F7A8 FF E5 79          STX    TEMP1      SAVE X
F7AB FF E5 7B          STX    TEMP2
F7AE B6 E5 7B          LDA A  TEMP2
F7B1 84 07             AND A  £7         MODULO 2K
F7B3 8B EC             ADD A  £HADRF     PHYSICAL ADDRESS OF REFRESH MEMORY
F7B5 88 1C             EOR A  £$1C
F7B7 B7 E5 7B          STA A  TEMP2
F7BA FE E5 7B          LDX    TEMP2      ABSOLUTE ADDRESS
F7BD 86 20             LDA A  £$20
F7BF BA E5 7D          ORA A  VIDEO
F7C2 A7 00             STA A  0,X
F7C4 FE E5 79          LDX    TEMP1      RESTORE X
F7C7 5A                DEC B
F7C8 26 DD             BNE    DELLI1     LOOP IF B NOT ZERO
F7CA 39                RTS

              *
              * SCROLL DOWN
              *
F7CB B6 E5 75 SCROLD   LDA A  DPAGE
F7CE F6 E5 76          LDA B  DPAGE+1
F7D1 C0 50             SUB B  £80
F7D3 82 00             SBC A  £0
F7D5 84 07             AND A  £7
F7D7 B7 E5 75          STA A  DPAGE
F7DA F7 E5 76          STA B  DPAGE+1
F7DD FE E5 75          LDX    DPAGE
F7E0 09                DEX
F7E1 8D C2             BSR    DELLIN     ERASE NEW LINE

              * UPDATE END ADDRESS
F7E3 B6 E5 77          LDA A  EPAGE
F7E6 F6 E5 78          LDA B  EPAGE+1
F7E9 C0 50             SUB B  £80
F7EB 82 00             SBC A  £0
F7ED 84 07             AND A  £7
F7EF B7 E5 77          STA A  EPAGE
F7F2 F7 E5 78          STA B  EPAGE+1
F7F5 20 98             BRA    UCRTC

              *
              * OUTPUT CHARACTER CONTAINED IN REG-A
              *
```

```
2400 SCREEN HANDLER              9-12-81  ASSEMBLEUR TSC 6800   PAGE   17

 F7F7 FF E5 7F  OUTPUT  STX    XREG
 F7FA 36                PSH A
 F7FB 37                PSH B
 F7FC 7D E4 16          TST    ESCFG
 F7FF 26 50             BNE    ESCS
 F801 4D        OUTPT1  TST A
 F802 27 27             BEQ    CRTRET
 F804 81 9F             CMP A  £$9F
 F806 22 23             BHI    CRTRET
 F808 81 1F             CMP A  £$1F
 F80A 23 25             BLS    FUNCT
 F80C F6 E5 73          LDA B  CURAD
 F80F C4 07             AND B  £7
 F811 CB EC             ADD B  £HADRF
 F813 C8 1C             EOR B  £$1C
 F815 F7 E5 79          STA B  TEMP1
 F818 F6 E5 74          LDA B  CURAD+1
 F81B F7 E5 7A          STA B  TEMP1+1
 F81E FE E5 79          LDX    TEMP1
 F821 84 7F             AND A  £$7F
 F823 BA E5 70          ORA A  VIDEO
 F826 A7 00             STA A  0,X
 F828 BD F6 DF          JSR    INCCL
 F82B FE E5 7F  CRTRET  LDX    XREG
 F82E 33                PUL B
 F82F 32                PUL A
 F830 39                RTS

                *
                * FUNCTIONS
                *
 F831 CE F9 91  FUNCT   LDX    £FCTCRT
 F834 A1 00     FUNCT2  CMP A  0,X
 F836 27 0A             BEQ    FUNCT1
 F838 08                INX
 F839 08                INX
 F83A 08                INX
 F83B 8C F9 AC          CPX    £ENFCT
 F83E 26 F4             BNE    FUNCT2
 F840 20 E9             BRA    CRTRET

 F842 EE 01     FUNCT1  LDX    1,X
 F844 AD 00             JSR    0,X
 F846 20 E3             BRA    CRTRET

                *
                * ESCAPE SEQUENCE
                *
 F848 7C E4 17  CPOS5   INC    CPOSFG
 F84B 86 FF     ESCAPE  LDA A  £$FF
 F84D B7 E4 16          STA A  ESCFG
 F850 39                RTS
 F851 7D E4 17  ESCS    TST    CPOSFG
 F854 26 2D             BNE    CPOS
 F856 7F E4 16          CLR    ESCFG
 F859 CE F9 AC          LDX    £ESCTBL
```

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

F85C A1 00     ESCSQ   CMP A  0,X
F85E 27 E2             BEQ    FUNCT1
F860 08                INX
F861 08                INX
F862 08                INX
F863 8C F9 D6          CPX    £ENTBL
F866 26 F4             BNE    ESCSQ
F868 20 97             BRA    OUTPT1

                * CURSOR OFF
F86A 86 20     CUROFF  LDA A  £$20
F86C C6 0A     CDISP   LDA B  £10
F86E F7 E8 70          STA B  CRTC
F871 B7 E8 71          STA A  CRTC+1
F874 39                RTS

                * CURSOR ON
F875 86 61     CURON   LDA A  £$61
F877 20 F3             BRA    CDISP

                * PAGE MODE
F879 86 FF     PAGMOD  LDA A  £$FF
F87B B7 E4 18          STA A  PAGFG
F87E 39                RTS

                * ROLL MODE
F87F 7F E4 18  ROLMOD  CLR    PAGFG
F882 39                RTS

                * PUT CURSOR AT GIVEN LOCATION
F883 81 60     CPOS    CMP A  £96       OFFSET ?
F885 23 02             BLS    CPOS6     BRANCH IF NOT
F887 80 60             SUB A  £96
F889 4A        CPOS6   DEC A
F88A F6 E4 17          LDA B  CPOSFG
F88D C1 02             CMP B  £2
F88F 27 0E             BEQ    CPOS1
F891 81 17             CMP A  £23
F893 23 02             BLS    CPOS3
F895 86 17             LDA A  £23
F897 B7 E5 71  CPOS3   STA A  LINE
F89A 7C E4 17          INC    CPOSFG
F89D 20 12             BRA    CRTRT1
F89F 81 4F     CPOS1   CMP A  £79
F8A1 23 02             BLS    CPOS4
F8A3 86 4F             LDA A  £79
F8A5 B7 E5 72  CPOS4   STA A  COLUMN
F8A8 7F E4 1E          CLR    ESCFG
F8AB 7F E4 17          CLR    CPOSFG
F8AE BD F7 37          JSR    MOVEA
F8B1 7E F8 2B  CRTRT1  JMP    CRTRET

                * INVERSE VIDEO
F8B4 7F E5 70  INVERS  CLR    VIDEO
F8B7 39                RTS
F8B8 86 80     SINVER  LDA A  £$80
```

```
F8BA B7 E5 70           STA A  VIDEO
F8BD 39                 RTS

                * ECHO OFF
F8BE 86 FF      ECHOFF  LDA A  £$FF      SET NO ECHO FLAG
F8C0 B7 E4 08           STA A  OUTSW
F8C3 39                 RTS

                * ECHO ON
F8C4 7F E4 08   ECHON   CLR    OUTSW     SET ECHO FLAG
F8C7 39                 RTS

                * ERASE LINE
F8C8 B6 E5 73   ERLIN   LDA A  CURAD     BEGINNING ADDRESS OF LINE
F8CB F6 E5 74           LDA B  CURAD+1
F8CE F0 E5 72           SUB B  COLUMN
F8D1 82 00              SBC A  £0
F8D3 B7 E5 79           STA A  TEMP1
F8D6 F7 E5 7A           STA B  TEMP1+1
F8D9 FE E5 79           LDX    TEMP1
F8DC 09                 DEX
F8DD 7E F7 A5           JMP    DELLIN    GOTO ERASE LINE AND RETURN

                * ERASE TO END OF LINE
F8E0 C6 50      EREOL   LDA B  £80
F8E2 F0 E5 72           SUB B  COLUMN    NUMBER OF CHAR. TO ERASE
F8E5 FE E5 73           LDX    CURAD     CURSOR ADDRESS
F8E8 09                 DEX
F8E9 7E F7 A7           JMP    DELLI1    GOTO ERASE TO END OF LINE

                * ERASE TO END OF SCREEN
F8EC 8D F2      EREOS   BSR    EREOL     ERASE TO END OF LINE
F8EE 86 17              LDA A  £23
F8F0 B0 E5 71           SUB A  LINE      NUMBER OF LINE TO ERASE
F8F3 27 0B              BEQ    EREOS2
F8F5 36         EREOS1  PSH A
F8F6 FE E5 79           LDX    TEMP1
F8F9 BD F7 A5           JSR    DELLIN    ERASE LINE
F8FC 32                 PUL A
F8FD 4A                 DEC A
F8FE 26 F5              BNE    EREOS1    CONTINUE TILL NOT ZERO
F900 39         EREOS2  RTS              RETURN

                *
                * HARD COPY ON PARALLEL PRINTER
                *
F901 FE E5 75   HCPY    LDX    DPAGE     BEGINNING OF PAGE
F904 FF E5 79           STX    TEMP1
F907 86 18              LDA A  £24       NUMBER OF LINE
F909 B7 E5 7D           STA A  LINCNT
F90C 86 50      HCPY1   LDA A  £80       NUMBER OF COLUMN
F90E B7 E5 7E           STA A  COLCNT
F911 FF E5 7B   HCPY2   STX    TEMP2     LOGICAL ADDRESS
F914 B6 E5 7B           LDA A  TEMP2
F917 84 07              AND A  £7        MODULO 2K
F919 8B EC              ADD A  £HADRF    ADDRESS OF REFRESH MEMORY
```

```
F91B 88 1C                EOR A  £$1C
F91D B7 E5 7B             STA A  TEMP2     ADDRESS OF CHAR.
F920 FE E5 7B             LDX    TEMP2
F923 A6 00                LDA A  0,X       GET CHAR.
F925 84 7F                AND A  £$7F      MASK VIDEO
F927 81 1F                CMP A  £$1F      ASCII ?
F929 22 02                BHI    HCPY3     BRANCH IF YES
F92B 86 2D                LDA A  £'-       NOT ASCII
F92D BD FC 8F    HCPY3    JSR    POUT      OUTPUT CHAR.
F930 FE E5 79             LDX    TEMP1     INC LOGICAL ADDRESS
F933 08                   INX
F934 FF E5 79             STX    TEMP1
F937 7A E5 7E             DEC    COLCNT    DEC COLUMN COUNT
F93A 26 05                BNE    HCPY2     BRANCH IF NOT ZERO
F93C 86 0D                LDA A  £VCR      NEW LINE
F93E BD FC 8F             JSR    POUT      OUTPUT CR LF
F941 86 0A                LDA A  £VLF
F943 BD FC 8F             JSR    POUT
F946 7A E5 7D             DEC    LINCNT    DEC LINE COUNT
F949 26 C1                BNE    HCPY1     CONTINUE IF NOT ZERO
F94B 86 0C                LDA A  £12       OUTPUT FORM FEED
F94D BD FC 8F             JSR    POUT
F950 39                   RTS              RETURN

                 *
                 * VERSION
                 *
F951 CE F9 81    RESERV   LDX    £VERSIO
F954 FF E5 7B    RESV1    STX    TEMP2
F957 A6 00                LDA A  0,X
F959 81 04                CMP A  £4
F95B 27 23                BEQ    RESV2
F95D F6 E5 73             LDA B  CURAD
F960 C4 07                AND B  £7
F962 CB EC                ADD B  £HADRF
F964 C8 1C                EOR B  £$1C
F966 F7 E5 79             STA B  TEMP1
F969 F6 E5 74             LDA B  CURAD+1
F96C F7 E5 7A             STA B  TEMP1+1
F96F FE E5 79             LDX    TEMP1
F972 BA E5 72             ORA A  VIDEO
F975 A7 00                STA A  0,X
F977 BD F6 DF             JSR    INCOL
F97A FE E5 7B             LDX    TEMP2
F97D 08                   INX
F97E 20 D4                BRA    RESV1
F980 39          RESV2    RTS
F981 47          VERSIO   FCC    'GOUPIL 1.4 8/21'
F990 04                   FCB    4

                 *
                 * FUNCTIONS TABLE
                 *
F991 07          FCTCRT   FCB    $7        BELL
F992 F6 BE                FDB    BELL
F994 08                   FCB    $8        CURSOR LEFT
```

```
F995 F6 F2          FDB   DECCL
F997 1D             FCB   $1D      CURSOR RIGHT
F998 F6 DF          FDB   INCCL
F99A 0A             FCB   $A       CURSOR DOWN
F99B F7 00          FDB   INCLN
F99D 0B             FCB   $B       CURSOR UP
F99E F7 22          FDB   DECLN
F9A0 0C             FCB   $C       CLEAR
F9A1 F6 D5          FDB   CLEAR
F9A3 0D             FCB   $D       LINE RETURN
F9A4 F6 DA          FDB   LINRET
F9A6 1B             FCB   $1B      ESCAPE
F9A7 F8 4B          FDB   ESCAPE
F9A9 1C             FCB   $1C      HOME
F9AA F6 D7          FDB   HOME
F9AC         ENFCT  EQU   *

             *
             * ESCAPE SEQUENCE TABLE
             *
F9AC 49      ESCTBL FCC   'I' INVERSE VIDEO
F9AD F8 B4          FDB   INVERS
F9AF 4A             FCC   'J' NORMAL VIDEO
F9B0 F8 B3          FDB   SINVER
F9B2 50             FCC   'P' PAGE MODE
F9B3 F8 79          FDB   PAGMOD
F9B5 52             FCC   'R' ROLL MODE
F9B6 F8 7F          FDB   ROLMOD
F9B8 45             FCC   'E' CURSOR OFF
F9B9 F8 6A          FDB   CUROFF
F9BB 43             FCC   'C' CURSOR ON
F9BC F8 75          FDB   CURON
F9BE 3D             FCC   '=' MOVE CURSOR
F9BF F8 48          FDB   CPOSS
F9C1 48             FCC   'H' ECHO ON
F9C2 F8 C4          FDB   ECHON
F9C4 47             FCC   'G' ECHO OFF
F9C5 F8 BE          FDB   ECHOFF
F9C7 4C             FCC   'L' ERASE LINE
F9C8 F8 C8          FDB   ERLIN
F9CA 58             FCC   'X' ERASE TO END OF LINE
F9CB F8 E0          FDB   EREOL
F9CD 59             FCC   'Y' ERASE TO END OF PAGE
F9CE F8 EC          FDB   EREOS
F9D0 44             FCC   'D' HARD COPY
F9D1 F9 01          FDB   HCPY
F9D3 5A             FCC   'Z' RESERVED
F9D4 F9 51          FDB   RESERV
F9D6         ENTBL  EQU   *

             *
             * RESERVED  RAM
             *
E416                ORG   $E416
E416         ESCFG  RMB   1
E417         CPOSFG RMB   1
```

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

 E418          PAGFG   RMB    1
 E570                  ORG    $E570
 E570          VIDEO   RMB    1
 E571          LINE    RMB    1
 E572          COLUMN  RMB    1
 E573          CURAD   RMB    2
 E575          DPAGE   RMB    2
 E577          EPAGE   RMB    2
 E579          TEMP1   RMB    2
 E57B          TEMP2   RMB    2
 E57D          LINCNT  RMB    1
 E57E          COLCNT  RMB    1
 E57F          XREG    RMB    2
```

```
F9D6            ORG    $F9D6
                NAM    KEYBOARD HANDLER
                OPT    PAG

          **********************************
          *                                *
          *       KEYBOARD HANDLER         *
          *                                *
          **********************************


          *
          *     I/O ROUTINES
          *     ------------
          *
          *        INITCV    INITIALIZATIONS
          *        TINPUT    CHECK FOR TYPED CHARACTER
          *        INPUT     INPUT ONE CHARACTER
          *
          *
          * EQUATES
          *
E840      VIACL   EQU    $E840
E820      DATA1   EQU    $E820
E830      DATA2   EQU    $E830
E821      CMDE1   EQU    $E821
E831      CMDE2   EQU    $E831
E860      VIAMUS  EQU    $E860
EBF8      SETMEM  EQU    $EBF8
0000      ORB     EQU    0
0001      ORA     EQU    1
0002      DDRB    EQU    2
0003      DDRA    EQU    3
0004      TILL    EQU    4
000C      PCR     EQU    $C
000D      IFR     EQU    $D
000E      IER     EQU    $E
00A1      SHIFT   EQU    $A1      SHIFT CODE
00A0      REPT    EQU    $A0      REPEAT CODE
```

```
                        *
                        * KEYBOARD 1 TABLE DEFINITION
                        *
      F9D6              ADTAB1  EQU   *
      F9D6 1A                   FCB   $1A,$24,$05,$13,$8E,$03,$85,$18
      F9DE 12                   FCB   $12,$07,$14,$06,$88,$02,$87,$16
      F9E6 19                   FCB   $19,$0A,$15,$08,$8A,$9D,$89,$0E
      F9EE 09                   FCB   $09,$11,$01,$00,$84,$17,$F1,$A1
      F9F6 0B                   FCB   $0B,$0C,$0F,$0B,$8C,$9B,$8B,$9E
      F9FE 0A                   FCB   $0A,$0D,$00,$7F,$08,$00,$1F,$20
      FA06 10                   FCB   $10,$1C,$1D,$0D,$9C,$00,$00,$1E
      FA0E 00                   FCB   $00,$A0,$00,$00,$00,$00,$00,$00

      FA16 1A                   FCB   $1A,$24,$05,$13,$8E,$03,$85,$18
      FA1E 12                   FCB   $12,$07,$14,$06,$88,$02,$87,$16
      FA26 19                   FCB   $19,$0A,$15,$08,$8A,$9D,$89,$0E
      FA2E 09                   FCB   $09,$11,$01,$00,$84,$17,$F1,$A1
      FA36 0B                   FCB   $0B,$0C,$0F,$0B,$8C,$9B,$8B,$9E
      FA3E 0A                   FCB   $0A,$0D,$00,$7F,$08,$00,$1F,$20
      FA46 10                   FCB   $10,$1C,$1D,$0D,$9C,$00,$00,$1E
      FA4E 00                   FCB   $00,$A0,$00,$00,$00,$00,$00,$00

      FA56 5A                   FCB   $5A,$44,$45,$53,$33,$43,$32,$5B
      FA5E 52                   FCB   $52,$47,$54,$46,$35,$42,$34,$56
      FA66 59                   FCB   $59,$4A,$55,$48,$37,$3F,$36,$4E
      FA6E 03                   FCB   $03,$51,$41,$00,$31,$57,$F1,$A1
      FA76 4B                   FCB   $4B,$4C,$4F,$4B,$39,$2F,$38,$2E
      FA7E 0A                   FCB   $0A,$0D,$00,$7F,$08,$00,$5F,$20
      FA86 50                   FCB   $50,$25,$7E,$4D,$81,$00,$30,$2B
      FA8E 00                   FCB   $00,$A0,$00,$00,$00,$00,$00,$00

      FA96 7A                   FCB   $7A,$64,$65,$73,$22,$63,$7B,$7B
      FA9E 72                   FCB   $72,$67,$74,$66,$28,$62,$27,$76
      FAA6 79                   FCB   $79,$6A,$75,$68,$7D,$2C,$93,$6E
      FAAE 03                   FCB   $03,$71,$61,$00,$26,$77,$F1,$A1
      FAB6 6B                   FCB   $6B,$6C,$6F,$6B,$82,$3A,$21,$3B
      FABE 0A                   FCB   $0A,$0D,$00,$7F,$08,$00,$2D,$20
      FAC6 70                   FCB   $70,$7C,$5E,$6D,$29,$00,$40,$3D
      FACE 00                   FCB   $00,$A0,$00,$00,$00,$00,$00,$00
                        *
                        * KEYBOARD 2 TABLE DEFINITION
                        *
      FAD6              ADTAB2  EQU   *
      FAD6 24                   FCB   $24,$F2,$0B,$F7,$0C,$00,$0A,$F6
      FADE 22                   FCB   $00,$00,$00,$F5,$23,$00,$00,$00
      FAE6 5C                   FCB   $5C,$00,$20,$00,$00,$F4,$30,$00
      FAEE 3D                   FCB   $3C,$00,$31,$34,$30,$F3,$2E,$37
      FAF6 25                   FCB   $25,$32,$2D,$2A,$10,$00,$00,$2F
      FAFE 5B                   FCB   $5B,$2F,$00,$F9,$1D,$00,$00,$F8
      FB06 2A                   FCB   $2A,$5D,$33,$3E,$00,$00,$00,$39
      FB0E 23                   FCB   $23,$13,$32,$35,$00,$00,$00,$38
```

```
                     * I/O INITIALIZATIONS
                     *************
                     *
                     *
FB16 BD F6 71  INIT    JSR    INITVS    INITIALIZE SCREEN
FB19 86 01             LDA A  £1
FB1B B7 EB F8          STA A  SETMEM
                     *
                     * VIA FOR BELL
FB1E 86 7F     INITCV  LDA A  £$7F
FB20 B7 E8 6E          STA A  VIAMUS+IER
FB23 86 FF             LDA A  £$FF
FB25 B7 E8 6C          STA A  VIAMUS+PCR
FB28 B7 E8 63          STA A  VIAMUS+DDRA
                     *
                     * 8279
                     *
                     * ENCODED KEYBOARD N-KEY ROLLOVER
                     *
FB2B 86 02     INITB2  LDA A  £2
FB2D B7 E8 21          STA A  CMDE1
FB30 B7 E8 31          STA A  CMDE2
FB33 86 C1             LDA A  £$C1
FB35 B7 E8 21          STA A  CMDE1
FB38 B7 E8 31          STA A  CMDE2
FB3B 86 2F             LDA A  £$2F
FB3D B7 E8 21          STA A  CMDE1
FB40 B7 E8 31          STA A  CMDE2
FB43 86 A0             LDA A  £$A0
FB45 B7 E8 21          STA A  CMDE1
                     *
                     * PARALLEL I/O PORT
FB48 4F        INITVC  CLR A
FB49 B7 E8 43          STA A  VIACL+DDRA
FB4C 43                COM A
FB4D B7 E8 42          STA A  VIACL+DDRB
FB50 86 C3             LDA A  £$C3
FB52 B7 E8 4C          STA A  VIACL+PCR
FB55 86 E3             LDA A  £$E3
FB57 B7 E8 4C          STA A  VIACL+PCR
                     *
                     * WAIT FOR KEYBOARD
FB5A C6 0D             LDA B  £13
FB5C 86 C2     STAB1   LDA A  £$C2
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
```

```
                    *
                    * GET THE CHARACTER TYPED ON KEYBOARD
                    *
FB6E B6 E4 C2  INPUT   LDA A  SAVEA     GET CHARACTER
FB71 7F E4 13          CLR    FLGIN     CLEAR FLAG
FB74 39                RTS              RETURN
                    *
                    * REPEAT PROCESSING
                    *
FB75 C6 47     REPTA   LDA B  £$47      READ FIFO SENSOR RAM AT ROW 7
FB77 F7 E8 21          STA B  CMDE1
FB7A CE 03 00          LDX    £$300     DELAY AFTER COMMAND
FB7D 09        REPTA4  DEX
FB7E 26 FD             BNE    REPTA4
FB80 F6 E8 20          LDA B  DATA1     GET DATA
FB83 C4 02             AND B  £2        CHECK FOR REPEAT CODE
FB85 27 0B             BEQ    REPTA2    NO MORE REPEAT
FB87 CE 20 00          LDX    £$2000    DELAY
FB8A 09        REPTA1  DEX
FB8B 26 FD             BNE    REPTA1
FB8D 73 E4 13          COM    FLGIN     SET INPUT FLAG
FB90 20 08             BRA    REPTA3
FB92 C6 02     REPTA2  LDA B  £2        SET ENCODED SCAN KEYBOARD
FB94 F7 E8 21          STA B  CMDE1
FB97 7F E4 12          CLR    FLGRPT    CLEAR REPEAT FLAG
FB9A C6 C1     REPTA3  LDA B  £$C1      CLEAR INTEL 8279
FB9C F7 E8 21          STA B  CMDE1
FB9F 20 11             BRA    TINPT1    RETURN
                    *
                    * CHECK IF A CHARACTER HAS BEEN TYPED
                    *
FBA1 7D E4 13  TINPUT  TST    FLGIN     CHARACTER ALREADY PRESENT
FBA4 26 14             BNE    TINPT2    YES RETURN
FBA6 FF E4 C3          STX    SAVEX     SAVE X
FBA9 36                PSH A            SAVE A
FBAA 37                PSH B            SAVE B
FBAB 7D E4 12          TST    FLGRPT    REPEAT ?
FBAE 26 C5             BNE    REPTA     YES GO PROCESS REPEAT
FBB0 8D 09             BSR    KEYB      CHECK AND READ KEYBOARD
FBB2 FE E4 C3  TINPT1  LDX    SAVEX     RESTORE X
FBB5 7D E4 13          TST    FLGIN     SET Z
FBB8 33                PUL B            RESTORE B
FBB9 32                PUL A            RESTORE A
FBBA 39        TINPT2  RTS              RETURN
                    *
                    * CHECK AND READ KEYBOARD
                    *
FBBB B6 E8 21  KEYB    LDA A  CMDE1     CHECK KEYBOARD 1
FBBE 84 0F             AND A  £$F
FBC0 26 11             BNE    KEYB1     WE HAVE CHAR.
FBC2 B6 E8 31          LDA A  CMDE2     CHECK KEYBOARD 2
FBC5 84 0F             AND A  £$F
FBC7 27 3B             BEQ    KEYB4     NOTHING,RETURN
FBC9 B6 E8 30          LDA A  DATA2     GET CHAR.
FBCC CE FA D6          LDX    £ADTAB2   ADDRESS OF TABLE 2
```

```
FBCF 84 3F              AND A  £$3F       ONLY 6 BITS
FBD1 20 0D              BRA    KEYB2
FBD3 B6 E8 20  KEYB1    LDA A  DATA1      GET CHAR.
FBD6 CE F9 D6           LDX    £ADTAB1    ADDRESS OF TABLE 1
FBD9 7D E4 11           TST    FLGSHF     CHECK SHIFT LOCK
FBDC 26 02              BNE    KEYB2
FBDE 84 BF              AND A  £$BF       IF LOCKED,MASK THE BIT
FBE0 FF E4 C0  KEYB2    STX    PTTAB
FBE3 BB E4 C1           ADD A  PTTAB+1    ADDRESS OF CHAR. IN THE TABLE
FBE6 24 03              BCC    KEYB3
FBE8 7C E4 C0           INC    PTTAB      ADD CARRY IF ANY
FBEB B7 E4 C1  KEYB3    STA A  PTTAB+1
FBEE FE E4 C0           LDX    PTTAB      GET ADDRESS OF CHAR.
FBF1 A6 00              LDA A  0,X        GET CODE
FBF3 B7 E4 C2           STA A  SAVEA      SAVE IT
FBF6 81 A1              CMP A  £SHIFT     SHIFT KEY ?
FBF8 27 08              BEQ    GSHIFT     YES GO PROCESS IT
FBFA 81 A0              CMP A  £REPT      REPEAT KEY ?
FBFC 27 15              BEQ    GREPT      IF YES GO PROCESS IT
FBFE 73 E4 13           COM    FLGIN      SET INPUT FLAG
FC01 39        KEYB4    RTS               RETURN
               *
               * SHIFT LOCK PROCESSING
               *
FC02 7D E4 11  GSHIFT   TST    FLGSHF     CHECK FLAG
FC05 27 03              BEQ    GSHFT1     SKIP IF LOCKED
FC07 86 A0              LDA A  £$A0       LIGHT ON
FC09 8C                 FCB    SKIP2
FC0A 86 A3     GSHFT1   LDA A  £$A3       LIGHT OFF
FC0C B7 E8 21           STA A  CMDE1
FC0F 73 E4 11           COM    FLGSHF     SWITCH THE FLAG
FC12 39                 RTS               RETURN
               *
               * FIRST TIME REPEAT HAS BEEN TYPED
               *
FC13 8D A6     GREPT    BSR    KEYB       CHECK AND READ KEYBOARD
FC15 7D E4 13           TST    FLGIN      CHAR. TYPED ?
FC18 27 F9              BEQ    GREPT      NO,WAIT
FC1A C6 04              LDA B  £4         YES SET ENCODED SCAN SENSOR MATRIX
FC1C F7 E8 21           STA B  CMDE1
FC1F 86 FF              LDA A  £$FF       SET REPEAT FLAG
FC21 B7 E4 12           STA A  FLGRPT
FC24 39                 RTS               RETURN
               *
               * RAM RESERVED TO VARIABLE INFORMATIONS
               *
E411                    ORG    $E411      INITIALIZED TO ZERO AT START
E411           FLGSHF   RMB    1
E412           FLGRPT   RMB    1
E413           FLGIN    RMB    1
E4C0                    ORG    $E4C0
E4C0           PTTAB    RMB    2
E4C2           SAVEA    RMB    1
E4C3           SAVEX    RMB    2
E4C5           VTEMP    RMB    2
```



```
                        END
```

```
                   NAM    GPMON V1.4
                   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    $B         VIA AUXILIARY CONTROL REGISTER
0004        TILL   EQU    4          VIA T1 LOW ORDER LATCH
            *
            * INITIALIZE ACIA (CPU)
            *    A:=ACIA COMMAND
            *    X:=CLOCK VIA
            *
FC25 C6 03  INICIA LDA B  £3         MASTER RESET CODE
FC27 F7 E8 0C      STA B  ACIAC      RESET ACIA
FC2A B7 E8 0C      STA A  ACIAC      INITIALIZE ACIA
FC2D 86 C0         LDA A  £$C0       INZ CLOCK FOR ACIA
FC2F B7 E8 1B      STA A  VIAC+ACR
FC32 FF E8 14      STX    VIAC+TILL  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 08  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
FC50 85 01         BIT A  £1         SHOW READY OR NOT READY
FC52 39            RTS               RETURN
```

```
                   *
                   * PROGRAM STARTS HERE
                   *
FC53 BD FD CC COM      JSR    PCRLF
FC56 CE FC 8C          LDX    £QUEST    ASK FOR SPEED
FC59 BD FD 91          JSR    PDATA1
FC5C BD FD 98          JSR    INCH      INPUT SPEED
FC5F 16                TAB              SAVE IT IN B
FC60 BD FD CC          JSR    PCRLF     OUTPUT CR/LF
FC63 C1 31             CMP B  £$31
FC65 27 09             BEQ    COM1      300BDS
FC67 C1 32             CMP B  £$32
FC69 26 E8             BNE    COM       NOT CORRECT
FC6B CE 18 00          LDX    £$1800    1200 BDS
FC6E 20 03             BRA    COM2
FC70 CE 64 00 COM1     LDX    £$6400    300 BDS
FC73 86 01    COM2     LDA A  £1        EVEN PARITY,2 STOP BITS,CLOCK/16
FC75 8D AE             BSR    INICIA    INITIALIZE ACIA
                   *
FC77 BD FB A1 TERM     JSR    CHKCHR    CHECK FOR TYPED CHAR.
FC7A 27 05             BEQ    TERM1     NO CHAR.
FC7C BD FB EE          JSR    INPUT     INPUT CHAR.
FC7F 8D B5             BSR    TOACIA    XMIT CHAR.
                   *
FC81 8D CA    TERM1    BSR    CHKACI    CHECK FOR RECEIVED CHAR.
FC83 27 F2             BEQ    TERM      NO CHAR.
FC85 8D BC             BSR    INACIA    GET RECEIVED CHAR.
FC87 BD FD 82          JSR    OUTCH     OUTPUT IT
FC8A 20 EB             BRA    TERM      CONTINUE
                   *
FC8C 56       QUEST    FCC    'V='
FC8E 04                FCB    4
```

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

```
                    *********************************
                    *                               *
                    *    PARALLEL PRINTER HANDLER   *
                    *                               *
                    *********************************


                    *
                    * OUTPUT ONE CHARACTER TO PRINTER
                    *
FC8F 37             POUT   PSH B          SAVE B
FC90 F6 E8 40       CPOUT  LDA B  VIACL+IFR WAIT FOR READY
FC93 C5 10                 BIT B  £$10
FC95 27 F9                 BEQ    CPOUT
FC97 B7 E8 40              STA A  VIACL+ORB OUTPUT CHAR.
FC9A C6 C3                 LDA B  £$C3
FC9C F7 E8 4C              STA B  VIACL+PCR HANDSHAKE
FC9F C6 E3                 LDA B  £$E3
FCA1 F7 E8 4C              STA B  VIACL+PCR
FCA4 33                    PUL B          RESTORE B
FCA5 39                    RTS            RETURN
```

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

```
               ************************
               *                      *
               *       MONITOR        *
               *                      *
               ************************


               *
               * EQUATES
               *
FCA6           ROM     EQU   $FCA6
EA00           RAM     EQU   $E400
003F           SWI     EQU   $3F         SWI OP CODE
0030           SZRCL   EQU   SP-NID-2    SIZE OF RAM TO CLEAR
000D           CR      EQU   $D          CARRIAGE RETURN
000A           LF      EQU   $A          LINE FEED
002B           PRPT    EQU   '+          PROMPT CHARACTER
002E           DOWN    EQU   '.          MEMORY INCREMENT CHAR.
002D           UP      EQU   '-          MEMORY DECREMENT CHAR.
               *
EC00           K7MOD   EQU   $EC00       K7/MODEM HANDLER ADDRESS
8000           BASIC   EQU   $8000       ROM BASIC EXECUTION ADDRESS
FBA1           CHKCHR  EQU   TINPUT
```

```
FCA6                    ORG    ROM
                *
                * JUMP TABLE TO ROUTINES PERFORMING GPMON FUNCTIONS
                *
FCA6            FCTABL  EQU    *
FCA6 4D                 FCC    /M/ -M- MEMORY CHANGE
FCA7 FD ED              FDB    CHANGE
FCA9 47                 FCC    /G/ -G- GOTO ENTERED ADDRESS
FCAA FE 37              FDB    GOTO
FCAC 52                 FCC    /R/ -R- PRINT STACK
FCAD FF 1E              FDB    PSTAK
FCAF 56                 FCC    /V/ -V- SET A BREAKPOINT
FCB0 FE 2A              FDB    SETBRK
FCB2 55                 FCC    /U/ -U- RESET A BREAKPOINT
FCB3 FE 1E              FDB    RSTBRK
FCB5 57                 FCC    /W/ -W- DELETE ALL BREAKPOINTS
FCB6 FE 26              FDB    DELBRK
FCB8 43                 FCC    /C/ -C- CONTINUE
FCB9 FE 43              FDB    CONT
FCBB 4E                 FCC    /N/ -N- NEXT (TRACE 1 INSTRUCTION)
FCBC FE 4B              FDB    NEXT
FCBE 54                 FCC    /T/ -T- TRACE N INSTRUCTIONS
FCBF FE 4F              FDB    TRACE
FCC1 42                 FCC    /B/ -B- PRINT ALL BREAKS
FCC2 FE 23              FDB    PNTBRK
FCC4 44                 FCC    /D/ -D- MEMORY DUMP
FCC5 FE 65              FDB    DUMP
FCC7 F3                 FCB    $F3       -BOOT- EXECUTE DISK BOOTSTRAP
FCC8 F5 CC              FDB    DKBOOT
FCCA 4B                 FCC    /K/ -K- LINK TO K7/MODEM HANDLER
FCCB FD 37              FDB    WHAT
FCCD 4F                 FCC    /O/ -O- SWITCH OUTPUT TO PRINTER
FCCE FE 8C              FDB    SWITCH
FCD0 F1                 FCB    $F1       -BASIC- EXECUTE ROM BASIC
FCD1 FD 37              FDB    WHAT
FCD3 F2                 FCB    $F2       -COM- TERMINAL MODE
FCD4 FC 53              FDB    COM
FCD6 F4                 FCB    $F4       -GR- GRAPHIC MODE
FCD7 FD 37              FDB    WHAT
FCD9            FCTBEN  EQU    *
```

```
                         *
                         * 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 38               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,$14 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 8E E4 34            LDS    SP
FD0F BD FB 16            JSR    INIT       INITIALIZE I/O
```

```
                        *
                        * MAIN COMMAND/CONTROL LOOP
                        *
FD12                    CONTRL  EQU    *
FD12 BE E4 34                   LDS    SP        RESTORE STACK POINTER
FD15 CE FF 06                   LDX    £SWI1S    RESTORE SWI VECTOR
FD18 FF E4 36                   STX    SWI1
FD1B BD FD D4                   JSR    PROMPT    PRINT PROMPT
FD1E 7F E4 0E                   CLR    OUTSW     MAKE SURE INPUT IS ECHOED
FD21 BD FD 98                   JSR    INCH      READ COMMAND CHARACTER
FD24 16                         TAB              SAVE IT IN B
FD25 BD FD B3                   JSR    OUTS      PRINT SPACE AFTER COMMAND
                        *
                        * B REGISTER HOLDS CHARACTER INPUT BY USER
                        * USE JUMP TABLE TO GO TO APPROPRIATE ROUTINE
                        *
FD28 CE FC A6                   LDX    £FCTABL   X:= ADDRESS OF JUMP TABLE
FD2B E1 00              NXTCHR  CMP B  0,X       DOES INPUT CHAR. MATCH ?
FD2D 27 0F                      BEQ    GOODCH    YES,GO TO APPROPRIATE ROUTINE
FD2F 08                         INX              NO UPDATE INDEX
FD30 08                         INX
FD31 08                         INX
FD32 8C FC D9                   CPX    £FCTBEN   END OF TABLE REACHED ?
FD35 26 F4                      BNE    NXTCHR    NO,TRY NEXT COMMAND
FD37 86 3F              WHAT    LDA A  £'?       NO MATCH PRINT "?"
FD39 BD FD 82                   JSR    OUTCH
FD3C 20 D4                      BRA    CONTRL    REPROMPT USER
                        *
FD3E EE 01              GOODCH  LDX    1,X       GET ADDRESS FROM J.T.
FD40 EE 00                      JMP    0,X       GO TO APPROPRIATE ROUTINE
```

```
FD42                    ORG    $FD42
                 *
                 * BASIC ROUTINES
                 *

                 *
                 * INPUT HEX CHARACTER
                 *
FD42 8D 54       INHEX   BSR    INCH
FD44 80 30       INHEX2  SUB A  £$30
FD46 2B CA               BMI    CONTRL      NOT HEX
FD48 81 09               CMP A  £$9
FD4A 2F 0A               BLE    IN1HG
FD4C 81 11               CMP A  £$11
FD4E 2B C2               BMI    CONTRL      NOT HEX
FD50 81 16               CMP A  £$16
FD52 2E BE               BGT    CONTRL      NOT HEX
FD54 80 07               SUB A  £7
FD56 39          IN1HG   RTS
                 *
                 * BUILD ADDRESS
                 *
FD57 8D 10       BADDR   BSR    BYTE        READ 2 CHAR.
FD59 B7 E4 0A            STA A  XHI
FD5C 8D 0B               BSR    BYTE        READ 2 CHAR.
FD5E B7 E4 0B            STA A  XLOW
FD61 FE E4 0A            LDX    XHI         X:= ADDRESS WE BUILT
FD64 39                  RTS
FD65 8D F0       BADDRS  BSR    BADDR       BUILD ADDRESS
FD67 20 4A               BRA    OUTS        PRINT SPACE AND RETURN
                 *
                 * INPUT BYTE
                 *
FD69 8D D7       BYTE    BSR    INHEX       GET HEX CHAR.
FD6B 48          BYTE2   ASL A
FD6C 48                  ASL A
FD6D 48                  ASL A
FD6E 48                  ASL A
FD6F 16                  TAB
FD70 8D D0               BSR    INHEX       GET HEX CHAR.
FD72 1B                  ABA
FD73 39                  RTS
                 *
                 * OUT HEXA BCD DIGIT
                 *
FD74 44          OUTHL   LSR A              OUT HEX LEFT BCD DIGIT
FD75 44                  LSR A
FD76 44                  LSR A
FD77 44                  LSR A
FD78 84 0F       OUTHR   AND A  £$F         OUT HEX RIGHT BCD DIGIT
FD7A 8B 30               ADD A  £$30
FD7C 81 39               CMP A  £$39
FD7E 23 02               BLS    OUTCH
FD80 8B 07               ADD A  £7
                 *
```

```
                       * OUTPUT ONE CHARACTER
                       *
FD82 BD F7 F7   OUTCH    JSR    OUTPUT    OUTPUT CHAR. TO SCREEN
FD85 7D E4 09            TST    PRTFLG    TO PRINTER ?
FD88 26 01               BNE    POUTS     IF YES PRINT CHAR.
FD8A 39                  RTS              RETURN
FD8B 7E FC 8F   POUTS    JMP    POUT
                       *
                       * PRINT DATA POINTED BY X-REG
                       *
FD8E 8D F2      PDATA2   BSR    OUTCH     OUTPUT DATA
FD90 08         PDATA3   INX
FD91 A6 00      PDATA1   LDA A  X
FD93 81 04               CMP A  £4        STOP IF EOT
FD95 26 F7               BNE    PDATA2
FD97 39                  RTS              RETURN
                       *
                       * INPUT ONE CHARACTER
                       *
FD98 BD FB A1   INCH     JSR    TINPUT    CHECK FOR TYPED CHAR
FD9B 27 FB               BEQ    INCH      NOT READY
FD9D BD FB 6E            JSR    INPUT     INPUT CHAR.
FDA0 7D E4 02            TST    OUTSW     SHOULD INPUT BE ECHOED?
FDA3 27 DD               BEQ    OUTCH     IF SO ,OUTPUT CHAR.
FDA5 39                  RTS
                       *
                       * OUTPUT BYTE
                       *
FDA6 A6 00      OUT2H    LDA A  0,X       OUTPUT 2 HEX CHAR.
FDA8 08                  INX
FDA9 36         OUT2HA   PSH A
FDAA 8D C8               BSR    OUTHL     OUT LEFT HEX CHAR.
FDAC 32                  PUL A
FDAD 20 C9               BRA    OUTHR     OUTPUT RIGHT HEX CHAR AND RTS
                       *
                       * OUTPUT ADDRESS
                       *
FDAF 8D F5      OUT4HS   BSR    OUT2H     OUTPUT 4 HEX CHAR.+SPACE
FDB1 8D F3      OUT2HS   BSR    OUT2H     OUTPUT 2 HEX CHAR.+SPACE
FDB3 86 20      OUTS     LDA A  £$20      OUTPUT SPACE
FDB5 20 CB               BRA    OUTCH     (BSR+RTS)
                       *
                       * PRINT STACK CONTENTS
                       *
FDB7 8D 13      PRINT    BSR    PCRLF     OUTPUT CR/LF
FDB9 FE E4 34            LDX    SP        PRINT OUT STACK
FDBC 08                  INX
FDBD 8D F2               BSR    OUT2HS    CC-REG
FDBF 8D F0               BSR    OUT2HS    ACC-B
FDC1 8D EE               BSR    OUT2HS    ACC-A
FDC3 8D EA               BSR    OUT4HS    X-REG
FDC5 8D E8               BSR    OUT4HS    P-COUNTER
FDC7 CE E4 34            LDX    £SP
FDCA 20 E3               BRA    OUT4HS    PRINT SP AND RETURN
                       *
                       * OUTPUT CR/LF
```

```
                *
FDCC 86 0D      PCRLF   LDA A  £CR       OUTPUT CR
FDCE 8D B2              BSR    OUTCH
FDD0 86 0A              LDA A  £LF       OUTPUT LF
FDD2 20 AE              BRA    OUTCH     AND RETURN
                *
                * OUTPUT PROMPT
                *
FDD4 8D F6      PROMPT  BSR    PCRLF     DO CR/LF
FDD6 86 2B              LDA A  £PRPT
FDD8 20 A8              BRA    OUTCH     OUTPUT PROMPT AND RETURN
                *
                * ERROR : VALUE IN RAM DO NOT MATCH
                *
FDDA 16         ERROR   TAB              SAVE A
FDDB 8D EF              BSR    PCRLF
FDDD CE E4 0A           LDX    £XHI
FDE0 8D CD              BSR    OUT4HS    PRINT ADDRESS
FDE2 FE E4 0A           LDX    XHI
FDE5 8D CA              BSR    OUT2HS    PRINT CONTENT
FDE7 17                 TBA              RESTORE A
FDE8 8D BF              BSR    OUT2HA    PRINT VALUE WE WANT TO STORE
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
FE2E 8D 63                  BSR     BRKSUB
FE30 86 05                  LDA A   £BKFRCD   SET ONE BREAKPOINT FLAG
```

```
FE32 8D 5F    SETB2  BSR    BRKSUB    BREAK HANDLING SUBROUTINE
FE34 7E FD 12 CTRLJ  JMP    CONTRL    RETURN TO COMMAND LEVEL
                     *
                     * -G- COMMAND
                     *   GO TO ENTERED ADDRESS
                     *
FE37 8D 29    GOTO   BSR    BADDRJ    GET ADDR. FROM USER IN XHI,XLOW
FE39 FE E4 34        LDX    SP        GET VALUE SAVED WHEN ENTERING GPMON
FE3C A7 07           STA A  7,X       A-REG CONTAINS XLOW
FE3E B6 E4 0A        LDA A  XHI
FE41 A7 06           STA A  6,X       PLACE ADDRESS ON STACK
                     *
                     * -C- COMMAND
                     *   CONTINUE EXECUTION
                     *
FE43 7C E4 10 CONT   INC    BRKTRC    TRACE 1 TO RESTORE SWI'S
FE46 CE 00 01        LDX    £1
FE49 20 09           BRA    TRACE3
                     *
                     * -N- COMMAND
                     *   SINGLE INSTRUCTION TRACE REQUESTED
                     *
FE4B CE 00 01 NEXT   LDX    £1        £INSTRUCTION TO TRACE
FE4E 8C              FCB    SKIP2     SKIP 2 BYTES
                     *
                     * -T- COMMAND
                     *   MULTIPLE INSTRUCTION TRACE
                     *
FE4F 8D 11    TRACE  BSR    BADDRJ    GET N. OF INSTRUCTION TO TRACE
FE51 7F E4 10 TRACE1 CLR    BRKTRC    CLEAR FLAG INDICATING TRACE IS DUE TO BREAK
FE54 FF E4 06 TRACE3 STX    NTRACE    SAVE £ INST'S TO TRACE
FE57 FE E4 34        LDX    SP        X:=STACK POINTER
FE5A EE 06           LDX    6,X       X:=ADDR OF INSTR TO BE EXECUTED
FE5C BD FF 9A        JSR    SAVTRC    SAVE ADDR/OPCODE FOR TRACE
FE5F 7E FF 47        JMP    CONTRC    GOTO CONTINUE TRACE
FE62 7E FD 57 BADDRJ JMP    BADDR
                     *
                     * -D- COMMAND
                     *   MEMORY DUMP
                     *
FE65 BD FD 65 DUMP   JSR    BADDRS    GET START ADDRESS
FE68 FF E4 0C        STX    XSAVE     SAVE IT
FE6B 8D F5           BSR    BADDRJ    GET END ADDRESS IN XHI,XLOW
FE6D BD FD CC DUMP1  JSR    PCRLF     OUTPUT CR,LF
FE70 C6 10           LDA B  £16       16 BYTES PER LINE
FE72 CE E4 0C        LDX    £XSAVE    PRINT ADDRESS
FE75 BD FD AF        JSR    OUT4HS
FE78 FE E4 0C DUMP2  LDX    XSAVE     PRINT BYTE
FE7B BD FD B1        JSR    OUT2HS
FE7E FF E4 0C        STX    XSAVE     SAVE NEW ADDRESS
FE81 09              DEX
FE82 BC E4 0A        CPX    XHI       FINISHED ?
FE85 27 AD           BEQ    CTRLJ     YES,EXIT
FE87 5A              DEC B            DEC THE COUNT
FE88 26 EE           BNE    DUMP2     SAME LINE IF NOT 0
FE8A 20 E1           BRA    DUMP1     NEW LINE
```



```
                     *
                     * -O- COMMAND
                     *   SWITCH OUTPUT TO PRINTER
                     *
FE8C 73 E4 09 SWITCH COM    PRTFLG    SWITCH FLAG
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
0082                 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 02                   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
```

```
FEBB 35                 PSH A
FEBC 86 3F              LDA A  #SWI
FEBE A7 00              STA A  X          WRITE SWI
FEC0 32                 PUL A
FEC1 FE E4 0C           LDX    XSAVE      ADDR OF BREAK IN BREAK TABLE
FEC4 A7 02              STA A  2,X        SAVE USER CODE IN BREAK TABLE
                   *
                   * RECORD BREAKPOINT IN TABLE
                   *
FEC6 FE E4 0C      BRKP3  LDX    XSAVE     GET ADDRESS OF BREAK IN TABLE
FEC9 54                   LSR B            NEXT FUNCTION
FECA 24 0E                BCC    BRKP4
FECC B6 E4 0A             LDA A  XHI       GET HI-BYTE OF NEW ADDRESS
FECF A7 00                STA A  0,X       WRITE IN TABLE
FED1 B6 E4 0B             LDA A  XLOW      GET LOW BYTE
FED4 A7 01                STA A  1,X
FED6 32                   PUL A            GET FUNCTION NUMBER FROM STACK
FED7 84 FB                AND A  #%11111011 DO NOT PLACE ADDRESS MORE THAN ONCE
FED9 36                   PSH A            CONTINUE TO TAKE OUT BREAKPOINTS
                   *
                   * PURGE BREAKPOINT WHOSE ADDRESS IS IN XHI,XLOW
                   *
FEDA 54            BRKP4  LSR B
FEDB 24 0E                BCC    BRKP5
FEDD A6 00                LDA A  X         GET HI-BYTE FROM TABLE
FEDF B1 E4 0A             CMP A  XHI       COMPARE
FEE2 26 07                BNE    BRKP5     NOT THIS BREAK TO BE PURGED
FEE4 A6 01                LDA A  1,X       GET LOW BYTE
FEE6 B1 E4 0B             CMP A  XLOW
FEE9 27 03                BEQ    BRKP52    THIS BREAK IS TO BE PURGED
                   *
                   * PURGE ALL BREAKPOINTS
                   *
FEEB 54            BRKP5  LSR B
FEEC 24 04                BCC    BRKP6
FEEE 6F 00         BRKP52 CLR    0,X       CLEAR BREAKPOINT HIGH
FEF0 6F 01                CLR    1,X       CLEAR LOW BYTE
                   *
                   * PRINT BREAKPOINT
                   *
FEF2 54            BRKP6  LSR B
FEF3 24 04                BCC    BRKPE
FEF5 BD FC AF             JSR    OUT4HS    PRINT BREAKPOINT ADDRESS
FEF8 8C                   FCB    SKIP2     SKIP 2 BYTES
                   *
                   * UPDATE LOOP INDEX AND LOOP IF APPROPRIATE
                   *
FEF9 08            BRKPE  INX
FEFA 08                   INX
FEFB 08            BRKPE1 INX              X POINTS TO NEXT ADDR IN TABLE
FEFC 8C E4 34             CPX    #BRKINS
FEFF 26 B5                BNE    BRKP0
                   *
                   * WRAP-UP PROCESSING AND EXIT
                   *
FF01 F7 E4 0F             STA B  BRKSIN    STORE APPROPRIATE FLAG
```



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

```
                     *
                     * SWI SOFTWARE INTERRUPT PROCESSING
                     *
FF06                 SWI15   EQU   *
FF06 BF E4 34                STS   SP         SAVE USER'S SP
FF09 86 01                   LDA A £BKFRMV
FF0B BD FE 93                JSR   BRKSUB     GO TAKE OUT ALL BREAKS
                     * DECREMENT P-COUNTER
FF0E 30                      TSX              X:=STACK POINTER-1
FF0F 6D 06                   TST   6,X        IF LOWER BYTE=0 : BORROW
FF11 26 02                   BNE   SWI1S1     BRANCH IF BORROW NOT REQUIRED
FF13 6A 05                   DEC   5,X        DECREMENT UPPER BYTE
FF15 6A 06           SWI1S1  DEC   6,X        DECREMENT LOWER BYTE
                     * TEST FOR ADDRESS TRACE OR BREAK
FF17 EE 05                   LDX   5,X        X:=P-COUNTER
FF19 BC E4 04                CPX   TRCADR     IS SWI FOR TRACE ?
FF1C 27 06                   BEQ   TRCINH     YES,GO TO TRACE INT HANDLER
                     *
                     * BREAK INTERRUPT HANDLER
                     *
                     *
                     * -R- COMMAND
                     *
FF1E                 PSTAK   EQU   *
FF1E BD FD B7                JSR   PRINT      STOP AND SHOW REGISTER TO USER
FF21 7E FD 12        CNTRL3  JMP   CONTRL     RETURN
                     *
                     * TRACE INTERRUPT HANDLER
                     * P-COUNTER HAS BEEN DECREMENTED TO POINT AT SWI
                     * TRCINS HOLDS OP CODE REPLACED BY SWI
                     * X HOLDS ADDRESS OF WHERE TRACE SWI IS
                     *
FF24 B6 E4 0E        TRCINH  LDA A TRCINS     GET OPCODE OF TRACED INSTR.
FF27 A7 00                   STA A 0,X        RESTORE TO USER'CODE
FF29 7D E4 10                TST   BRKTRC     IS PROCESSING TO BE IMMEDIATELY CONTINUED ?
FF2C 27 0D                   BEQ   NBKTRC     BRANCH IF NOT
                     *
                     * PROCESSING IS TO "CONTINUE"
                     *
FF2E 7F E4 10                CLR   BRKTRC     RESET CONTINUE FLAG
FF31 BD FE 91                JSR   INSBRK     INSERT BREAKS IN USER CODE
FF34 7F E4 04                CLR   TRCADR     NO MORE TRACE,SO CLEAR ADDRESS
FF37 7F E4 05                CLR   TRCADR+1
FF3A 3B                      RTI              CONTINUE
                     *
                     * TRACE IS DUE TO N OR T TRACE COMMAND
                     *
FF3B BD FD B7        NBKTRC  JSR   PRINT      PRINT STACK
FF3E FE E4 06                LDX   NTRACE     GET £ OF INSTRUCTIONS TO TRACE
FF41 09                      DEX              DECREMENT COUNT
FF42 FF E4 06                STX   NTRACE     AND RESTORE
FF45 27 DA                   BEQ   CNTRL3     BRANCH IF ALL TRACES DONE
                     *
                     * TRACE NOT DONE - TRACE NEXT INSTRUCTION
                     *
```

```
FF47 B6 E4 0E  CONTRC  LDA A  TRCINS    GET CURRENT INSTRUCTION
FF4A B7 E4 38          STA A  BRINS     SAVE IN CASE IT'S A BRANCH
FF4D 8D 6F             BSR    OPCBYT    GO GET £ OF BYTES/TYPE
FF4F 4D                TST A            CHECK FOR BRANCH
FF50 2A 0A             BPL    CKOBRA    CHECK FOR OTHER THAN BRANCH
               *
               * RELATIVE BRANCH TYPE INSTRUCTION
               * DETERMINE WHERE TO PUT SWI
               * S HOLDS POINTER TO USER STACK AFTER SWI
               *
FF52 32                PUL A            GET CONDITION CODE
FF53 34                DES              UPDATE STACK PTR AFTER PULL
FF54 8A 10             ORA A  %00010000 MAKE INT'S INHIBITED
FF56 06                TAP              RESTORE USER'S C CODE REG
FF57 7E E4 38          JMP    BRINS     GO SEE HOW RELATIVE BRANCH FARES
               *
               * BRANCH WAS NOGO - PUT SWI AT NEXT INSTRUCTION
               *
FF5A 86 02     BRNOGO  LDA A  £2        A:=£ OF BYTES AFTER CURRENT INSTR.
               *
               * INSTRUCTION TO BE TRACED IS NOT A BRANCH
               *
FF5C FE E4 04  CKOBRA  LDX    TRCADR    X:=TRACE ADDRESS
FF5F E6 00             LDA B  0,X       GET INSTR. TO BE TRACED
FF61 C1 6E             CMP B  £$6E      IS IT A JUMP,INDEXED
FF63 27 44             BEQ    JMPIDX    YES GO SIMULATE JUMP IDXED
FF65 C1 7E             CMP B  £$7E      JUMP EXTENDED ?
FF67 27 51             BEQ    JMPEXT
FF69 C1 AD             CMP B  £$AD      JSR,INDEXED ?
FF6B 27 3C             BEQ    JMPIDX    JUMP IDXED IS SAME AS TRANSFERT OF CONTROL
FF6D C1 BD             CMP B  £$BD      JSR,EXTENDED ?
FF6F 27 49             BEQ    JMPEXT
FF71 C1 3B             CMP B  £$3B      RTI ?
FF73 27 3B             BEQ    RTISIM
FF75 C1 39             CMP B  £$39      RTS ?
FF77 27 3C             BEQ    RTSSIM
FF79 C1 3F             CMP B  £$3F      SWI ? IF SWI USER DON'T INCREMENT ADDR.
FF7B 27 13             BEQ    BRG2
FF7D C1 8D             CMP B  £$8D      BSR ? IF YES BRANCH PROCESSING
               *
               * NOT A BRANCH,JUMP,RTI,RTS
               * A REGISTER HOLDS £ OF BYTES IN INSTRUCTION
               *
FF7F 26 0D             BNE    BRG1      PUT IN NEW SWI AND TRACE NEXT INSTR.
               *
               * BRANCH WAS GO,PUT SWI AT ADDRESS BEING JUMP TO
               *
FF81 FE E4 04  BRGO    LDX    TRCADR    X:=TRACE ADDRESS
FF84 A6 01             LDA A  1,X       GET BRANCH OFFSET
FF86 08                INX              OFFSET IS RELATIVE TO
FF87 08                INX              INSTR FOLLOWING BRANCH
FF88 2A 04             BPL    BRG1      BRANCH IS OFFSET POSITIVE
               * X NEEDS TO BE DECREMENTED (OFFSET NEGATIVE)
FF8A 09        BRGODC  DEX              DECREMENT ADDRESS
FF8B 4C                INC A            INCREMENT COUNTER
FF8C 26 FC             BNE    BRGODC
```

```
                 * CONTENT OF A IS 0
FF8E 8D 15       BRG1   BSR   INCX      INCREMENT X BY AMOUNT IN A-REG
FF90 8D 08       BRG2   BSR   SAVTRC    SAVE ADDR/OPC OF NEXT INSTR TO STOP ON
FF92 86 3F              LDA A £SWI      REPLACE INSTR. WITH SWI
FF94 A7 00              STA A X
FF96 BE E4 34    RTIPC  LDS   SP        GET ORIGINAL STACK POINTER
FF99 3B                 RTI             TRACE ANOTHER INSTR.
                 *
                 * SUBROUTINE TO SAVE ADDR IN X INTO TRCADR AND OPC INTO TRCINS
                 *
FF9A             SAVTRC EQU   *
FF9A FF E4 04           STX   TRCADR
FF9D A6 00              LDA A X
FF9F B7 E4 0E           STA A TRCINS
FFA2 39                 RTS
                 *
                 * SUBROUTINE TO INCREMENT X BY CONTENT OF A
                 *
FFA3 08          INXLP  INX             INCREMENT X
FFA4 4A                 DEC A           DECREMENT COUNT
FFA5 4D          INCX   TST A
FFA6 26 FB              BNE   INXLP     IF COUNT NOT YET 0 LOOP
FFA8 39                 RTS             RETURN
                 *
                 * JUMP,JSR INDEXED SIMULATION
                 *
FFA9 A6 01       JMPIDX LDA A 1,X       A:=ADDR OFFSET
FFAB 30                 TSX
FFAC EE 03              LDX   3,X       GET TARGET'S X REG
FFAE 20 DE              BRA   BRG1      UPDATEX TRACE NEXT INSTR
                 *
                 * RTI ENCOUNTERED
                 *
FFB0 30          RTISIM TSX
FFB1 EE 0C              LDX   12,X      GET P-COUNTER FROM STACK
FFB3 20 DB              BRA   BRG2      GO TRACE NEXT INSTR.
                 *
                 * RTS ENCOUNTERED
                 *
FFB5 30          RTSSIM TSX
FFB6 EE 07              LDX   7,X       GET RETURN P-REG FROM STACK
FFB8 20 D6              BRA   BRG2
                 *
                 * JUMP,JSR EXTENDED
                 *
FFBA EE 01       JMPEXT LDX   1,X
FFBC 20 D2              BRA   BRG2
```

```
                       *
                       * OPBCYT
                       *
                       * THIS ROUTINE DETERMINES THE £ OF BYTES IN AN INSTRUCTION
                       * GIVEN ITS OP CODE
                       *
                       * INPUT : A HOLDS THE OP CODE
                       *
                       * OUTPUT : X HOLDS INDEX OF TABLE ELEMENT
                       * B NOT RESTORED
                       * A HOLDS £ OF BYTES IN INSTRUCTION
                       * EXCEPT FOR BRANCHES IN WHICH CASE A IS NEGATIVE
                       *
FFBE                   OPCBYT  EQU   *
FFBE 16                        TAB           B:=OP CODE
FFBF 44                        LSR A
FFC0 44                        LSR A
FFC1 44                        LSR A         PUT 4 UPPER BITS OF OP CODE INTO
FFC2 44                        LSR A         LOWER 4 BITS OF A
                       *
FFC3 CE FF D6                  LDX   £OPBTTB X:=ADDR OF TABLE
FFC6 8D DD                     BSR   INCX    INC X TO POINT TO CORRECT ENTRY
                       *
FFC8 A6 00                     LDA A 0,X     GET TABLE ENTRY
FFCA 26 09                     BNE   OPBTRT  IF NOT 0 THEN NO FURTHER PROCESSING NEEDED
                       *
                       * IF TOP 4 BITS=8 OR C ,THEN THERE ARE TWO CLASSE
                       * OF INSTRUCTIONS : 2 BYTE INSTRUCTIONS AND
                       * CE,8C AND 8E WHICH ARE 3 BYTE INSTRUCTIONS
                       *
FFCC 86 02                     LDA A £2      £ OF BYTES IN MOST OF 8£ INSTRUCTIONS
FFCE C4 0D                     AND B £%00001101
FFD0 C1 0C                     CMP B £%00001100
FFD2 26 01                     BNE   OPBTRT  NO ,RETURN
FFD4 4C                        INC A         £ OF BYTES IN INSTRUCTION :=3
FFD5 39                OPBTRT  RTS           RETURN TO CALLER
                       *
                       * OP CODE TO NUMBER OF BYTES CONVERSION TABLE
                       *
                       *     £ BYTES TOP 4 BITS OF OPCODE
                       *
FFD6                   OPBTTB  EQU   *
FFD6 01                        FCB   1          0
FFD7 01                        FCB   1          1
FFD8 82                        FCB   2+%10000000 2  MINUS = BRANCHES
FFD9 01                        FCB   1          3
FFDA 01                        FCB   1          4
FFDB 01                        FCB   1          5
FFDC 02                        FCB   2          6
FFDD 03                        FCB   3          7
FFDE 00                        FCB   0          8 £ BYTES=2 EXCEPT 8C,8E
FFDF 02                        FCB   2          9
FFE0 02                        FCB   2          A
FFE1 03                        FCB   3          B
FFE2 00                        FCB   0          C £ BYTES=2 EXCEPT CE
```

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

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

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

                     *
                     * JUMP TABLE TO BASIC ROUTINES OF GPMON
                     *
FFE6                       ORG    $FFE6
FFE6 7E FD 12              JMP    CONTRL     WARM START ENTRY POINT
FFE9 7E FB A1              JMP    CHKCHR     CHECK FOR TYPED CHARACTER
FFEC 7E FD 82              JMP    OUTCH      OUTPUT ONE CHARACTER
FFEF 7E FD 98              JMP    INCH       INPUT ONE CHARACTER
FFF2 7E FD 91              JMP    PDATA1     PRINT STRING POINTED BY X-REG
FFF5 7E FD CC              JMP    PCRLF      PRINT CR/LF
                     *
                     * INTERRUPTS VECTORS
                     *
FFF8                       ORG    $FFF8
FFF8 FC D9                 FDB    IQ         REGULAR INTERRUPT
FFFA FC E3                 FDB    SFEI       SOFTWARE INTERRUPT
FFFC FC DE                 FDB    POWDWN     NON MASKABLE INTERRUPT
FFFE FC F4                 FDB    START      RESET INTERRUPT
```

```
                *
                * RAM - LOCATIONS DEVOTED TO VARIABLE INFORMATION
                *
      E400              ORG     RAM         START OF RAM
      0008      NBRBPT  EQU     8           £ OF BREAKPOINTS SUPPORTED
                *
      E400      IOV     RMB     2           I/O INTERRUPT POINTER
      E402      NIO     RMB     2           NMI INTERRUPT POINTER
                *
                * THE FOLLOWING ARE INITIALIZED TO ZERO AT START
                *
      E404      TRCADR  RMB     2           TRACE ADDRESS
      E406      NTRACE  RMB     2           NO. OF INSTRUCTIONS TO TRACE
      E408      OUTSW   RMB     1           OUTPUT SWITCH (0=ECHO INPUT)
      E409      PRTFLG  RMB     1           PRINT FLAG
      E40A      XHI     RMB     1           X REG HIGH (TEMP)
      E40B      XLOW    RMB     1           X REG LOW
                *
      E40C      XSAVE   EQU     *
      E40C              RMB     2
                *
      E40E      TRCINS  RMB     1           OPCODE REPLACED BY SWI IN TRACE MODE
      E40F      BRKSIN  RMB     1           1=BREAKS ARE IN USER PGM
      E410      BRKTRC  RMB     1           1=P-COUNTER IS AT BREAKPOINT AND
                *               USER WANTS TO CONTINUE - ONE TRACE WILL
                *               BE DONE AND BREAKPOINTS RESTORED
      E411              RMB     3           RESERVED FOR IO HANDLER
      E414              RMB     8           FREE SPACE
                *
      E41C      BRKADR  RMB     NBRBPT*3    BREAKPOINT ADDRESS TABLE
      E434      BRKINS  EQU     *
                *
                * THE FOLLOWING ARE INITIALIZED AT START
                *
      E434      SP      RMB     2           USER STACK POINTER
      E436      SWI1    RMB     2           SWI VECTOR
      E438      BRINS   RMB     8           STORAGE FOR CONDITIONNAL BRANCH ROUTINE
      E440      BRANEN  EQU     *           END OF BRANCH ROUTINE+1
                *
                * STACK
                *
      E440      STKEND  RMB     128         RAM USED FOR STACK
      E4BF      STACK   EQU     *-1         START OF STACK AREA
                *
                *
                *
                        END     START
                        END
```

AUCUNE ERREUR DETECTEE

TABLE SYMBOLES

ACIAC	EB0C	ACIAD	EB0D	ACR	000B	ADREND	FCF3	ADRSTR	FCE8
ADTAB1	F9D6	ADTAB2	FAD6	ADVISU	E7F8	BADDR	FD57	BADDRJ	FE52
BADDRS	FD65	BASIC	8000	BELDE1	F5D1	BELDEL	F6CF	BELL	F6BE
BELL1	F6C0	BKFINS	0082	BKFPG1	0009	BKFPGR	0011	BKFPRT	0021
BKFRCD	0005	BKFRMV	0001	BLF	FE1E	BRANEX	E440	BRG	FCF1
BRG1	FF8E	BRG2	FF90	BRG0	FF81	BRGODC	FF8A	BRINS	E435
BRKADR	E41C	BRKINS	E434	BRKP0	FE97	BRKP01	FEA5	BRKP02	FEAF
BRKP03	FEA7	BRKP04	FEB1	BRKP2	FEB6	BRKP3	FEC5	BRKP4	FEDA
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	EBF0
CONT	FE43	CONTRC	FF47	CONTRL	FD12	CPOS	F883	CPOS1	F89F
CPOS3	F897	CPOS4	F8A5	CPOS5	F848	CPOS6	F889	CPOSFG	E417
CPOUT	FC50	CR	000D	CRTC	E870	CRTI1	F680	CRTINT	F671
CRTRET	F82B	CRTRT1	F821	CRTTAB	F8AE	CTRLJ	FE34	CURAD	E573
CUROFF	F8EA	CURON	F875	D15	F634	DATA1	E820	DATA2	E830
DATR91	EBF3	DATRED	EB23	DDRA	0003	DDRB	0002	DECCL	FEF2
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	DKACON	EB02	DMAPRI	EE14
DOWN	002E	DPAGE	E575	DRVRB1	E8E0	DRVREG	E824	DUMP	FE55
DUMP1	FE6D	DUMP2	FE78	ECHOFF	F86E	ECHON	F8C4	ENFCT	F9AC
ENTBL	F9DE	EPAGE	E577	EREOL	F8E0	EREOS	F8EC	EREOS1	F8F5
EREOS2	F900	ERLIN	F8C8	ERROR	FDDA	ESCAPE	F849	ESCFG	E416
ESCS	F851	ESCSQ	F85C	ESCTBL	F9AC	FCTABL	FCAE	FCTBEN	FCD9
FCTCRT	F991	FLBIN	E413	FLGRPT	E412	FLGSHF	E411	FUNCT	F831
FUNCT1	F842	FUNCT2	F834	GAINC	F5BC	GBCRT	F421	GBDIV	F57F
GBPLT1	F4E8	GBPLT2	F51D	GBPLT3	F552	GEWAIT	F43A	GCINC	F56D
GCMD0	F475	GCMD1	F478	GCMD2	F485	GCMDE	F495	GCMPC	F4D6
GCMPT	E402	GCOLOR	F434	GCONTR	E7FC	GCRTC	E7FE	GCRTTB	F454
GCUR	E4C7	GCURX	E7FE	GCURY	E7F9	GDELTA	E4CA	GDSUB	F58D
GDX	E4CE	GDY	E4CF	GFEDIV	F58F	GINFX	F486	GINFY	F4D2
GINIT	F41A	GMOVE	F496	GMVY	F48C	GOLOAD	FE22	GOODCH	FD3E
GOTO	FE37	GPABX	F504	GFASY	F539	GPLOT	F46F	GPRET	F503
GRAPH	F449	GRCINC	F5AE	GREPT	FC13	GSD	E4CD	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	F920	HOME	F6D7	IER	000E	IFR	000D
IN1HG	F956	INACIA	FC43	INCCL	F6DF	INCH	FD98	INCLN	F700
INCLN1	F716	INCX	FFA5	INHEX	FD42	INHEX2	FD44	INICIA	FC25
INIT	F816	INITB2	FB2B	INITCV	FB1E	INITVC	FB48	INITVS	F671
INPUT	F86E	INSBRK	FE91	INVERS	F834	INXLP	FFA3	IO	FCD9
IOV	E400	JMPEXT	FFBA	JMPIDX	FFA9	K7MOD	EC00	KEYB	F8BE
KEYB1	F8D3	KEYB2	F8E0	KEYB3	F8EE	KEYB4	FC01	LF	000A
LF1	FE17	LINCNT	E57D	LINE	E571	LINRET	F6DA	LOADER	A102
LOOP1	F64F	LOOP2	FE5F	MFBOOT	F63F	MOVE1	F743	MOVE2	F740
MOVEA	F737	NBKTRC	FF3B	NBRBPT	000E	NEWLIN	F6FC	NEXT	FE43
NIO	E402	NTRACE	E40E	NXTCHR	F029	ONESEC	F631	OPBTRT	FFDE
OPBTTB	FFD6	OPCBYT	FF8E	ORA	0001	ORB	0000	OUT2H	FDA5
OUT2HA	FDA9	OUT2HS	FDB1	OUT4HS	FDAF	OUTC1	FC37	OUTCH	FC5C

```
SPMON V1.4                              5-12-81  ASSEMBLEUR TSC 6800   PAGE   53

OUTCUR F75B     OUTHL  FD74     OUTHR  FD78     OUTPT1 F801     OUTPUT F7F7
OUTS   FD83     OUTSW  E40B     PADFG  E41B     PAGMOD F873     PCR    000C
PCRLF  FDCC     PDATA1 FD31     PDATA2 FD9E     PDATA3 FD90     PNTBRK FE23
POUT   FC5F     POUTS  FD2B     POWDWN FC0E     PRINT  FDB7     PROMPT FD84
PRPT   002B     PRTFLG E409     PSTAK  FF1E     PTTAB  E4C0     QUEST  FC8C
RAM    E400     READY  F628     REPT   00A0     REPTA  F875     REPTA1 FBBA
REPTA2 FB92     REPTA3 FB9A     REPTA4 F87D     RESERV F951     RESV1  F954
RESV2  F980     ROLMOD F87F     ROM    FCAE     RSTBRK FE1E     RTIPC  FF96
RTISIM FF20     RTN    FE3E     RTSSIM FF65     SAVEA  E4C2     SAVEX  E4C3
SAVTRC FF9A     SBOOT  F5EF     SCROLD F7C8     SCROLL F7EE     SECR91 E8F2
SECREG E822     SETB2  FE32     SETBRK FE2A     SETMEM E8F3     SFE1   FCE3
SHIFT  00A1     SINVER F8B8     SKIP1  0085     SKIP2  008C     SP     E434
STAB1  FB67     STABI  FB5C     STACK  E43F     STAREG E820     START  FCF4
START1 FDFA     START2 FD08     STKEND E440     SWI    003F     SWI1   E436
SWI1S  FF05     SWI1S1 FF15     SWITCH FE8C     SZRCL  0030     TILL   0004
TEMP1  E579     TEMP2  E578     TERM   FC77     TERM1  FC81     TEST   F625
TILL   0024     TINPT1 FBB2     TINPT2 FB9A     TINPUT FBA1     TOACIA FC36
TRACE  FE4F     TRACE1 FE51     TRACE3 FE54     TRCADR E404     TRCINH FF24
TRCIN3 E40E     TRKREG E821     UA     FE14     UCRTC  F73F     UP     002D
UPLINE F71B     VCR    000D     VERSIO F921     VIAC   E810     VIACL  E840
VIAMUS E850     VIDEO  E570     VLF    000A     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           SQ1     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

E50C            SQ2    EQU    B2+2
E50D            STX    EQU    B2+3
                *
E50E            B3     RMB    1
E50E            EOT    EQU    B3
                *
E50F            BAK    RMB    2
E50F            ACKNAK EQU    BAK
E510            SQ3    EQU    BAK+1
                *
                * RAM pour modem BASIC
                *
E511            ERRB   RMB    1         code d erreur
E512            PTBUF  RMB    2         pointeur tampon
E514            BUF    RMB    80        tampon 80 octets
E564            ENDBF  EQU    *         fin tampon
                *
                * RAM mamoeuvre pour cassette
                *
E564            ETIQ   RMB    2         * etiquette demandee
E566            ADRESD RMB    2
E568            ADRESF RMB    2
E56A            CSUM   RMB    1
E56B            COMPTA RMB    2
E56D            ETIQLU RMB    2
E566            AD     EQU    ADRESD    ADRESSE DEBUT
E568            AF     EQU    ADRESF    ADRESSE FIN
```

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

EC00                   ORG    ORGROM
                *---------------------------------------------------- VECTS
                *
                *  Vecteurs de branchement pour MON et BASIC
                *
EC00 7E EC 0F          JMP    KMON+D    COMMANDE K MONITEUR
EC03 7E EC 8B          JMP    SPMOD     ENVOI PROGRAMME BASIC
EC06 7E EC AC          JMP    RPMOD     RECEPTION PROGRAMME BASIC
EC09 7E EC DC          JMP    SDMOD     ENVOI DONNEES
EC0C 7E ED 18          JMP    RDMOD     RECEPTION DONNEES
                *
```

```
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 36  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

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

```
                       *------------------------------------------------------ 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 4D          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 29             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

ECF2 FE E5 12  SDM1    LDX    PTBUF
ECF5 A7 00             STA A  0,X
ECF7 08                INX
ECF8 8C E5 64          CPX    £ENDBF
ECFB 27 09             BEQ    SDM2
ECFD 81 0D             CMP A  £$D
ECFF 27 05             BEQ    SDM2
ED01 FF E5 12          STX    PTBUF
ED04 20 C8     RETB1   BRA    RETB
ED06 09        SDM2    DEX
ED07 FF E5 68          STX    AF
ED0A CE E5 14          LDX    £BUF
ED0D FF E5 66          STX    AD
ED10 BD ED 4D          JSR    EZ
ED13 7F E4 14          CLR    BFLAG
ED16 20 B6             BRA    RETB
               *
               * RECEPTION DONNEES BASIC
               *
ED18 BF E4 FF  RDMOD   STS    STACK
ED1B FF E4 FD          STX    SVXS
ED1E 8D B5             BSR    INBAS
ED20 7D E4 14          TST    BFLAG
ED23 26 0F             BNE    RDM1
ED25 CE E5 14          LDX    £BUF
ED28 FF E5 66          STX    AD
ED2B FF E5 12          STX    PTBUF
ED2E BD EE F6          JSR    RZ
ED31 73 E4 14          COM    BFLAG
ED34 FE E5 12  RDM1    LDX    PTBUF
ED37 A6 00             LDA A  0,X
ED39 08                INX
ED3A 8C E5 64          CPX    £ENDBF
ED3D 27 09             BEQ    RDM2
ED3F 81 0D             CMP A  £$D
ED41 27 05             BEQ    RDM2
ED43 FF E5 12          STX    PTBUF
ED46 20 BC             BRA    RETB1
ED48 7F E4 14  RDM2    CLR    BFLAG
ED4B 20 B7             BRA    RETB1
```

```
                        *-------------------------------------------------------- EZB
                        *
                        *  Emission Zone  de (AD) a (AF)
                        *
ED4D B6 E5 69   EZ      LDA A  AF+1      * calcul L := AF - AD + 1
ED50 B0 E5 67           SUB A  AD+1
ED53 16                 TAB
ED54 B6 E5 68           LDA A  AF
ED57 B2 E5 66           SBC A  AD
ED5A 24 05              BCC    AOK
ED5C 86 01              LDA A  £1
ED5E 7E F2 7B           JMP    ERREUR+D
ED61 CB 01      AOK     ADD B  £1
ED63 F7 E5 07           STA B  L+1
ED66 89 00              ADC A  £0
ED68 B7 E5 06           STA A  L
ED6B 86 16              LDA A  £$16      * preparation B1
ED6D B7 E5 05           STA A  SYN
ED70 4F                 CLR A
ED71 B7 E5 01           STA A  NESSAIS
ED74 B7 E5 08           STA A  SQ1
ED77 86 05              LDA A  £5
ED79 B7 E5 09           STA A  ENQ
ED7C CE E5 05   EB1     LDX    £B1
ED7F C6 05              LDA B  £5
ED81 BD EE 16           JSR    EB+D      * emission B1
ED84 CE ED 7C           LDX    £EB1+D
ED87 BD ED E5           JSR    ATTACR+D
ED8A CE ED D1   EBSUIV  LDX    £EB3+D
ED8D BD EE B2           JSR    CALCLP+D
ED90 B7 E5 0B           STA A  LPB2
ED93 86 01              LDA A  £1        * preparation B2
ED95 B7 E5 0A           STA A  SOH
ED98 B6 E5 08           LDA A  SQ1
ED9B 4C                 INC A            *      +++   +1 sequence emission  +++
ED9C B7 E5 08           STA A  SQ1
ED9F B7 E5 0C           STA A  SQ2
EDA2 86 02              LDA A  £2
EDA4 B7 E5 0D           STA A  STX
                        *
EDA7 CE E5 0A   EB2     LDX    £B2
EDAA C6 04              LDA B  £4
EDAC BD EE 16           JSR    EB+D      * emission B2 *
EDAF 7F E5 02           CLR    LRCC
EDB2 FE E5 66           LDX    AD
EDB5 F6 E5 03           LDA B  LPC
EDB8 BD EE 35           JSR    EMOTS+D   *  emission DONNEES
EDBB B6 E5 02           LDA A  LRCC
EDBE BD EE 47           JSR    E1MOT+D   * emission LRC
EDC1 86 03              LDA A  £3
EDC3 BD EE 47           JSR    E1MOT+D   * emission ETX
EDC6 CE ED A7           LDX    £EB2+D
EDC9 BD ED E5           JSR    ATTACR+D
```

```
EDCC BD EE D3            JSR    ADJADL+D
EDCF 20 B9               BRA    EBSUIV
                  *
EDD1 86 04        EB3    LDA A  £4
EDD3 B7 E5 0E            STA A  EOT
EDD6 CE E5 0E            LDX    £B3
EDD9 C6 01               LDA B  £1
EDDB BD EE 16            JSR    EB+D        * emission B3 ( EOT )
EDDE CE ED D1            LDX    £EB3+D
EDE1 BD ED E5            JSR    ATTAQR+D
EDE4 39                  RTS                *       // fin normale EZ //
                  **********************
                  *
                  *   ATTente Accuse de Reception
                  *

EDE5 FF E4 F5     ATTAQR STX    SVX1
EDE8 CE E5 0F            LDX    £BAK
EDEB C6 02               LDA B  £2
EDED BD EF DA            JSR    RB+D
EDF0 B6 E5 0F            LDA A  ACKNAK
EDF3 81 06               CMP A  £6
EDF5 26 0C               BNE    RNAK
EDF7 B6 E5 10            LDA A  S03
EDFA B1 E5 0E            CMP A  S01
EDFD 26 04               BNE    RNAK
EDFF 7F E5 01            CLR    NESSAIS     * on a recu ACK et un no seq correct
EE02 39                  RTS
EE03 F6 E5 01     RNAK   LDA B  NESSAIS
EE06 5C                  INC B
EE07 C1 04               CMP B  £4
EE09 26 05               BNE    INSAKR
EE0B 86 02               LDA A  £2
EE0D 7E F2 78            JMP    ERREUR+D
EE10 F7 E5 01     INSAKR STA B  NESSAIS
EE13 7E EF DE            JMP    INSAK+D     ** retour a l adresse de reprise
                  **********************
                  *
                  *  Emission Bloc d adresse ( X ) long. ( B )
                  *

EE16 FF E4 F8     EB     STX    SVX4
EE19 86 7E               LDA A  £X01111110
EE1B B7 E6 52            STA A  VIPT+DORB
EE1E CE 00 64            LDX    £100        *  zero pendant 300 ms.
EE21 FF E4 F7     BAL    STX    SVX2
EE24 8D 5C               BSR    EMEZRO
EE26 FE E4 F7            LDX    SVX2
EE29 09                  DEX
EE2A 26 F5               BNE    BAL
EE2C CE 04 E2            LDX    £1250       * un pendant 10 ms
EE2F BD EE AE            JSR    DELAI+D
EE32 FE E4 F8            LDX    SVX4
EE35 A6 00        EXCTS  LDA A  0,X
```

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

  382  EE37 36              PSH A
  383  EE38 B8 E5 02        ADD A  LACC
  384  EE3B B7 E5 02        STA A  LACC
  385  EE3E 32              PUL A
  386  EE3F 8D 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        E1MOT   PSH B
     EE48 36                PSH A
     EE49 FF E4 F9          STX   SVX3
     EE4C 86 60             LDA A £X01100000 * 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 £X11110001
     EE56 8A 0C             ORA A £X00001100 *  CA2 = 0  sync osc.
     EE58 B7 E8 6C          STA A VIAT+PCR
                    *
     EE5B 86 20             LDA A £X00100000 * 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 £X00001110 * 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 2D             BRA   SHIFTC
                    ***
     EE75 86 20     EMIS1   LDA A £X00100000 *  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     EM1P    LDA A £X01100000 * 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 £X00000000 * PB6-5 = 0
     EE91 B7 E8 60          STA A VIAT+ORB
     EE94 CE 00 1B          LDX   £X0B
```

```
EE97 8D 15               BSR    DELAI
EE99 5A                  DEC B
EE9A 26 E9               BNE    EM1P
EE9C 86 20               LDA A  £%00100000 * PB6-5 = 1/2
EE9E B7 E8 60            STA A  VIAT+ORB
EEA1 33                  PUL B
EEA2 39                  RTS
                 ***
EEA3 32          SHIFTC  PUL A
EEA4 0D                  SEC
EEA5 46                  ROR A
EEA6 5A                  DEC B
EEA7 26 C6               BNE    EMBIT
                 ***
EEA9 FE E4 F9            LDX    SVX3
EEAC 33                  PUL B
EEAD 39                  RTS
                 **
EEAE 09          DELAI   DEX
EEAF 26 FD               BNE    DELAI
EEB1 39                  RTS
                 ***
```

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

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

EEB2 FF E4 F5   CALCLP  STX     SVX1
EEB5 FE E5 06           LDX     L
EEB8 26 07              BNE     COX
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   COX     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
```

```
                        *----------------------------------------------------- RZB
                        *
                        *  Reception Zone de ( AD) a ( AD ) + ( L )
                        *
                        *

EEF6 4F         RZ      CLR A
EEF7 B7 E5 01           STA A  NESSAIS
EEFA CE E5 05   RB1     LDX    £B1
EEFD C6 05              LDA B  £5
EEFF BD EF DA           JSR    RB+D        * reception B1
                        *
EF02 CE EE FA           LDX    £RB1+D
EF05 B6 E5 05           LDA A  SYN
EF08 81 16              CMP A  £$16
EF0A 26 0A              BNE    RI1
EF0C B6 E5 08           LDA A  SQ1
EF0F 26 05              BNE    RI1
EF11 B6 E5 09           LDA A  ENQ
EF14 81 05              CMP A  £5
EF16 BD EF 9C   RI1     JSR    ENVAQR+D
EF19 CE EF 89   RBSUIV  LDX    £RB3+D
EF1C BD EE B2           JSR    CALCLP+D
EF1F 7C E5 08           INC    SQ1         *      ///  + 1  sequence  ///
EF22 CE E5 0A   RB2     LDX    £B2
EF25 C6 04              LDA B  £4
EF27 BD EF DA           JSR    RB+D        * reception debut B2
EF2A 7F E5 02           CLR    LRCC
EF2D B6 E5 0A           LDA A  SOH
EF30 81 01              CMP A  £1
EF32 26 17              BNE    RI2
EF34 B6 E5 0B           LDA A  LPB2
EF37 B1 E5 03           CMP A  LPC
EF3A 26 0F              BNE    RI2
EF3C B6 E5 0C           LDA A  SQ2
EF3F B1 E5 08           CMP A  SQ1
EF42 26 07              BNE    RI2
EF44 B6 E5 0D           LDA A  STX
EF47 81 02              CMP A  £2
EF49 27 08              BEQ    RSUIB2
EF4B CE FF FF   RI2     LDX    £$FFFF      *  VERIFIER SI SUFFISANT ////
EF4E BD EE AE   DELSAUT JSR    DELAI+D     *  laisser passer les donnees ....
EF51 20 26              BRA    ENVNAK      *   avant d envoyer NAK
EF53 FE E5 66   RSUIB2  LDX    AD
EF56 F6 E5 03           LDA B  LPC
EF59 BD F0 01           JSR    RMOTS+D     * reception DONNEES
EF5C 09                 DEX
EF5D FF E5 68           STX    AF
EF60 08                 INX
                        *
EF61 BD F0 10           JSR    R1MOT+D     * reception LRC
EF64 B1 E5 02           CMP A  LRCC
EF67 27 05              BEQ    RETX
```

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

EF69 CE 11 11           LDX    £$1111
EF6C 20 E0              BRA    DELSAUT    *  laisser passer ETX
EF6E BD F0 10  RETX     JSR    R1MOT+D    * reception ETX
EF71 81 03              CMP A  £3
EF73 20 06              BRA    ENVV2
EF75 86 00     ENVACK   LDA A  £0
EF77 20 02              BRA    ENVV2
EF79 86 FF     ENVNAK   LDA A  £$FF
EF7B 07        ENVV2    TPA
EF7C 36                 PSH A
EF7D CE EF 22           LDX    £RB2+D
EF80 32                 PUL A
EF81 06                 TAP
EF82 8D 18              BSR    ENVAQR
EF84 BD EE D3           JSR    ADJADL+D
EF87 20 90              BRA    RBSUIV
               *
EF89 CE E5 0E  RB3      LDX    £B3
EF8C C6 01              LDA B  £1
EF8E BD EF DA           JSR    RB+D       *  reception B3 ( EOT )
EF91 CE EF 89           LDX    £RB3+D
EF94 B6 E5 0E           LDA A  EOT
EF97 81 04              CMP A  £4
EF99 8D 01              BSR    ENVAQR
EF9B 39                 RTS

               ****************************
               *
               *   ENVoi Accuse de Reception
               *
               *     Z=1  =) ACK et   0-)NESSAIS
               *
               *     Z=0  =) NAK et controle NESSAIS :: 4

EF9C 07        ENVAQR   TPA
EF9D 36                 PSH A
EF9E FF E4 F5           STX    SVX1
EFA1 06                 TAP
EFA2 27 07              BEQ    EAK
EFA4 86 0F              LDA A  £15        * NAK Ascii
EFA6 7C E5 01           INC    NESSAIS
EFA9 20 03              BRA    STABQ
               *
EFAB 86 06     EAK      LDA A  £06        *  ACK Ascii
EFAD 5F                 CLR B
EFAE B7 E5 0F  STABQ    STA A  ACKNAK
EFB1 F7 E5 01           STA B  NESSAIS
EFB4 B6 E5 08           LDA A  SQ1
EFB7 B7 E5 10           STA A  SQ3
EFBA CE E5 0F           LDX    £BAK
EFBD C6 02              LDA B  £2
EFBF BD EE 16           JSR    EB+D       *  emission buffer ACK ou NAK  + seq
EFC2 B6 E5 01           LDA A  NESSAIS    *  controler si moins de 4 essais
EFC5 81 04              CMP A  £4
```

```
EFC7 27 0C              BEQ    ERR3
EFC9 32                 PUL A
EFCA 06                 TAP
EFCB 26 01              BNE    INSAK
EFCD 39                 RTS           *      === retour normal   ===
EFCE 31        INSAK    INS
EFCF 31                 INS           *  ajuster pointeur de pile *
EFD0 FE E4 F5           LDX    SVX1
EFD3 6E 00              JMP    0,X    * retour d exception
EFD5 86 03     ERR3     LDA A  £3     *  erreur ---) retour
EFD7 7E F2 7B           JMP    ERREUR

               ********************************
               *
               *     Reception de Bloc  adresse ( X ) long. ( B )
               *

EFDA FF E4 F7  RB       STX    SVX2
EFDD 37                 PSH B
EFDE 86 7E              LDA A  £%01111110
EFE0 B7 E8 62           STA A  VIAT+DDRB
EFE3 5F        ATTZRD   CLR B
EFE4 B6 E8 60  TB00     LDA A  VIAT+IRB
EFE7 84 01              AND A  £1
EFE9 26 F8              BNE    ATTZRD
EFEB CE 01 36           LDX    £310
EFEE BD EE AE           JSR    DELAI+D   *  environ 2.5 ms
EFF1 5C                 INC B
EFF2 C1 14              CMP B  £20
EFF4 26 EE              BNE    TB00
               *  on est sur du zero initial sur 50 ms
EFF6 B6 E8 60  TB01     LDA A  VIAT+IRB
EFF9 84 01              AND A  £1
EFFB 27 F9              BEQ    TB01
               *  on a detecte le pseudo bit STOP
EFFD FE E4 F7           LDX    SVX2
F000 33                 PUL B

F001 8D 0D     RMOTS    BSR    R1MOT
F003 A7 00              STA A  0,X
F005 BB E5 02           ADD A  LRCC
F008 B7 E5 02           STA A  LRCC
F00B 08                 INX
F00C 5A                 DEC B
F00D 26 F2              BNE    RMOTS
F00F 39                 RTS
```

```
                     *------------------------------------------------------------------ RMOT
                     *
                     *
                     *    Reception 300 bauds
                     *
                     
                     **   definition des constantes de delai
                     
      0014           XSTOP   EQU   20
      0032           X01     EQU   53-3
      001A           X02     EQU   26
      008D           X1AR    EQU   141
      00E2           X3AR    EQU   225
      00EA           X3      EQU   240-6
                     
                     
                     ***
                     
      F010 37        R1MOT   PSH B
      F011 FF E4 F9          STX   SVX3
                     
      F014 B6 E8 60  RCC     LDA A VIAT+IRB
      F017 84 01             AND A £1
      F019 27 F9             BEQ   RCC        * recherche du stop
      F01B CE 00 14          LDX   £XSTOP
      F01E 8D 4A             BSR   DELAIR
      F020 B6 E8 60          LDA A VIAT+IRB
      F023 84 01             AND A £1
      F025 27 ED             BEQ   RCC        * PB0=1 instable re-chercher
      F027 86 01             LDA A £1         * PB0=1 STABLE
      F029 B5 E8 60  ATTSTR  BIT A VIAT+IRB   * boucle de
      F02C 26 FB             BNE   ATTSTR     * detection du start
      F02E 01        CLRCA2  NOP              **
      F02F B6 E8 6C          LDA A VIAT+PCR   **
      F032 84 11             AND A £X00010001
      F034 8A CC             ORA A £X11001100 ** CA2 + CB2 niveau bas
      F036 B7 E8 6C          STA A VIAT+PCR   **
      F039 CE 00 32          LDX   £X01
      F03C 8D 2C             BSR   DELAIR     * 01 = 24+8*X01
      F03E B6 E8 6C          LDA A VIAT+PCR
      F041 84 11             AND A £X00010001
      F043 8A EE             ORA A £X11101110 ** CA2 + CB2 niveau haut
      F045 B7 E8 6C          STA A VIAT+PCR
      F048 B6 E8 60          LDA A VIAT+IRB
      F04B 84 01             AND A £1
      F04D 26 C5             BNE   RCC        * START INSTABLE RE_SYNCHRO
      F04F CE 00 1A          LDX   £X02
      F052 8D 16             BSR   DELAIR     * T02 = 36 + 8*X02
      F054 8D 18             BSR   R1BIT      * pour confirmer le start
      F056 25 BC             BCS   RCC        * perte du start RE-SYNCHRO
      F058 C6 08             LDA B £8
      F05A 8D 12     BITSUI  BSR   R1BIT
      F05C 76 E5 04          ROR   CRECU
```

```
F05F 5A                  DEC B
F060 26 F8               BNE    BITSUI

F062 FE E4 F9            LDX    SVX3
F065 B6 E5 04            LDA A  CRECU
F068 33                  PUL B
F069 39                  RTS               * T4 = 20

F06A 09          DELAIR  DEX
F06B 26 FD               BNE    DELAIR
F06D 39                  RTS

                 *
                 **
                 ***  RECEPTION 1 BIT
                 **
                 *
F06E CE 00 80    R1BIT   LDX    £X1AR
F071 8D F7               BSR    DELAIR
F073 B6 E8 6C            LDA A  VIAT+PCR  *    INVERSION
F076 88 20               EOR A  £X00100000 *      DE
F078 B7 E8 6C            STA A  VIAT+PCR  *      C B 2
F07B B6 E8 60            LDA A  VIAT+IRB
F07E 46                  ROR A
F07F CE 00 E2            LDX    £X3AR
F082 8D E6               BSR    DELAIR
F084 39                  RTS
```

```
RBASK7                   12-9-81  TSC ASSEMBLER  PAGE   19

                     *---------------------------------------------------------------- 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

F0C8 B6 E8 60  TIM2    LDA A  VIAT+IRB
F0CB BA E8 60          ORA A  VIAT+IRB
F0CE BA E8 60          ORA A  VIAT+IRB
F0D1 84 80             AND A  £$80
F0D3 26 F0             BNE    OUTTIM
F0D5 C6 50             LDA B  £$50      PARTIE DU DELAI 780 MIC.SEC
F0D7 8D AC             BSR    DELAY
F0D9 C6 10             LDA B  £$10      * FENETRE DE 300 MIC SEC
F0DB B6 E8 60  TEST1   LDA A  VIAT+IRB
F0DE 84 80             AND A  £$80
F0E0 27 0A             BEQ    LULEUN
F0E2 5A                DEC B
F0E3 26 F6             BNE    TEST1
F0E5 C6 20             LDA B  £$20
F0E7 8D 9C             BSR    DELAY
F0E9 0C                CLC              * SORTIE POUR LE ZERO
F0EA 20 05             BRA    LBX
F0EC C6 30     LULEUN  LDA B  £$30      * DELAI APRES LE UN
F0EE 8D 95             BSR    DELAY     * 210 MIC SEC
F0F0 0D                SEC              * SORTIE POUR LE UN
F0F1 33        LBX     PUL B
F0F2 32                PUL A
F0F3 39                RTS

                       * S/P DE LECTURE 1 MOT =(A) = MOT1
F0F4 C6 08     LIRMOT  LDA B  £8
F0F6 4F                CLR A
F0F7 8D C8     TOUJOU  BSR    LIRBIT    *
F0F9 2A 05             BPL    TIM3
F0FB 31                INS
F0FC 31                INS
F0FD 7E F1 81          JMP    COMLR+D
F100 46        TIM3    ROR A
F101 5A                DEC B
F102 26 F3             BNE    TOUJOU    * COMPTAGE 1 A 8
F104 4D                TST A
F105 39                RTS

                       * S/P DE CHEK-SUM
F106 16        CSOMME  TAB
F107 FB E5 6A          ADD B  CSUM      * COMPLETER
F10A F7 E5 6A          STA B  CSUM      * CSUM PAR (A)
F10D 39                RTS

F10E 86 7F     INIPR   LDA A  £$7F      INIT DES BITS DE GESTION K7
F110 B7 E8 62          STA A  VIAT+DDRB
F113 39                RTS

                       ************************************************
                       *
                       * MODULE D'ECRITURE K7
                       *
                       *******************************************
F114 8D F8     EZK     BSR    INIPR
```

```
F116 CE 13 88  COMETR LDX    £5000
F119 FF E5 6B  BAL3   STX    COMPTA    * 3" DE BALISES (DES ZEROS)
F11C 4F               CLR A
F11D BD F0 8A         JSR    ECRBIT+D
F120 FE E5 6B         LDX    COMPTA
F123 09               DEX
F124 26 F3            BNE    BAL3
F126 86 01            LDA A  £1        * LE "1" TOUT SEUL
F128 BD F0 8A         JSR    ECRBIT+D
F12B 7F E5 6A         CLR    CSUM
F12E CE E5 64         LDX    £ETIQ
F131 FF E4 F5  ECEK7  STX    SVX1
F134 A6 00            LDA A  0,X
F136 BD F1 06         JSR    CSOMME+D
F139 BD F0 B4         JSR    ECRMOT+D
F13C FE E4 F5         LDX    SVX1
F13F 08               INX
F140 8C E5 6A         CPX    £ETIQ+6
F143 26 EC            BNE    ECEK7
F145 FE E5 66         LDX    ADRESD    * EXTRAIRE 1 MOT EN RAM
F148 A6 00     UNMOT  LDA A  0,X
F14A BD F1 06         JSR    CSOMME+D  * COMPLETER "CS"
F14D BD F0 B4         JSR    ECRMOT+D  * ECRIRE LE MOT SUR LA K7
F150 BC E5 68         CPX    ADRESF    * TESTER LA FIN
F153 27 03            BEQ    ECRICS
F155 08               INX              * PROGRESSER L'ADRESSE MOT
F156 20 F0            BRA    UNMOT
F158 B6 E5 6A  ECRICS LDA A  CSUM      * ECRIRE LE CS EN FIN
F15B BD F0 B4         JSR    ECRMOT+D
F15E 7D E4 19         TST    MFLG
F161 26 14            BNE    EZKRT
F163 8D 13            BSR    SIGNAL
F165 CE F2 6F         LDX    £MECR+D
F168 BD FF F2         JSR    CHAIN     * ECRITURE TERMINE
F16B B6 E5 64         LDA A  ETIQ
F16E BD FF EC         JSR    OUTCH     * IMP.L'ETIQUETTE
F171 B6 E5 65         LDA A  ETIQ+1
F174 BD FF EC         JSR    OUTCH
F177 39        EZKRT  RTS

               * envoi du message fin
               *
F178 CE F2 67  SIGNAL LDX    £MFIN+D
F17B 7E FF F2         JMP    CHAIN     * qui fait RTS

               ***************************
               *
               * MODULE DE LECTURE K7
               *
               *
F17E BD F1 0E  LZK    JSR    INIPR+D
F181 86 32     COMLR  LDA A  £50
F183 B7 E5 68         STA A  COMPTA
F185 BD F0 F4  ZERMOT JSR    LIRMOT+D  * ATTENTE DE 50 MOTS
```

```
F189 26 F6              BNE     COMLR      * SUPPOSES DANS LES BALISES
F18B 7A E5 6B           DEC     COMPTA
F18E 26 F6              BNE     ZERMOT
F190 86 07              LDA A   £7
F192 BD FF EC           JSR     OUTCH
F195 86 32     COMLR1   LDA A   £50
F197 B7 E5 6B           STA A   COMPTA
F19A BD F0 F4  ZERMO1   JSR     LIRMOT+D
F19D 26 F6              BNE     COMLR1
F19F 7A E5 6B           DEC     COMPTA
F1A2 26 F6              BNE     ZERMO1
F1A4 BD F0 C1  ATTEN1   JSR     LIRBIT+D   * ATTENTE DU "1" TOUT SEUL
F1A7 2B D8              BMI     COMLR
F1A9 24 F9              BCC     ATTEN1
F1AB BD F0 F4           JSR     LIRMOT+D   * LIRE ETIQUETTE
F1AE 26 05              BNE     ETQL1
F1B0 86 05              LDA A   £5
F1B2 7E F2 7B           JMP     ERREUR+D
F1B5 B7 E5 6D  ETQL1    STA A   ETIQLU
F1B8 BD F0 F4           JSR     LIRMOT+D
F1BB B7 E5 6E           STA A   ETIQLU+1
F1BE BB E5 6D           ADD A   ETIQLU
F1C1 B7 E5 6A           STA A   CSUM
F1C4 FE E5 64           LDX     ETIQ       * TEST DE CONFORMITE
F1C7 8C 30 30           CPX     £$3030
F1CA 27 05              BEQ     INFO
F1CC BC E5 6D           CPX     ETIQLU     * ETIQUETTES
F1CF 26 B0              BNE     COMLR
F1D1 CE E5 66  INFO     LDX     £ADRESD
F1D4 FF E4 F5  INFOSX   STX     SVX1
F1D7 BD F0 F4           JSR     LIRMOT+D
F1DA FE E4 F5           LDX     SVX1
F1DD A7 00              STA A   0,X
F1DF BD F1 06           JSR     CSOMME+D
F1E2 08                 INX
F1E3 8C E5 6A           CPX     £ADRESD+4
F1E6 26 EC              BNE     INFOSX
F1E8 FE E5 66           LDX     ADRESD
F1EB BD F0 F4  LIR1MO   JSR     LIRMOT+D   * LIRE UN MOT
F1EE A7 00              STA A   X          * LE GARER
F1F0 BD F1 06           JSR     CSOMME+D   * COMPLETER LE "CS"
F1F3 BC E5 68           CPX     ADRESF     * TEST TERMINE
F1F6 27 03              BEQ     LIRCS
F1F8 08                 INX
F1F9 20 F0              BRA     LIR1MO
F1FB BD F0 F4  LIRCS    JSR     LIRMOT+D   * LIRE "CS"
F1FE B1 E5 6A           CMP A   CSUM       * comparer au checksum recalcule
F201 27 0E              BEQ     BONNLEC
F203 CE F2 75           LDX     £MLEC
F206 BD FF F2           JSR     CHAIN
F209 CE F2 5B           LDX     £MANDR+D
F20C BD FF F2           JSR     CHAIN
F20F 20 0E              BRA     LZKS
F211 7D E4 19  BONNLEC  TST     MFLG
```

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

```
                    *-------------------------------------------------------- MSGS
                    *
                    *  MESSAGES
                    *
F22C 0A 0D          MLERT   FDB    $0A0D
F22E 4C                     FCC    /LERT ? /
F235 04                     FCB    4
F236 0A 0D          MAD     FDB    $0A0D
F238 44                     FCC    /DEB ?√/
F23E 04                     FCB    4
F23F 0A 0D          MAF     FDB    $0A0D
F241 46                     FCC    /FIN ? /
F247 04                     FCB    4
F248 0A 0D          METIQ   FDB    $0A0D
F24A 45                     FCC    /ETI ? /
F250 04                     FCB    4
F251 0A 0D          MERRU   FDB    $0A0D
F253 45                     FCC    /Erreur /
F25A 04                     FCB    4
F25B 07             MANOR   FCB    7
F25C 41                     FCC    /ANORMALE /
F265 07                     FCB    7
F266 04                     FCB    4
F267 0A 0D          MFIN    FDB    $0A0D
F269 07                     FCB    7
F26A 46                     FCC    /Fin /
F26E 04                     FCB    4
F26F 65             MECR    FCC    /ecr. /
F274 04                     FCB    4
F275 6C             MLEC    FCC    /lec. /
F27A 04                     FCB    4
                    *
                    * envoi du message d erreur
                    *
F27B 36             ERREUR  PSH A
F27C CE F2 51               LDX    £MERRU+D
F27F BD FF F2               JSR    CHAIN
F282 32                     PUL A
F283 B7 E5 11               STA A  ERRB
F286 8B 30                  ADD A  £'0
F288 BD FF EC               JSR    OUTCH
F28B BE E4 FF               LDS    STACK
F28E 7D E4 15               TST    EFLAG
F291 26 03                  BNE    ERRS
F293 7E FF E6               JMP    MONIT
F296 7F E4 15       ERRS    CLR    EFLAG
F299 7F E4 14               CLR    BFLAG
F29C FE E4 FD               LDX    SVXS
F29F 86 0D                  LDA A  £$D
F2A1 39                     RTS              RETOUR BASIC
```

```
                          *------------------------------------------------- KBAS
                          *
                          *  interface BASIC --) logiciel cassette
                          *
F300                             ORG    $F300      POUR BASIC REPROM
F300 7E F3 51                    JMP    PDTAPE
F303 7E F3 A0                    JMP    RDTAPE
F306 7E F3 2E                    JMP    LPTAPE
                          *
                          * SAUVEGARDE PROGRAMME BASIC'
                          *
F309 BF E4 FF    SPTAPE   STS    STACK
F30C BD EC D5                    JSR    INBAS
F30F A6 01                       LDA A  1,X
F311 B7 E5 66                    STA A  AD
F314 A6 02                       LDA A  2,X
F316 B7 E5 67                    STA A  AD+1       ADRESSE DEBUT
F319 A6 03                       LDA A  3,X
F31B B7 E5 68                    STA A  AF
F31E A6 04                       LDA A  4,X
F320 B7 E5 69                    STA A  AF+1       ADRESSE FIN
F323 FF E4 FD                    STX    SVX5
F326 BD EC 68                    JSR    DETIQ      ETIQUETTE
F329 BD F1 14                    JSR    EZK        ECRIRE PROGRAMME
F32C 20 1C                       BRA    RETB2      RETOUR BASIC
                          *
                          * CHARGEMENT PROGRAMME BASIC
                          *
F32E BF E4 FF    LPTAPE   STS    STACK
F331 BD EC D5                    JSR    INBAS
F334 FF E4 FD                    STX    SVX5
F337 BD EC 68                    JSR    DETIQ      ETIQUETTE
F33A BD F1 7E                    JSR    LZK        LIRE PROGRAMME
F33D FE E4 FD                    LDX    SVX5
F340 B6 E5 68                    LDA A  AF
F343 A7 03                       STA A  3,X
F345 B6 E5 69                    LDA A  AF+1
F348 A7 04                       STA A  4,X        ADRESSE FIN
F34A 7F E4 15    RETB2           CLR    EFLAG
F34D FE E4 FD                    LDX    SVX5
F350 39                          RTS               RETOUR BASIC
                          *
                          * ECRITURE DONNEES BASIC
                          *
F351 BF E4 FF    PDTAPE   STS    STACK
F354 FF E4 FD                    STX    SVX5
F357 BD EC D5                    JSR    INBAS
F35A 81 0A                       CMP A  £$A
F35C 27 EC                       BEQ    RETB2
F35E 7D E4 14                    TST    BFLAG
F361 26 09                       BNE    PDT1
F363 CE E5 14                    LDX    £BUF
F366 FF E5 12                    STX    PTBUF
```

```
F369 73 E4 14          COM   BFLAG
F36C FE E5 12  PDT1    LDX   PTBUF
F36F A7 00             STA A 0,X
F371 08                INX
F372 8C E5 64          CPX   £ENDBF
F375 27 09             BEQ   PDT2
F377 81 0D             CMP A £$D
F379 27 05             BEQ   PDT2
F37B FF E5 12          STX   PTBUF
F37E 20 CA     RETB3   BRA   RETB2
F380 09        PDT2    DEX
F381 FF E5 68          STX   AF
F384 CE 30 30          LDX   £$3030
F387 FF E5 64          STX   ETIQ
F38A CE E5 14          LDX   £BUF
F38D FF E5 66          STX   AD
F390 73 E4 19          COM   MFLG
F393 36                PSH A
F394 BD F1 14          JSR   EZK
F397 32                PUL A
F398 7F E4 19          CLR   MFLG
F39B 7F E4 14          CLR   BFLAG
F39E 20 DE             BRA   RETB3
                       *
                       * LECTURE DONNEES BASIC
                       *
F3A0 BF E4 FF  RDTAPE  STS   STACK
F3A3 FF E4 FD          STX   SVX5
F3A6 BD EC D5          JSR   INBAS
F3A9 7D E4 14          TST   BFLAG
F3AC 26 18             BNE   RDT1
F3AE CE 30 30          LDX   £$3030
F3B1 FF E5 64          STX   ETIQ
F3B4 CE E5 14          LDX   £BUF
F3B7 FF E5 12          STX   PTBUF
F3BA 73 E4 19          COM   MFLG
F3BD BD F1 7E          JSR   LZK
F3C0 7F E4 19          CLR   MFLG
F3C3 73 E4 14          COM   BFLAG
F3C6 FE E5 12  RDT1    LDX   PTBUF
F3C9 A6 00             LDA A 0,X
F3CB 08                INX
F3CC 8C E5 64          CPX   £ENDBF
F3CF 27 09             BEQ   RDT2
F3D1 81 0D             CMP A £$D
F3D3 27 05             BEQ   RDT2
F3D5 FF E5 12          STX   PTBUF
F3D8 20 A4             BRA   RETB3
F3DA 7F E4 14  RDT2    CLR   BFLAG
F3DD 20 9F             BRA   RETB3
                       END
```

NO ERROR(S) DETECTED

```
 SYMBOL TABLE:

ACKNAK E50F   AD     E566   ADJADL EED3   ADRESD E566   ADRESF E568
AF     E568   ADK    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 F1E8   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    EF89   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    EF4B   RMOTS  F001   RNAK   EE03
RPMOD  ECAC   RSUIB2 EF53   RZ     EEF6   SDM1   ECF2   SDM2   ED06
SDMOD  ECDC   SHIFTC EEA3   SIGNAL F178   SOH    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  F0D8   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
```

- VI -

- 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 $\begin{cases} X = T \text{ pour transmettre} \\ X = R \text{ pour recevoir} \end{cases}$

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

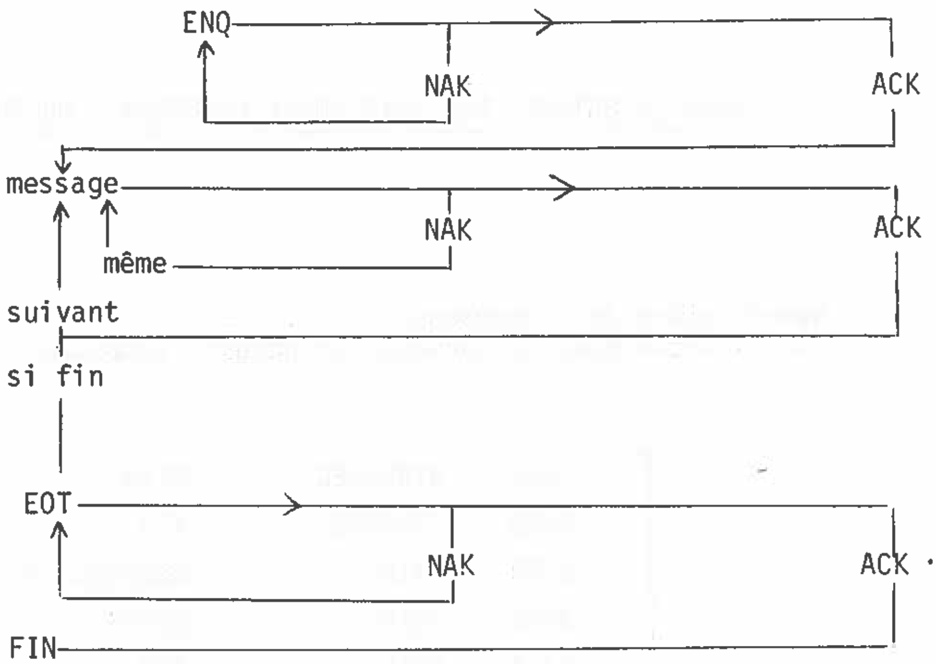
Si la transmission totale est incorrecte le message suivant apparait sur les deux écrans :

TRANSMISSIØN INCØRRECTE
(Il faut reprendre la procédure)

Annexe 1 : Diagramme d'une liaison

Emetteur Récepteur

Initialisation



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.

1/ NOMENCLATURE DES COMPOSANTS

NOMENCLATURE GOUPIL : PRESENTATION GENERALE

Carte CPU (1) Carte E/S (1) Carte BUS (1) Carte MODEM-K7 (1) Carte MEMOIRE (1)	Connectique Cordon V24 (1) Cordon clavier + connecteur (2) Cordon liaison // (1) Câble secteur (1) Connecteurs Molex (4)	Haut parleur (1) Serre-Câbles (5) Documentation (2)
Boite (1) Rack (1) Alimentation (1) Moniteur vidéo (1)(op.) Clavier (1)	Prises Entrée Secteur (1) Prise canon v24 (1) Prise DIN (2) Prise vidéo UHF (1)	Pop visserie ventilateur (1)
Oreillettes (option) Grilles (2) Coupelles (2) pastille microphonique (1) Haut parleur (1) Boite-câble connexion	Boutons Interrupteur (1) Bouton poussoir (1) Fils électriques	

NOMENCLATURE DE LA CARTE CRT-CPU

Circuits intégrés	TTL	Quartz	Capacités	Résistances 1/4W	Résistances 1/8W	Supports C.I.
2708 (1)	74LS00(2)	4 Mz (1)	27pF(2)	150Ω(2)	10 KΩ(7)	14 p (5)
2114 (4)	74LS04(1)	1,008Mz (1)	100pF(1)	220Ω(1)		16 p (11)
6522 (1)	74LS132(1)		1µF TG(10)	680Ω(2)		18 p (4)
6808 (1)	74LS138(2)		6,8µF TG(1)	1KΩ(2)	ConnecteurOEC(1)	20 p (3)
6850 (1)	74LS161(1)	Diodes	Chim 6,8µF(1)	2,2KΩ(1)		24 p (2)
96364 (1)	74LS244(1)		Chim 68µF(1)	4,7KΩ(3)	Cossefaston (2)	28 p (1)
	74LS273(1)	1N4148 (1)		5,6K (9)		40 p (2)
	74LS139(1)		10nf(1)	10K Ω (5)	SupUMD Bas profil 16 (1)	
CMOS	74LS166(1)	Transistors	Potentiomètre	100KΩ (1)		
	74LS367(3)			1MΩ (1)	Broche Wrapping UMD	
4049 (1)	74LS368(1)	BC-327-25(1)	Cermet		Vernis épargne circuit	
	74LS245(1)	BC-337-25(3)	220Ω (1)	10MΩ (2)		
	74LS123(1)					

NOMENCLATURE DE LA CARTE MEMOIRE STATIQUE 16 K

Circuits intégrés	TTL	Capacité
2114 (32)	74LS138 (4)	6,8 µf TG (10)
Supports circuits intégrés 16 p (4) 18 p (32)	Broches Wrapping UMD Circuit vernis épargne	

NOMENCLATURE CARTE MODEM CASSETTE MUSIQUE

Circuits intégrés	TTL	Transistors	Potentiomètre	Résistances 1/4W 5%	Résistances 1/2W 5%
6522 (1) 2708 (2)	74LS138 (1) 7406 (1)	BD 137 (1) BD 138 (1) BC 170 (2) 2N3819 (1) 2N2222A (1) 2N2907 (1)	10 KΩ (5)	10Ω (2) 47Ω (2) 470Ω (2) 560Ω (1) 1KΩ (8) 1,2KΩ (2) 2,2KΩ (1) 22KΩ (3) 3,3KΩ (2) 4,7KΩ (1) 6,8KΩ (4) 3,9KΩ (1) 10KΩ (4) 15KΩ (6) 33KΩ (2) 47KΩ (3) 56KΩ (1) 75KΩ (1) 82KΩ (1) 11KΩ (2) 68KΩ (1) 100KΩ (7) 200KΩ (1) 820KΩ (2) 1MΩ (1)	820Ω (1) 300Ω (1) 33KΩ (1) 43KΩ (1)
Amplis Dual bifet (1) quadruple bifet (6)	**Supports C.I.** 8p (1) 14p (7) 16p (1) 24p (2) 40p (1)	**Broches** Wrapping UMD Cuivre vernis épargne	**Résistances 1%** 11,8Ω (1) 680 Ω (1) 787 Ω (1) 909 Ω (1) 1,43KΩ (1) 3,32KΩ (1) 3,4K Ω (6) 6,65KΩ (1) 6,81KΩ (10) 8,87KΩ (1) 11KΩ (1) 22,1KΩ (2) 27,4KΩ (1) 59KΩ (1) 66,5KΩ (1) 76,8KΩ (1) 88,7KΩ (1) 137KΩ (1) 442KΩ (1)		**Capacités** chim 100µf (1) chim 2,2µf (2) T.G.6,8µf (17) T.G.2,2µf (1) chim 6,8µf (2) 0,1µf (5) 2,2 nf (3) 4,7 nf (1) 10 nf (6)
Diodes 1N4148 (13)					

NOMENCLATURE DE LA CARTE E/S

Circuits intégrés	TTL	Support C.I.	Résistance
2716 (8) 6522 (1) 8279 (2)	74LS138 (2) 74LS139 (1)	14 p (1) 16 p (4) 24 p (8) 40 p (3)	1/4W 1KΩ (1)
CMOS 4049 (1)	Connecteur OEC (3) Broches Wrapping UMD Cuivre vernis épargne	Capacité 1 μf TG (6) 10 μf chimique (1)	

NOMENCLATURE CARTE BUS

Connecteurs HE901 (9)
Cossesfaston (5)
Cuivre vernis épargne

NOMENCLATURE GOUPIL : ALIMENTATION A TRANSFORMATEUR GOUPIL 2

Radiateur (1) Plaque (1) Transformateur (1)	Barettes (1) Visserie Cosses	Ponts redresseur 25 A (1) 10 A (2)
Régulateurs 5 V - 5 A (1) 5 V - 1,5 A (1) 12 V - 1,5 A (2)	Condensateurs 3300µf/16 V (1) 1500µf/16 V (1) 6800µf/25 V (2) 470µf/16 V (4)	

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 action devra être inférieure à 50 watt.

NOMENCLATURE DE LA CARTE MEMOIRE DYNAMIQUE 65 K

TTL	
74LS04	(1)
74LS20	(2)
74LS86	(1)
74LS133	(1)
74LS244	(2)
74LS251	(1)
74LS259	(1)
NE555	(1)
DM8837	(1)
MC3480	(1)
MC3242	(1)
TOKO 526P	(1)

Support	
8p	(1)
14p	(5)
16p	(36)
20p	(2)
24p	(1)
28p	(1)

Résistance	
3,3K	(3)
22 Ω	(6)
390Ω	(1)
330Ω	(3)
590Ω	(1)

Condensateur	
47 nf	(106)
10 nf	(1)
100 µf	(3)

Mémoires
4116-2 (4x8)

1 circuit vernis épargne

NOMENCLATURE DE LA CARTE FLOPPY 5" DOUBLE FACE

TTL		Supports		Résistance 1/4W		Condensateur	
74S04	(1)	8 p	(2)	150Ω	(4)	10 nf	(4)
7406	(1)	14 p	(5)	1KΩ	(3)	68 pf	(1)
7407	(1)	16 p	(8)	4,7KΩ	(3)	10 µf	(1)
74LS04	(1)	40 p	(1)	10KΩ	(2)	100 µf	(2)
74LS74	(2)			750KΩ	(1)	T.G.1µf	(13)
74LS123	(1)			1MΩ	(1)	T.G.0,1µf	(1)
74LS138	(2)			12KΩ	(1)		
74LS139	(1)						
74LS175	(1)						
74LS193	(1)						
74LS367	(1)						
NE555	(2)						

Connecteur

OEG 2x17pts (1)

Transistor

2N2905 (1)

Quartz 8 Mhz (1)

capa variable (1)

1791 (1)

1 circuit vernis épargne
sérigraphie

NOMENCLATURE DE LA CARTE GRAPHIQUE 8 COULEURS (256 x 256 points)

Support 14 p (16) Support 16 p (24) Support 20 p (4) Support 40 p (1) Support UMD (1)	TTL 74LS122 (1) 74LS05 (1) 74LS157 (2) 74LS32 (3) 74LS74 (2) 74LS175 (1) 74LS139 (1) 74LS00 (1) 74LS16 (1) 74LS04 (1) 74LS30 (2) 7LLS273 (3) 74LS244 (1) 74LS08 (1) 74LS195 (3) 7404 (1) 74153 (4) 7408 (1) 7414 (1)	Condensateur 10 pf (2) 47 nf (1) 10 nf 15 4,7 µf 9 47 µf (1) 10 µf (2) 100 nf (8)	Résistance 1/4W 5% 150 Ω (6) 2,2K Ω (3) 470 Ω (2) 75 Ω (2) 330 Ω (2) 33K Ω (1) 33 Ω (5) 1/8W 5% 68 Ω (8)
6845 (1) 4116 (12)		Transistor 2N222A (1)	Plots cometel (7)
Quartz 10,358 (1)		Potentiomètre 1KΩ (1)	Cosses faston 2.8 (2)
Circuit imprimé épargne sérigraphie (1)			
Câble péritel (1)			

NOMENCLATURE DE LA CARTE CONTROLEUR 24 x 80 (Vidéo)

Support

14 p	(8)
16 p	(7)
24 p	(1)
18 p	(4)
20 p	(2)
40 p	(1)

C.I.

6845	(1)
2114	(4)
2716	(1)

Cosses faston (2)

TTL

74LS74	(1)
74LS93	(1)
74LS20	(1)
74LS138	(1)
74LS86	(1)
74LS165	(1)
74LS273	(1)
74LS245	(1)
74LS157	(3)
74LS221	(1)
74LS08	(1)
74LS00	(2)
74LS04	(1)
74LS175	(1)

Condensateur

68 μf	(1)
10 nf	(1)
47 nf	(1)
T.G. 1 μf	(13)
Cond. Variable	(1)

Transistor 2N3866 (1)

Quartz 13 Mhz (1)

Circuit imprimé
verni épargne
Sérigraphie (1)

Résistance

1 KΩ	(2)
27 KΩ	(1)
10 KΩ	(2)
4,7KΩ	(1)
5,6KΩ	(2)
2,2KΩ	(1)
6,8KΩ	(1)
75 Ω	(1)

NOMENCLATURE DE LA CARTE CONTROLEUR FLOPPY 8" 5" DOUBLE DENSITE

TTL		CMOS		Diode		Résistance		
74LS04	(1)	4049	(1)	1N4148	(2)	1/4W 5 %	150Ω	(5)
74LS08	(1)						470Ω	(1)
74LS121	(1)						1KΩ	(5)
74LS00	(4)	C.I.		Condensateur			4,7KΩ	(2)
74LS320	(2)	6844	(1)				10KΩ	(6)
74LS138	(1)	1771	(1)	T.G. 6,8µf	(3)		47KΩ	(1)
74LS175	(1)			22 nf	(1)		120KΩ	(1)
74LS139	(1)			220 nf	(1)		910KΩ	(1)
74LS123	(1)	Support		22 pf	(2)		390 Ω	(1)
74LS180	(1)	14 p	(14)	56 pf	(1)	1/4W 1%	7,5KΩ	(2)
74LS74	(2)	16 p	(6)	470 pf	(3)			
74LS86	(1)	20 p	(6)	100 pf	(2)			
74LS163	(1)	40 p	(2)	10 nf	(3)	Réseau résistance 10 KΩ		
74LS240	(2)							
74LS245	(2)			Barette à Wrapper				
74LS244	(1)	Quartz 4 Mhz (1)				Circuit imprimé vernis épargne sérigraphie		(1)
74LS273	(1)			Transistor BC1073 (2)				
74LS 125	(1)			Connecteur 3 M 50 brakes				

NOMEMCLATURE DE LA CARTE FLOPPY (DISQUES)

Carte Floppy (1) 5" ou 8"	Connectique Alim. floppy (4) Cordon secteur (1) Cordon controleur (2)
Boite Drives 5" ou 8" (2) Alimentation Floppy	Ventilateur (1)
Documentation (2) Disquette DOS-FLEX (1)	Boutons Interrupteur Entrée/Secteur

2/ FICHES TECHNIQUES DE QUELQUES PRINCIPAUX COMPOSANTS

Fiche technique :

ACIA 6551

Asynchronous Communication Interface Adapter

Synertek INCORPORATED

SY6551

MICROPROCESSOR PRODUCTS

(PRELIMINARY)

SEPTEMBER 1978

- On-chip baud rate generator: 15 programmable baud rates derived from a standard 1.8432 MHz external crystal (50 to 19,200 baud).
- Programmable interrupt and status register to simplify software design.
- Single +5 volt power supply.
- Serial echo mode.
- False start bit detection.
- 8-bit bi-directional data bus for direct communication with the microprocessor.
- External 16x clock input for non-standard baud rates (up to 125 Kbaud).
- Programmable: word lengths; number of stop bits; and parity bit generation and detection.
- Data set and modem control signals provided.
- Parity: (odd, even, none, mark, space).
- Full-duplex or half-duplex operation.
- 5, 6, 7, 8 and 9 bit transmission.

The SY6551 is an Asynchronous Communication Adapter (ACIA) intended to provide for interfacing the 6500/6800 microprocessor families to serial communication data sets and modems. A unique feature is the inclusion of an on-chip programmable baud rate generator, with a crystal being the only external component required.

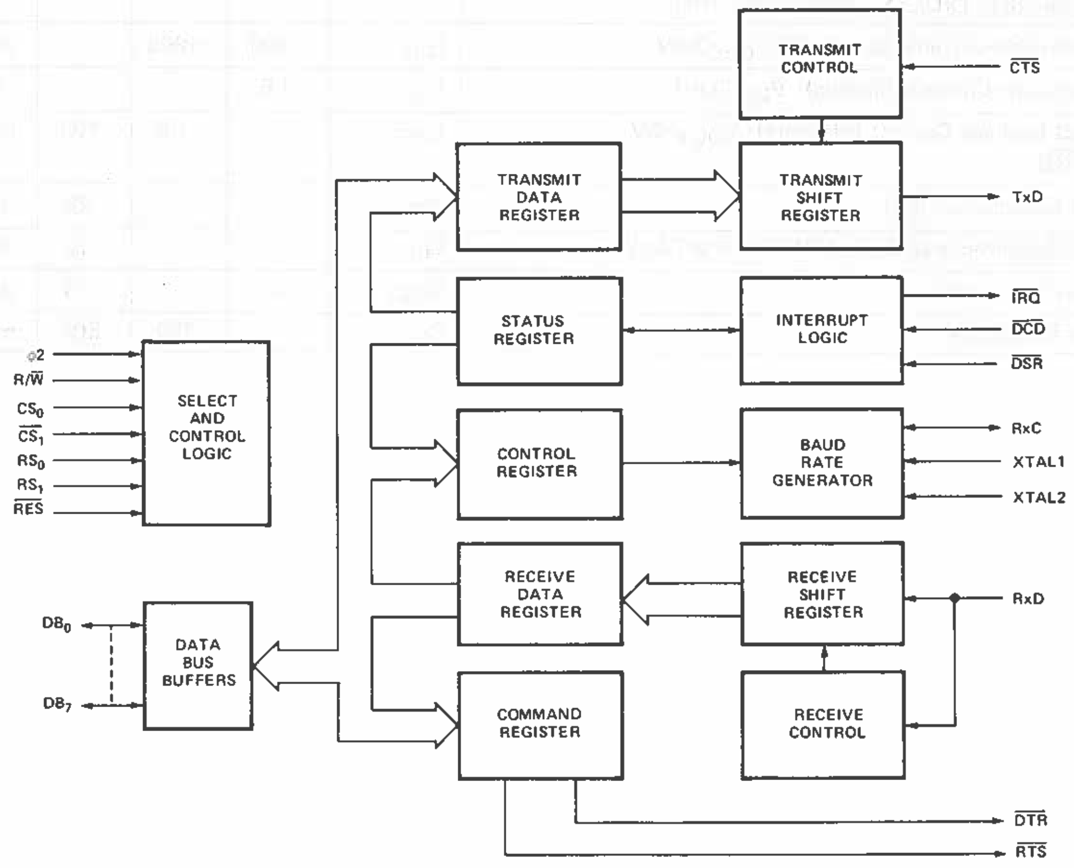


Figure 1. Block Diagram

ABSOLUTE MAXIMUM RATINGS

Rating	Symbol	Allowable Range
Supply Voltage	V_{CC}	-0.3V to +7.0V
Input/Output Voltage	V_{IN}	-0.3V to +7.0V
Operating Temperature	T_{OP}	0°C to 70°C
Storage Temperature	T_{STG}	-55°C to 150°C

All inputs contain protection circuitry to prevent damage to high static charges. Care should be exercised to prevent unnecessary application of voltages in excess of the allowable limits.

Stresses above 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.

ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0V ± 5%, T_A = 25°C, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Input High Voltage	V_{IH}	2.0	–	V_{CC}	V
Input Low Voltage	V_{IL}	-0.3	–	0.8	V
Input Leakage Current: V_{IN}=0 to 5V. ($\phi2$, R/$\overline{W}$, $\overline{RES}$, CS_0, $\overline{CS_1}$, RS_0, RS_1, $\overline{CTS}$, RxD, $\overline{DCD}$, $\overline{DSR}$)	I_{IN}	–	±1.0	±2.5	μA
Input Leakage Current for High Impedance State (Three State)	I_{TSI}	–	±2.0	±10.0	μA
Output High Voltage: I_{LOAD}=-100μA	V_{OH}	2.4	–	–	V
Output Low Voltage: I_{LOAD}=1.6mA (DB_0-DB_7, TxD, RxC, $\overline{RTS}$, $\overline{DTR}$, $\overline{IRQ}$)	V_{OL}	–	–	0.4	V
Output High Current (Sourcing): V_{OH}=2.4V	I_{OH}	-100	-1000	–	μA
Output Low Current (Sinking): V_{OL}=0.4V	I_{OL}	1.6	–	–	μA
Output Leakage Current (off state): V_{OUT}=5V ($\overline{IRQ}$)	I_{OFF}	–	1.0	10.0	μA
Clock Capacitance ($\phi2$)	C_{CLK}	–	–	20	pF
Input Capacitance (except XTAL1 and XTAL2)	C_{IN}	–	–	10	pF
Output Capacitance	C_{OUT}	–	–	10	pF
Power Dissipation	P_D	–	350	500	mw

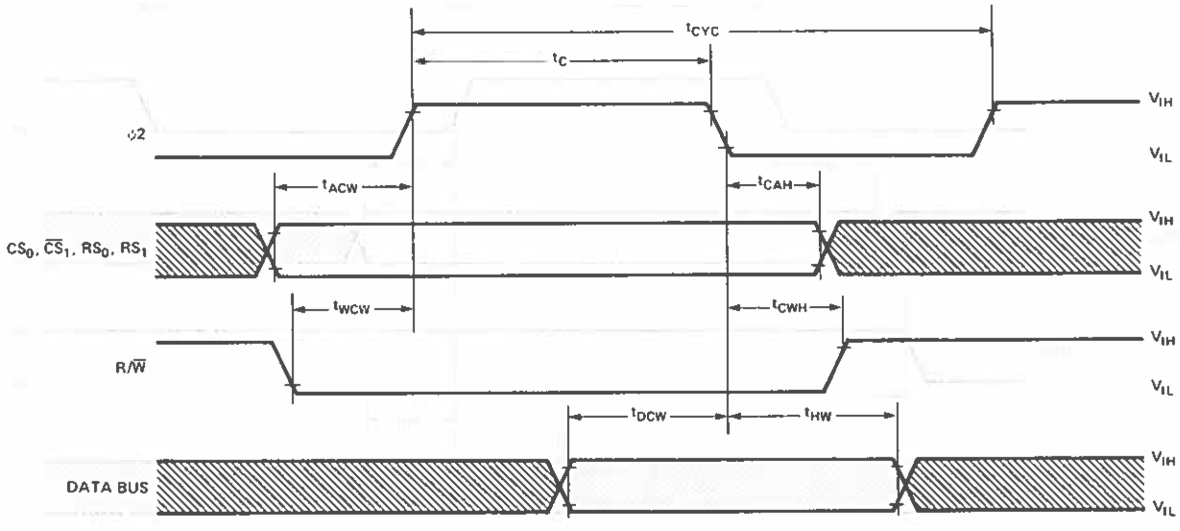


Figure 2. Write Timing Characteristics

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

Characteristic	Symbol	SY6551		SY6551A		Unit
		Min	Max	Min	Max	
Cycle Time	t_{CYC}	1.0	40	0.5	40	µs
$\phi 2$ Pulse Width	t_C	470	–	235	–	ns
Address Set-Up Time	t_{ACW}	180	–	90	–	ns
Address Hold Time	t_{CAH}	0	–	0	–	ns
R/$\overline{W}$ Set-Up Time	t_{WCW}	180	–	90	–	ns
R/$\overline{W}$ Hold Time	t_{CWH}	0	–	0	–	ns
Data Bus Set-Up Time	t_{DCW}	300	–	150	–	ns
Data Bus Hold Time	t_{HW}	10	–	10	–	ns

(t_r and t_f = 10 to 30 ns)

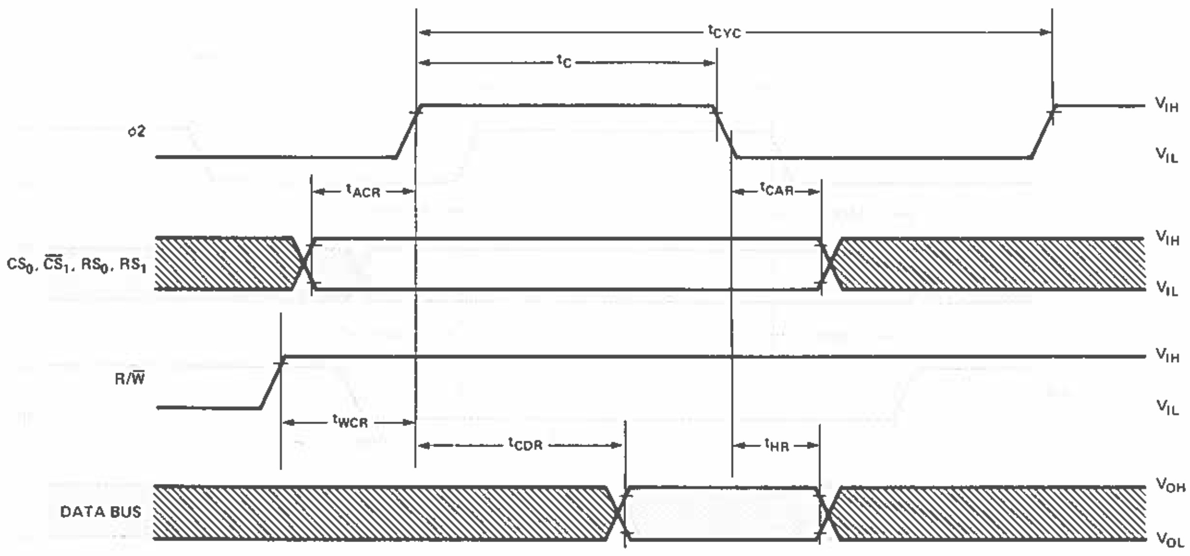
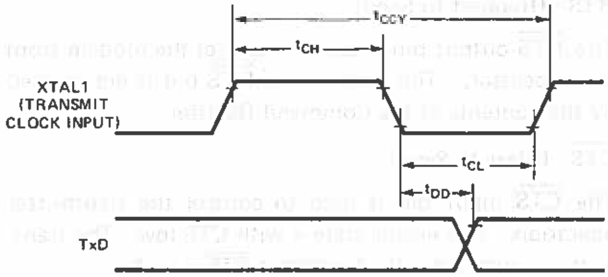


Figure 3. Read Timing Characteristics

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

Characteristic	Symbol	SY6551 Min	SY6551 Max	SY6551A Min	SY6551A Max	Unit
Cycle Time	t_{CYC}	1.0	40	0.5	40	μs
Pulse Width (φ2)	t_C	470	–	235	–	ns
Address Set-Up Time	t_{ACR}	180	–	90	–	ns
Address Hold Time	t_{CAR}	0	–	0	–	ns
R/W̄ Set-Up Time	t_{WCR}	180	–	90	–	ns
Read Access Time	t_{CDR}	–	395	–	200	ns
Read Hold Time	t_{HR}	10	–	10	–	ns

(t_r and t_f = 10 to 30 ns)



NOTE: TxD rate is 1/16 TxC rate.

Figure 4a. Transmit Timing with External Clock

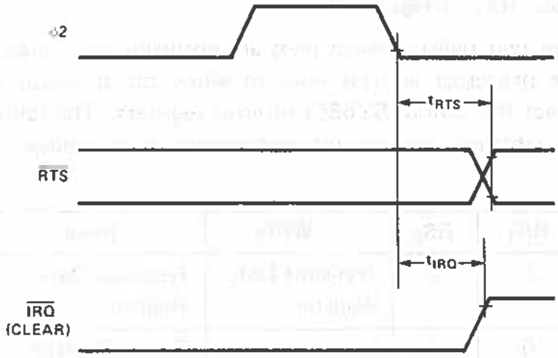
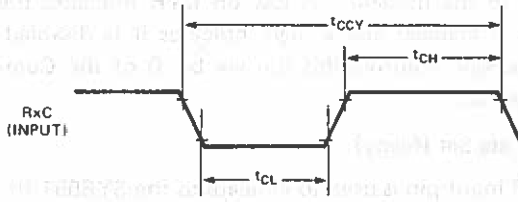


Figure 4b. Interrupt and RTS Timing



NOTE: RxD rate is 1/16 RxC rate.

Figure 4c. Receive External Clock Timing

TRANSMIT/RECEIVE CHARACTERISTICS

Characteristic	Symbol	SY6551 Min	SY6551 Max	SY6551A Min	SY6551A Max	Unit
Transmit/Receive Clock Rate	t_{CCY}	0.5*	–	0.5*	–	μs
Transmit/Receive Clock High Time	t_{CH}	235	–	235	–	ns
Transmit/Receive Clock Low Time	t_{CL}	235	–	235	–	ns
XTAL1 to TxD Propagation Delay	t_{DD}	–	500	–	500	ns
$\overline{\text{RTS}}$ Propagation Delay	t_{RTS}	–	500	–	500	ns
$\overline{\text{IRQ}}$ Propagation Delay (Clear)	t_{IRQ}	–	500	–	500	ns

(t_r, t_f = 10 to 30 nsec)

*The baud rate with external clocking is: Baud Rate = $\frac{1}{16 \times T_{CCY}}$

INTERFACE SIGNAL DESCRIPTION

$\overline{\text{RES}}$ (Reset)

During system initialization a low on the $\overline{\text{RES}}$ input will cause internal registers to be cleared.

$\phi 2$ (Input Clock)

The input clock is the system $\phi 2$ clock and is used to trigger all data transfers between the system microprocessor and the SY6551.

R/$\overline{\text{W}}$ (Read/Write)

The R/$\overline{\text{W}}$ is generated by the microprocessor and is used to control the direction of data transfers. A high on the R/$\overline{\text{W}}$ pin allows the processor to read the data supplied by the SY6551. A low on the R/$\overline{\text{W}}$ pin allows a write to the SY6551.

$\overline{\text{IRQ}}$ (Interrupt Request)

The $\overline{\text{IRQ}}$ pin is an interrupt signal from the interrupt control logic. It is an open drain output, permitting several devices to be connected to the common $\overline{\text{IRQ}}$ microprocessor input. Normally a high level, $\overline{\text{IRQ}}$ goes low when an interrupt occurs.

DB_0 - DB_7 (Data Bus)

The DB_0-DB_7 pins are the eight data lines used for transfer of data between the processor and the SY6551. These lines are bi-directional and are normally high-impedance except during Read cycles when selected.

CS_0, $\overline{CS_1}$ (Chip Selects)

The two chip select inputs are normally connected to the processor address lines either directly or through decoders. The SY6551 is selected when CS_0 is high and $\overline{CS_1}$ is low.

RS_ϕ, RS_1 (Register Selects)

The two register select lines are normally connected to the processor address lines to allow the processor to select the various SY6551 internal registers. The following table indicates the internal register select coding:

RS_1	RS_0	Write	Read
0	0	Transmit Data Register	Receiver Data Register
0	1	Programmed Reset (Data is "Don't Care")	Status Register
1	0	Command Register	
1	1	Control Register	

The table shows that only the Command and Control registers are read/write. The Programmed Reset operation does not cause any data transfer, but is used to clear the SY6551 registers. The Programmed Reset is slightly different from the Hardware Reset ($\overline{RES}$) and these differences are described in the individual register definitions.

ACIA/MODEM INTERFACE SIGNAL DESCRIPTION

XTAL1, XTAL2 (Crystal Pins)

These pins are normally directly connected to the external crystal (1.8432 MHz) used to derive the various baud rates. Alternatively, an externally generated clock may be used to drive the XTAL1 pin, in which case the XTAL2 pin must float. XTAL1 is the input pin for the transmit clock.

TxD (Transmit Data)

The TxD output line is used to transfer serial NRZ (non-return-to-zero) data to the modem. The LSB (least significant bit) of the Transmit Data Register is the first data bit transmitted and the rate of data transmission is determined by the baud rate selected.

RxD (Receive Data)

The RxD input line is used to transfer serial NRZ data into the ACIA from the modem, LSB first. The receiver data rate is either the programmed baud rate or the rate of an externally generated receiver clock. This selection is made by programming the Control Register.

RxC (Receive Clock)

The RxC is a bi-directional pin which serves as either the receiver 16x clock input or the receiver 16x clock output. The latter mode results if the internal baud rate generator is selected for receiver data clocking.

$\overline{RTS}$ (Request to Send)

The $\overline{RTS}$ output pin is used to control the modem from the processor. The state of the $\overline{RTS}$ pin is determined by the contents of the Command Register.

$\overline{CTS}$ (Clear to Send)

The $\overline{CTS}$ input pin is used to control the transmitter operation. The enable state is with $\overline{CTS}$ low. The transmitter is automatically disabled if $\overline{CTS}$ is high.

$\overline{DTR}$ (Data Terminal Ready)

This output pin is used to indicate the status of the SY6551 to the modem. A low on $\overline{DTR}$ indicates the SY6551 is enabled and a high indicates it is disabled. The processor controls this pin via bit 0 of the Command Register.

$\overline{DSR}$ (Data Set Ready)

The $\overline{DSR}$ input pin is used to indicate to the SY6551 the status of the modem. A low indicates the "ready" state and a high, "not-ready".

$\overline{DCD}$ (Data Carrier Detect)

The $\overline{DCD}$ input pin is used to indicate to the SY6551 the status of the carrier-detect output of the modem. A low indicates that the modem carrier signal is present and a high, that it is not.

INTERNAL ORGANIZATION

The Transmitter/Receiver sections of the SY6551 are depicted by the block diagram in Figure 5.

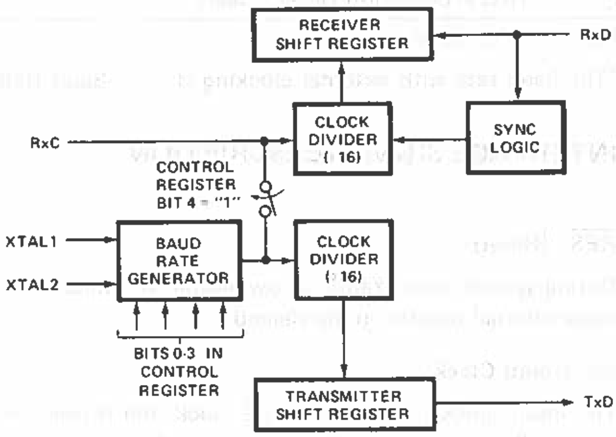


Figure 5. Transmitter/Receiver Clock Circuits

Bits 0-3 of the Control Register select the divisor used to generate the baud rate for the Transmitter. If the Receiver clock is to use the same baud rate as the Transmitter, then RxC becomes an output pin and can be used to slave other circuits to the SY6551.

CONTROL REGISTER

The Control Register is used to select the desired mode for the SY6551. The word length, number of stop bits, and clock controls are all determined by the Control Register, which is depicted in Figure 6.

CONTROL REGISTER

7	6	5	4	3	2	1	0

STOP BITS (bit 7)

0 = 1 Stop Bit
1 = 2 Stop Bits
1 Stop Bit if Word Length = 8 Bits and Parity*
1½ Stop Bits if Word Length = 5 Bits and No Parity.

WORD LENGTH (bits 6, 5)

BIT 6	BIT 5	DATA WORD LENGTH
0	0	8
0	1	7
1	0	6
1	1	5

RECEIVER CLOCK SOURCE (bit 4)

0 = External Receiver Clock
1 = Baud Rate Generator

BAUD RATE GENERATOR (bits 3–0)

3	2	1	0	
0	0	0	0	16x EXTERNAL CLOCK
0	0	0	1	50 BAUD
0	0	1	0	75
0	0	1	1	109.92
0	1	0	0	134.58
0	1	0	1	150
0	1	1	0	300
0	1	1	1	600
1	0	0	0	1200
1	0	0	1	1800
1	0	1	0	2400
1	0	1	1	3600
1	1	0	0	4800
1	1	0	1	7200
1	1	1	0	9600
1	1	1	1	19,200

*This allows for 9-bit transmission (8 data bits plus parity).

	7	6	5	4	3	2	1	0
HARDWARE RESET	0	0	0	0	0	0	0	0
PROGRAM RESET	–	–	–	–	–	–	–	–

Figure 6. Control Register Format

COMMAND REGISTER

The Command Register is used to control Specific Transmit/Receive functions and is shown in Figure 7.

COMMAND REGISTER

7	6	5	4	3	2	1	0

PARITY CHECK CONTROLS (bits 7, 6, 5)

BIT 7	BIT 6	BIT 5	OPERATION
–	–	0	Parity Disabled - No Parity Bit Generated - No Parity Bit Received
0	0	1	Odd Parity Receiver and Transmitter
0	1	1	Even Parity Receiver and Transmitter
1	0	1	Mark Parity Bit Transmitted, Parity Check Disabled
1	1	1	Space Parity Bit Transmitted, Parity Check Disabled

NORMAL/ECHO MODE FOR RECEIVER (bit 4)

0 = Normal
1 = Echo

TRANSMITTER CONTROLS (bits 3, 2)

BIT 3	BIT 2	TRANSMIT INTERRUPT	$\overline{RTS}$ LEVEL	OTHER
0	0	Disabled	High	–
0	1	Enabled	Low	–
1	0	Disabled	Low	–
1	1	Disabled	Low	Transmit BRK

RECEIVER INTERRUPT ENABLE (bit 1)

0 = $\overline{IRQ}$ Interrupt Enabled from Bit 0 of Status Register
1 = $\overline{IRQ}$ Interrupt Disabled

DATA TERMINAL READY (bit 0)

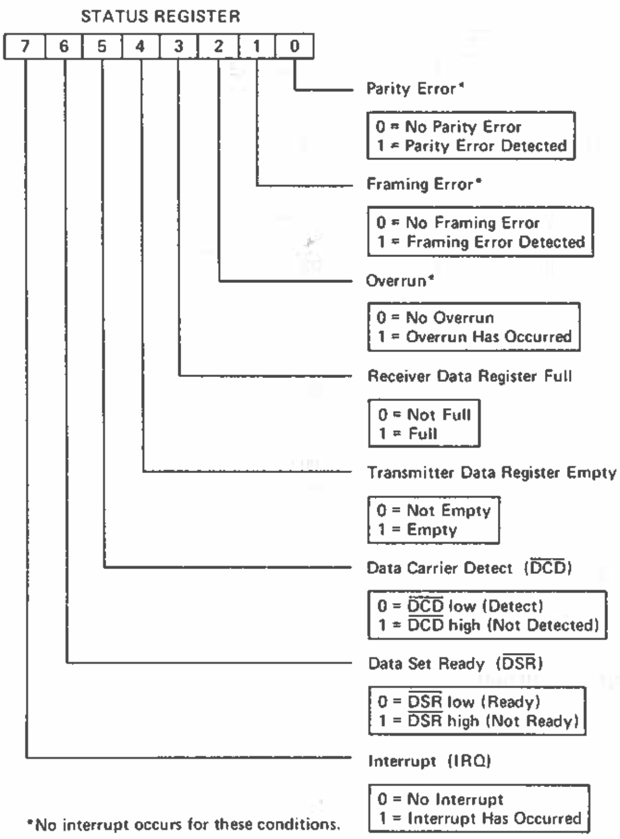
0 = Disable Receiver/Transmitter ($\overline{DTR}$ high)
1 = Enable Receiver/Transmitter ($\overline{DTR}$ low)

	7	6	5	4	3	2	1	0
HARDWARE RESET	0	0	0	0	0	0	1	0
PROGRAM RESET	–	–	–	0	0	0	1	0

Figure 7. Command Register Format

STATUS REGISTER

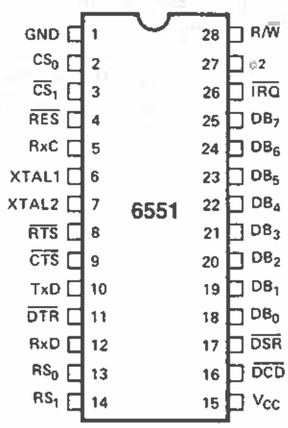
The Status Register is used to indicate to the processor the status of various SY6551 functions and is outlined in Figure 8.



	7	6	5	4	3	2	1	0
HARDWARE RESET	0	–	–	1	0	0	0	0
PROGRAM RESET	–	–	–	–	–	0	–	–

Figure 8. Status Register Format

PIN CONFIGURATION



TRANSMIT AND RECEIVE DATA REGISTERS

These registers are used as temporary data storage for the 6551 Transmit and Receive circuits. The Transmit Data Register is characterized as follows:

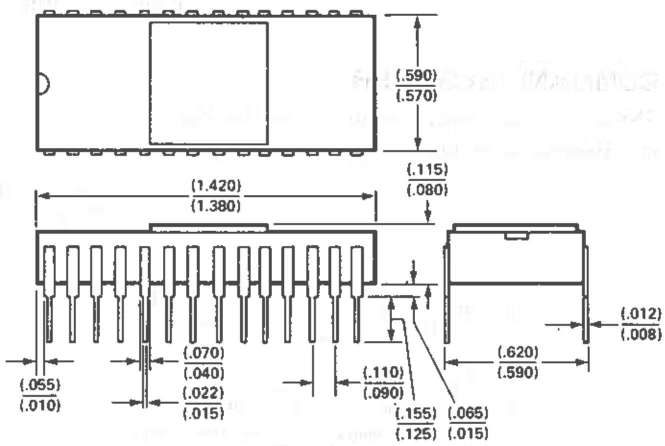
- Bit 0 is the leading bit to be transmitted.
- Unused data bits are the high-order bits and are "don't care" for transmission.

The Receive Data Register is characterized in a similar fashion:

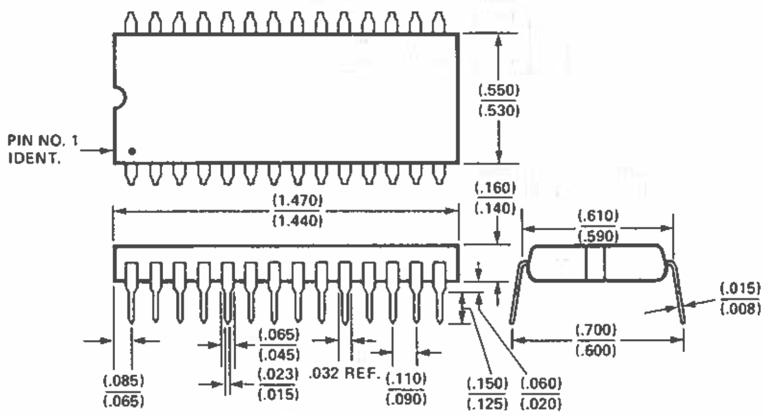
- Bit 0 is the leading bit received.
- Unused data bits are the high-order bits and are "0" for the receiver.
- Parity bits are not contained in the Receive Data Register, but are stripped-off after being used for external parity checking. Parity and all unused high-order bits are "0".

PACKAGE OUTLINES

28 LEAD CERAMIC



28 LEAD PLASTIC



Fiche technique :

VIA 6522

Synertek INCORPORATED

Versatile Interface Adapter (VIA)

SY6522 SY6522A

MICROPROCESSOR PRODUCTS

Preliminary

- Two 8-Bit Bidirectional I/O Ports
- Two 16-Bit Programmable Timer/Counters
- Serial Data Port
- Single +5V Power Supply
- TTL Compatible
- CMOS Compatible Peripheral Control Lines
- Expanded "Handshake" Capability Allows Positive Control of Data Transfers Between Processor and Peripheral Devices
- Latched Output and Input Registers
- 1 MHz and 2 MHz Operation

The SY6522 Versatile Interface Adapter (VIA) is a very flexible I/O control device. In addition, this device contains a pair of very powerful 16-bit interval timers, a serial-to-parallel/parallel-to-serial shift register and input data latching on the peripheral ports. Expanded handshaking capability allows control of bi-directional data transfers between VIA's in multiple processor systems.

Control of peripheral devices is handled primarily through two 8-bit bi-directional ports. Each line can be programmed as either an input or an output. Several peripheral I/O lines can be controlled directly from the interval timers for generating programmable frequency square waves or for counting externally generated pulses. To facilitate control of the many powerful features of this chip, an interrupt flag register, an interrupt enable register and a pair of function control registers are provided.

MICRO-PROCESSORS

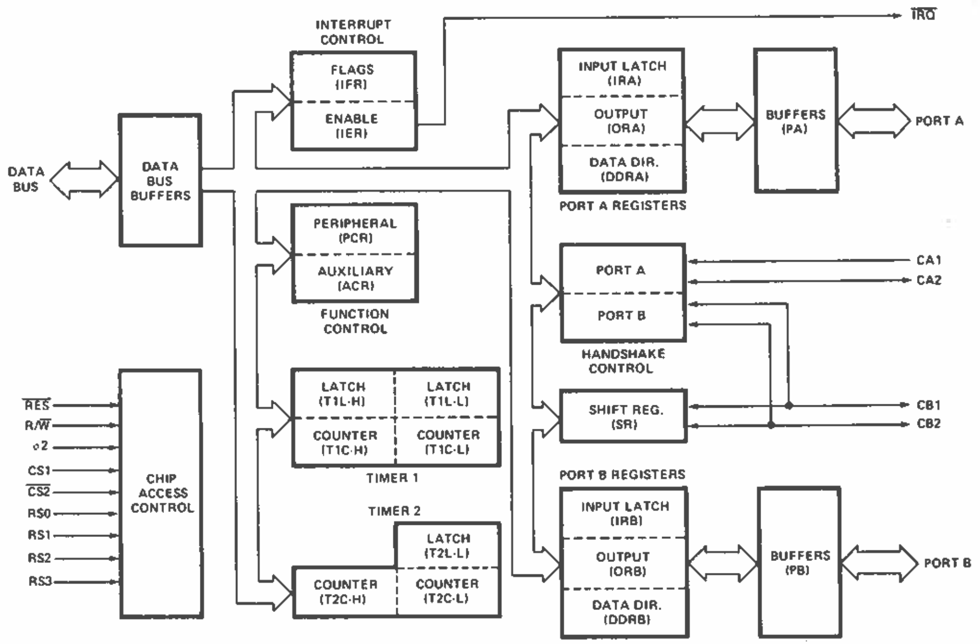


Figure 1. SY6522 Block Diagram

ABSOLUTE MAXIMUM RATINGS

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

This device contains circuitry to protect the inputs against damage due to high static voltages. However, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages.

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

Symbol	Characteristic	Min.	Max.	Unit
V_{IH}	Input High Voltage (all except φ2)	2.4	V_{CC}	V
V_{CH}	Clock High Voltage	2.4	V_{CC}	V
V_{IL}	Input Low Voltage	-0.3	0.4	V
I_{IN}	Input Leakage Current — V_{IN} = 0 to 5 Vdc R/$\overline{W}$, $\overline{RES}$, RS0, RS1, RS2, RS3, CS1, $\overline{CS2}$, CA1, Φ2	–	±2.5	μA
I_{TSI}	Off-state Input Current — V_{IN} = .4 to 2.4V V_{CC} = Max, D0 to D7	–	±10	μA
I_{IH}	Input High Current — V_{IH} = 2.4V PA0-PA7, CA2, PB0-PB7, CB1, CB2	-100	–	μA
I_{IL}	Input Low Current — V_{IL} = 0.4 Vdc PA0-PA7, CA2, PB0-PB7, CB1, CB2	–	-1.6	mA
V_{OH}	Output High Voltage V_{CC} = min, I_{load} = -100 μAdc PA0-PA7, CA2, PB0-PB7, CB1, CB2	2.4	–	V
V_{OL}	Output Low Voltage V_{CC} = min, I_{load} = 1.6 mAdc	–	0.4	V
I_{OH}	Output High Current (Sourcing) V_{OH} = 2.4V V_{OH} = 1.5V (PB0-PB7)	 -100 -1.0	 – –	 μA mA
I_{OL}	Output Low Current (Sinking) V_{OL} = 0.4 Vdc	1.6	–	mA
I_{OFF}	Output Leakage Current (Off state) $\overline{IRQ}$	–	10	μA
C_{IN}	Input Capacitance — T_A = 25°C, f = 1 MHz (R/$\overline{W}$, $\overline{RES}$, RS0, RS1, RS2, RS3, CS1, $\overline{CS2}$, D0-D7, PA0-PA7, CA1, CA2, PB0-PB7) (CB1, CB2) (Φ2 Input)	– – –	7.0 10 20	pF pF pF
C_{OUT}	Output Capacitance — T_A = 25°C, f = 1 MHz	–	10	pF
P_D	Power Dissipation	–	700	mW

MICRO-PROCESSORS

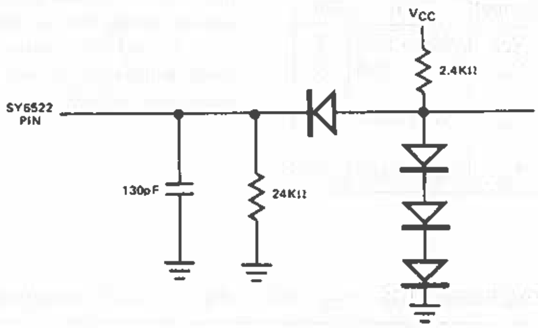


Figure 2. Test Load (for all Dynamic Parameters)

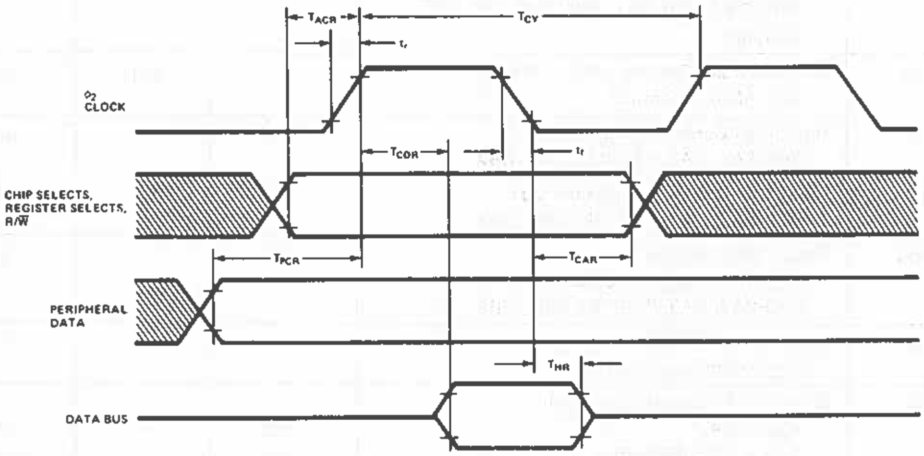


Figure 3. Read Timing Characteristics

READ TIMING CHARACTERISTICS (FIGURE 3)

Symbol	Parameter	SY6522 Min.	SY6522 Max.	SY6522A Min.	SY6522A Max.	Unit
T_{CY}	Cycle Time	1	50	0.5	50	μs
T_{ACR}	Address Set-Up Time	180	–	90	–	ns
T_{CAR}	Address Hold Time	0	–	0	–	ns
T_{PCR}	Peripheral Data Set-Up Time	300	–	300	–	ns
T_{CDR}	Data Bus Delay Time	–	395	–	200	ns
T_{HR}	Data Bus Hold Time	10	–	10	–	ns

NOTE: tr, tf = 10 to 30ns.

MICRO-PROCESSORS

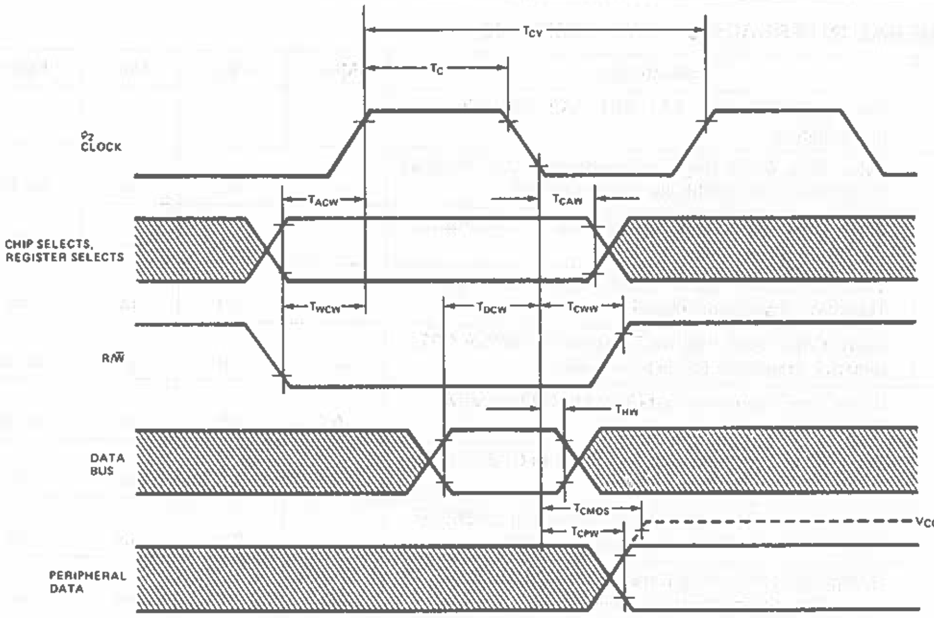


Figure 4. Write Timing Characteristics

WRITE TIMING CHARACTERISTICS (FIGURE 4)

Symbol	Parameter	SY6522 Min.	SY6522 Max.	SY6522A Min.	SY6522A Max.	Unit
T_{CY}	Cycle Time	1	50	0.50	50	μs
T_C	φ2 Pulse Width	0.47	25	0.25	25	μs
T_{ACW}	Address Set-Up Time	180	–	90	–	ns
T_{CAW}	Address Hold Time	0	–	0	–	ns
T_{WCW}	R/W̄ Set-Up Time	180	–	90	–	ns
T_{CWW}	R/W̄ Hold Time	0	–	0	–	ns
T_{DCW}	Data Bus Set-Up Time	300	–	150	–	ns
T_{HW}	Data Bus Hold Time	10	–	10	–	ns
T_{CPW}	Peripheral Data Delay Time	–	1.0	–	1.0	μs
T_{CMOS}	Peripheral Data Delay Time to CMOS Levels	–	2.0	–	2.0	μs

NOTE: tr, tf = 10 to 30ns.

MICRO-PROCESSORS

PERIPHERAL INTERFACE CHARACTERISTICS

Symbol	Characteristic	Min.	Max.	Unit	Figure
t_r, t_f	Rise and Fall Time for CA1, CB1, CA2, and CB2 Input Signals	–	1.0	μs	–
T_{CA2}	Delay Time, Clock Negative Transition to CA2 Negative Transition (read handshake or pulse mode)	–	1.0	μs	5a, 5b
T_{RS1}	Delay Time, Clock Negative Transition to CA2 Positive Transition (pulse mode)	–	1.0	μs	5a
T_{RS2}	Delay Time, CA1 Active Transition to CA2 Positive Transition (handshake mode)	–	2.0	μs	5b
T_{WHS}	Delay Time, Clock Positive Transition to CA2 or CB2 Negative Transition (write handshake)	–	1.0	μs	5c, 5d
T_{DS}	Delay Time, Peripheral Data Valid to CB2 Negative Transition	0	1.5	μs	5c, 5d
T_{RS3}	Delay Time, Clock Positive Transition to CA2 or CB2 Positive Transition (pulse mode)	–	1.0	μs	5c
T_{RS4}	Delay Time, CA1 or CB1 Active Transition to CA2 or CB2 Positive Transition (handshake mode)	–	2.0	μs	5d
T_{IL}	Set-up Time, Peripheral Data Valid to CA1 or CB1 Active Transition (input latching)	300	–	ns	5e
T_{SR1}	Shift-Out Delay Time – Time from ϕ_2 Falling Edge to CB2 Data Out	–	300	ns	5f
T_{SR2}	Shift-In Setup Time – Time from CB2 Data In to ϕ_2 Rising Edge	300	–	ns	5g
T_{IPW}	Pulse Width – PB6 Input Pulse	2	–	μs	5i
T_{ICW}	Pulse Width – CB1 Input Clock	2	–	μs	5h
I_{IPS}	Pulse Spacing – PB6 Input Pulse	2	–	μs	5i
I_{ICS}	Pulse Spacing – CB1 Input Pulse	2	–	μs	5h

MICRO-PROCESSORS

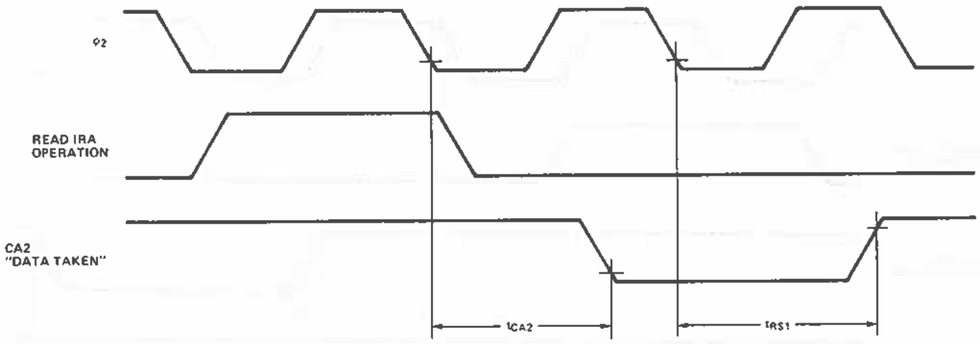


Figure 5a. CA2 Timing for Read Handshake, Pulse Mode

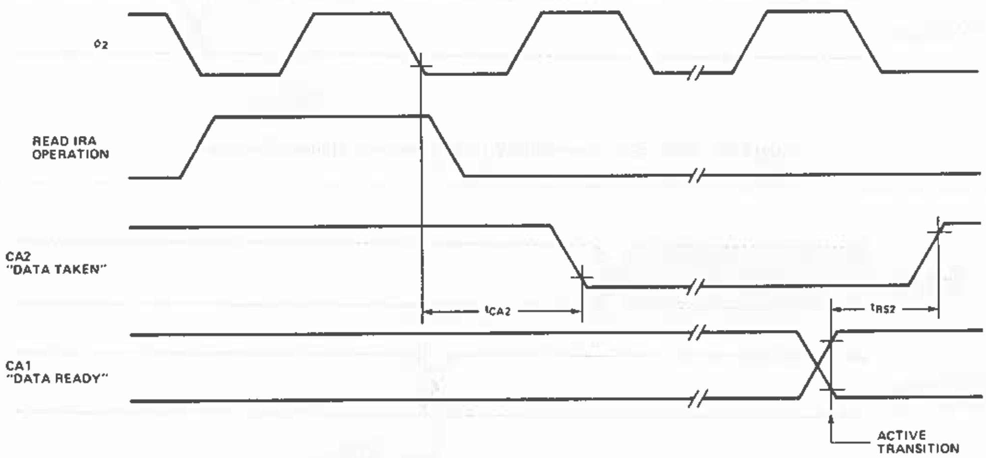


Figure 5b. CA2 Timing for Read Handshake, Handshake Mode

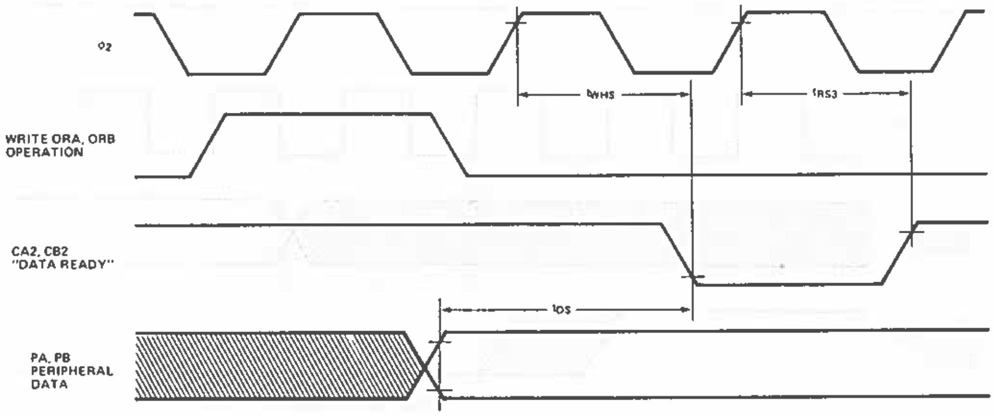


Figure 5c. CA2, CB2 Timing for Write Handshake, Pulse Mode

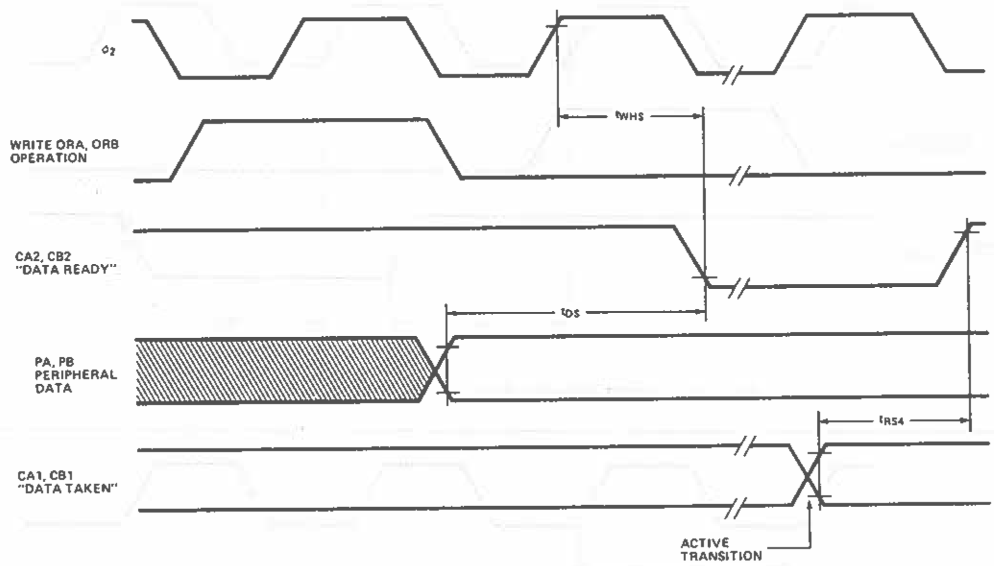


Figure 5d. CA2, CB2 Timing for Write Handshake, Handshake Mode

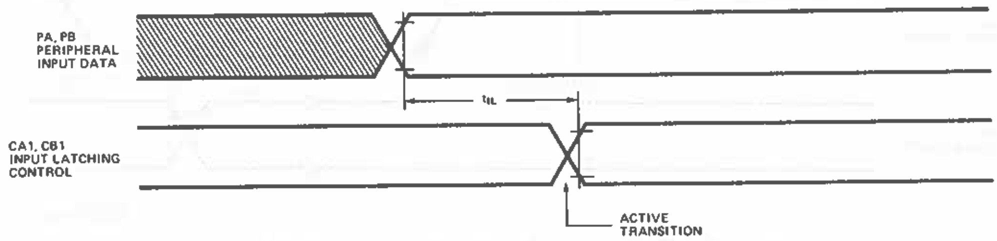


Figure 5e. Peripheral Data Input Latching Timing

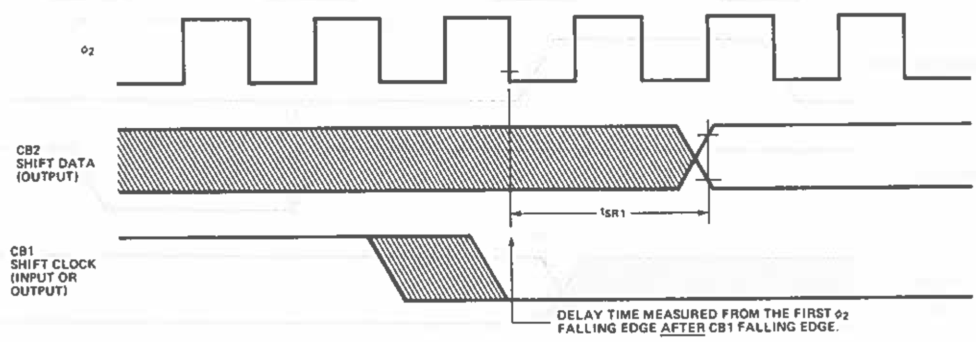


Figure 5f. Timing for Shift Out with Internal or External Shift Clocking

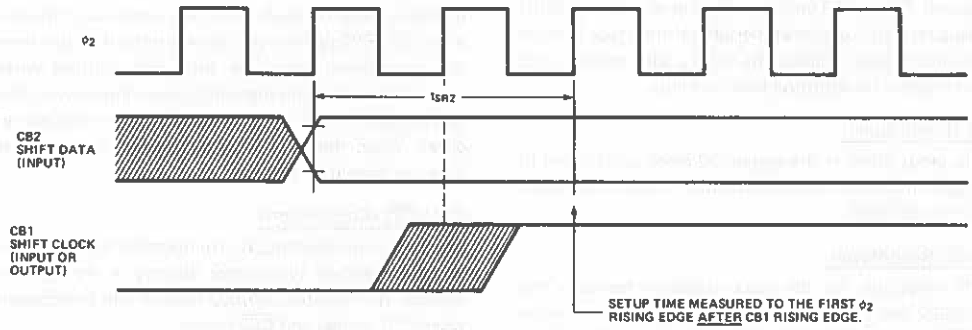


Figure 5g. Timing for Shift In with Internal or External Shift Clocking

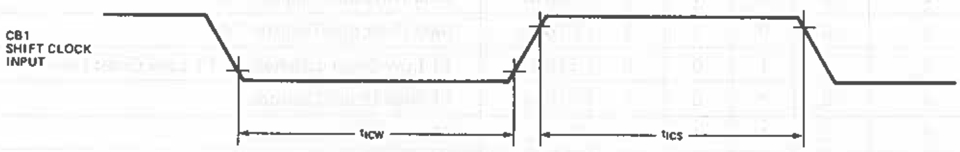


Figure 5h. External Shift Clock Timing

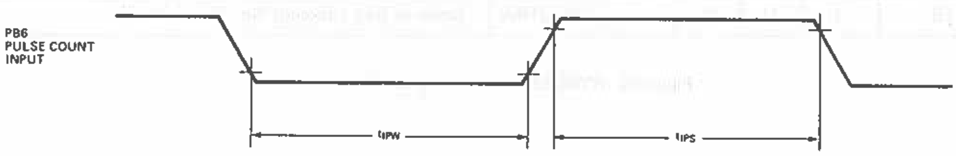


Figure 5i. Pulse Count Input Timing

PIN DESCRIPTIONS

RES (Reset)

The reset input clears all internal registers to logic 0 (except T1 and T2 latches and counters and the Shift Register). This places all peripheral interface lines in the input state, disables the timers, shift register, etc. and disables interrupting from the chip.

$\phi 2$ (Input Clock)

The input clock is the system $\phi 2$ clock and is used to trigger all data transfers between the system processor and the SY6522.

R/W (Read/Write)

The direction of the data transfers between the SY6522 and the system processor is controlled by the R/W line. If R/W is low, data will be transferred out of the processor into the selected SY6522 register (write operation). If R/W is high and the chip is selected, data will be transferred out of the SY6522 (read operation).

DB0–DB7 (Data Bus)

The eight bi-directional data bus lines are used to transfer data between the SY6522 and the system processor. During read cycles, the contents of the selected SY6522 register are placed on the data bus lines and transferred into the processor. During write cycles, these lines are high-impedance inputs and data is transferred from the processor into the selected register. When the SY6522 is unselected, the data bus lines are high-impedance.

CS1, CS2 (Chip Selects)

The two chip select inputs are normally connected to processor address lines either directly or through decoding. The selected SY6522 register will be accessed when CS1 is high and CS2 is low.

RS0–RS3 (Register Selects)

The four Register Select inputs permit the system processor to select one of the 16 internal registers of the SY6522, as shown in Figure 6.

Register Number	RS Coding				Register Desig.	Description	
	RS3	RS2	RS1	RS0		Write	Read
0	0	0	0	0	ORB/IRB	Output Register "B"	Input Register "B"
1	0	0	0	1	ORA/IRA	Output Register "A"	Input Register "A"
2	0	0	1	0	DDRB	Data Direction Register "B"	
3	0	0	1	1	DDRA	Data Direction Register "A"	
4	0	1	0	0	T1C-L	T1 Low-Order Latches	T1 Low-Order Counter
5	0	1	0	1	T1C-H	T1 High-Order Counter	
6	0	1	1	0	T1L-L	T1 Low-Order Latches	
7	0	1	1	1	T1L-H	T1 High-Order Latches	
8	1	0	0	0	T2C-L	T2 Low-Order Latches	T2 Low-Order Counter
9	1	0	0	1	T2C-H	T2 High-Order Counter	
10	1	0	1	0	SR	Shift Register	
11	1	0	1	1	ACR	Auxiliary Control Register	
12	1	1	0	0	PCR	Peripheral Control Register	
13	1	1	0	1	IFR	Interrupt Flag Register	
14	1	1	1	0	IER	Interrupt Enable Register	
15	1	1	1	1	ORA/IRA	Same as Reg 1 Except No "Handshake"	

Figure 6. SY6522 Internal Register Summary

MICRO-PROCESSORS

IRQ (Interrupt Request)

The Interrupt Request output goes low whenever an internal interrupt flag is set and the corresponding interrupt enable bit is a logic 1. This output is "open-drain" to allow the interrupt request signal to be "wire-or'ed" with other equivalent signals in the system.

PA0-PA7 (Peripheral A Port)

The Peripheral A port consists of 8 lines which can be individually programmed to act as inputs or outputs under control of a Data Direction Register. The polarity of output pins is controlled by an Output Register and input data may be latched into an internal register under control of the CA1 line. All of these modes of operation are controlled by the system processor through the internal control registers. These lines represent one standard TTL load in the input mode and will drive one standard TTL load in the output mode. Figure 7 illustrates the output circuit.

CA1, CA2 (Peripheral A Control Lines)

The two Peripheral A control lines act as interrupt inputs or as handshake outputs. Each line controls an internal interrupt flag with a corresponding interrupt enable bit. In addition, CA1 controls the latching of data on Peripheral A port input lines. CA1 is a high-impedance input only while CA2 represents one standard TTL load in the input mode. CA2 will drive one standard TTL load in the output mode.

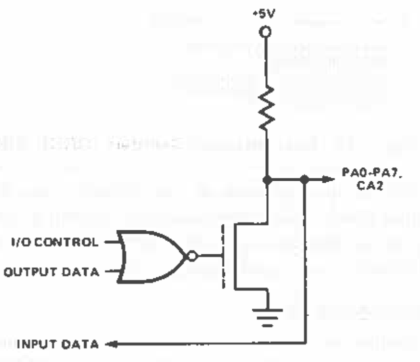


Figure 7. Peripheral A Port Output Circuit

PB0-PB7 (Peripheral B Port)

The Peripheral B port consists of eight bi-directional lines which are controlled by an output register and a data direction register in much the same manner as the PA port. In addition, the polarity of the PB7 output signal can be controlled by one of the interval timers while the second timer can be programmed to count pulses on the PB6 pin. Peripheral B lines represent one standard TTL load in the input mode and will drive one standard TTL load in the output mode. In addition, they are capable of sourcing 1.0mA at 1.5VDC in the output mode to allow the outputs to directly drive Darlington transistor circuits. Figure 8 is the circuit schematic.

CB1, CB2 (Peripheral B Control Lines)

The Peripheral B control lines act as interrupt inputs or as handshake outputs. As with CA1 and CA2, each line controls an interrupt flag with a corresponding interrupt enable bit. In addition, these lines act as a serial port under control of the Shift Register. These lines represent one standard TTL load in the input mode and will drive one standard TTL load in the output mode. Unlike PB0-PB7, CB1 and CB2 cannot drive Darlington transistor circuits.

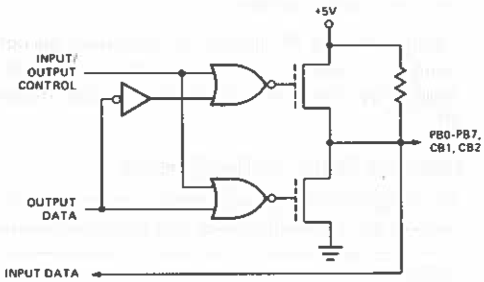


Figure 8. Peripheral B Port Output Circuit

FUNCTIONAL DESCRIPTION

Port A and Port B Operation

Each 8-bit peripheral port has a Data Direction Register (DDRA, DDRB) for specifying whether the peripheral pins are to act as inputs or outputs. A 0 in a bit of the Data Direction Register causes the corresponding peripheral pin to act as an input. A 1 causes the pin to act as an output.

Each peripheral pin is also controlled by a bit in the Output Register (ORA, ORB) and an Input Register (IRA, IRB). When the pin is programmed as an output, the voltage on the pin is controlled by the cor-

MICRO-PROCESSORS

responding bit of the Output Register. A 1 in the Output Register causes the output to go high, and a "0" causes the output to go low. Data may be written into Output Register bits corresponding to pins which are programmed as inputs. In this case, however, the output signal is unaffected.

Reading a peripheral port causes the contents of the Input Register (IRA, IRB) to be transferred onto the Data Bus. With input latching disabled, IRA will always reflect the levels on the PA pins. With input latching enabled, IRA will reflect the levels on the PA pins at the time the latching occurred (via CA1).

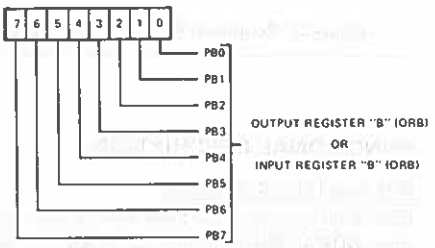
The IRB register operates similar to the IRA register. However, for pins programmed as outputs there is a difference. When reading IRA, the level on the pin determines whether a 0 or a 1 is sensed. When reading IRB, however, the bit stored in the output register, ORB, is the bit sensed. Thus, for outputs which have large loading effects and which pull an output "1" down or which pull an output "0" up, reading IRA may result in reading a "0" when a "1" was actually programmed, and reading a "1" when a "0" was programmed. Reading IRB, on the other hand, will read the "1" or "0" level actually programmed, no matter what the loading on the pin.

Figures 9, 10, and 11 illustrate the formats of the port registers. In addition, the input latching modes are selected by the Auxiliary Control Register (Figure 16.)

Handshake Control of Data Transfers

The SY6522 allows positive control of data transfers between the system processor and peripheral devices

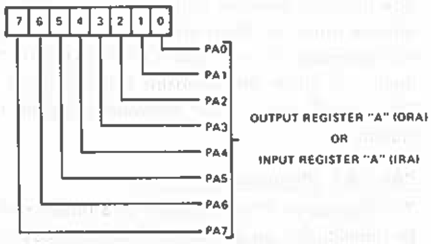
REG 0 — ORB/IRB



Pin Data Direction Selection	WRITE	READ
DDRB = "1" (OUTPUT)	MPU writes Output Level (ORB)	MPU reads output register bit in ORB. Pin level has no affect.
DDRB = "0" (INPUT) (Input latching disabled)	MPU writes into ORB, but no effect on pin level, until DDRB changed.	MPU reads input level on PB pin.
DDRB = "0" (INPUT) (Input latching enabled)		MPU reads IRB bit, which is the level of the PB pin at the time of the last CB1 active transition.

Figure 9. Output Register B (ORB), Input Register B (IRB)

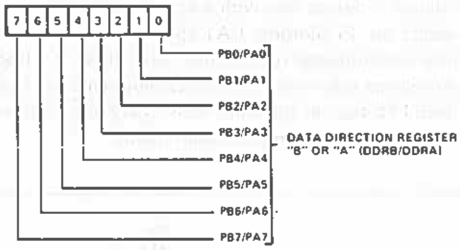
REG 1 — ORA/IRA



Pin Data Direction Selection	WRITE	READ
DDRA = "1" (OUTPUT) (Input latching disabled)	MPU writes Output Level (ORA)	MPU reads level on PA pin
DDRA = "1" (OUTPUT) (Input latching enabled)		MPU reads IRA bit which is the level of the PA pin at the time of the last CA1 active transition.
DDRA = "0" (INPUT) (Input latching disabled)	MPU writes into ORA, but no effect on pin level, until DDRA changed.	MPU reads level on PA pin.
DDRA = "0" (INPUT) (Input latching enabled)		MPU reads IRA bit which is the level of the PA pin at the time of the last CA1 active transition

Figure 10. Output Register A (ORA), Input Register A (IRA)

REG 2 (DDRB) AND REG 3 (DDRA)



"0" ASSOCIATED PB/PA PIN IS AN INPUT (HIGH IMPEDANCE)

"1" ASSOCIATED PB/PA PIN IS AN OUTPUT, WHOSE LEVEL IS DETERMINED BY ORB/ORA REGISTER BIT

Figure 11. Data Direction Registers (DDRB, DDRA)

through the operation of "handshake" lines. Port A lines (CA1, CA2) handshake data on both a read and a write operation while the Port B lines (CB1, CB2) handshake on a write operation only.

Read Handshake

Positive control of data transfers from peripheral devices into the system processor can be accomplished very effectively using Read Handshaking. In this case, the peripheral device must generate the equivalent of a "Data Ready" signal to the processor signifying that valid data is present on the peripheral port. This signal normally interrupts the processor, which then reads the data, causing generation of a "Data Taken" signal. The peripheral device responds by making new data available. This process continues until the data transfer is complete.

MICRO-PROCESSORS

SY6522/SY6522A

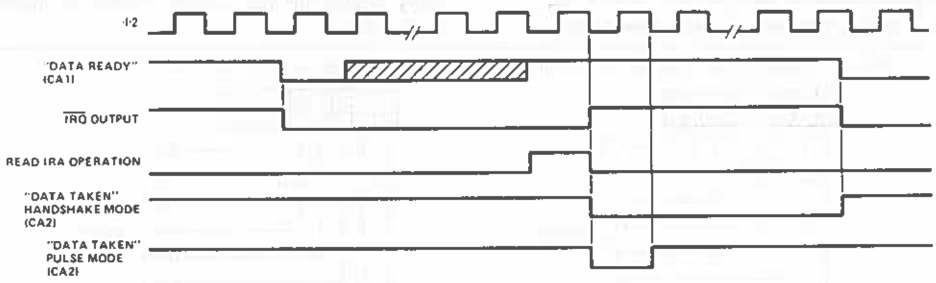


Figure 12. Read Handshake Timing (Port A, Only)

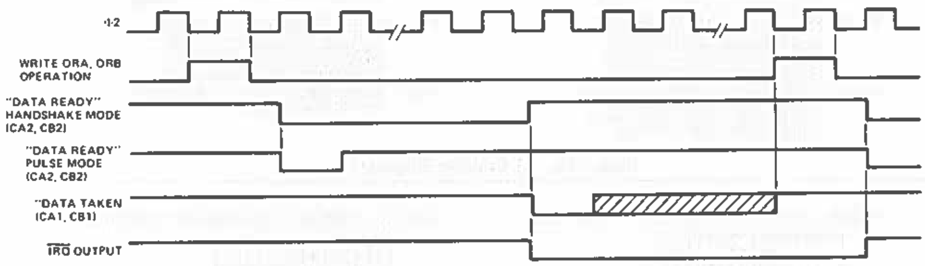


Figure 13. Write Handshake Timing

In the SY6522, automatic "Read" Handshaking is possible on the Peripheral A port only. The CA1 interrupt input pin accepts the "Data Ready" signal and CA2 generates the "Data Taken" signal. The "Data Ready" signal will set an internal flag which may interrupt the processor or which may be polled under program control. The "Data Taken" signal can either be a pulse or a level which is set low by the system processor and is cleared by the "Data Ready" signal. These options are shown in Figure 12 which illustrates the normal Read Handshaking sequence.

Write Handshake

The sequence of operations which allows handshaking data from the system processor to a peripheral device is very similar to that described for Read Handshaking. However, for Write Handshaking, the SY6522 generates the "Data Ready" signal and the peripheral device must respond with the "Data Taken" signal. This can be accomplished on both the PA port and the PB port on the SY6522. CA2 or CB2 act as a "Data Ready" output in either the handshake mode or pulse mode and CA1 or CB1 accept the "Data Taken" signal from the peripheral device, setting the interrupt flag and cleaning the "Data Ready" output. This sequence is shown in Figure 13.

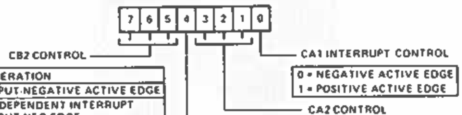
Selection of operating modes for CA1, CA2, CB1, and CB2 is accomplished by the Peripheral Control Register (Figure 14).

Timer Operation

Interval Timer T1 consists of two 8-bit latches and a 16-bit counter. The latches are used to store data which is to be loaded into the counter. After loading, the counter decrements at $\phi 2$ clock rate. Upon reaching zero, an interrupt flag will be set, and $\overline{IRQ}$ will go low if the interrupt is enabled. The timer will then disable any further interrupts, or will automatically transfer the contents of the latches into the counter and will continue to decrement. In addition, the timer may be programmed to invert the output signal on a peripheral pin each time it "times-out". Each of these modes is discussed separately below.

The T1 counter is depicted in Figure 15 and the latches in Figure 16.

REG 12 – PERIPHERAL CONTROL REGISTER



CB2 CONTROL

7	6	5	OPERATION
0	0	0	INPUT-NEGATIVE ACTIVE EDGE
0	0	1	INDEPENDENT INTERRUPT INPUT-NEG EDGE
0	1	0	INPUT-POSITIVE ACTIVE EDGE
0	1	1	INDEPENDENT INTERRUPT INPUT-POS EDGE
1	0	0	HANDSHAKE OUTPUT
1	0	1	PULSE OUTPUT
1	1	0	LOW OUTPUT
1	1	1	HIGH OUTPUT

CB1 INTERRUPT CONTROL

0 = NEGATIVE ACTIVE EDGE
1 = POSITIVE ACTIVE EDGE

CA1 INTERRUPT CONTROL

0 = NEGATIVE ACTIVE EDGE
1 = POSITIVE ACTIVE EDGE

CA2 CONTROL

3	2	1	OPERATION
0	0	0	INPUT-NEGATIVE ACTIVE EDGE
0	0	1	INDEPENDENT INTERRUPT INPUT-NEG EDGE
0	1	0	INPUT POSITIVE ACTIVE EDGE
0	1	1	INDEPENDENT INTERRUPT INPUT POS EDGE
1	0	0	HANDSHAKE OUTPUT
1	0	1	PULSE OUTPUT
1	1	0	LOW OUTPUT
1	1	1	HIGH OUTPUT

Figure 14. CA1, CA2, CB1, CB2 Control

MICRO-PROCESSORS

Two bits are provided in the Auxiliary Control Register (bits 6 and 7) to allow selection of the T1 operating modes. The four possible modes are depicted in Figure 17.

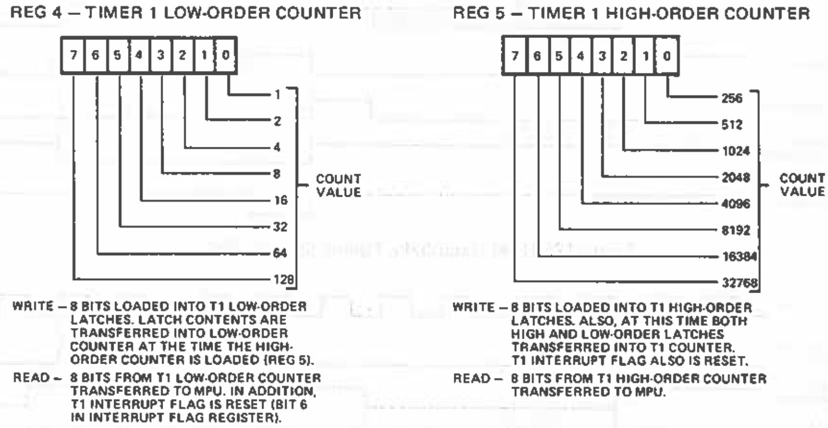


Figure 15. T1 Counter Registers

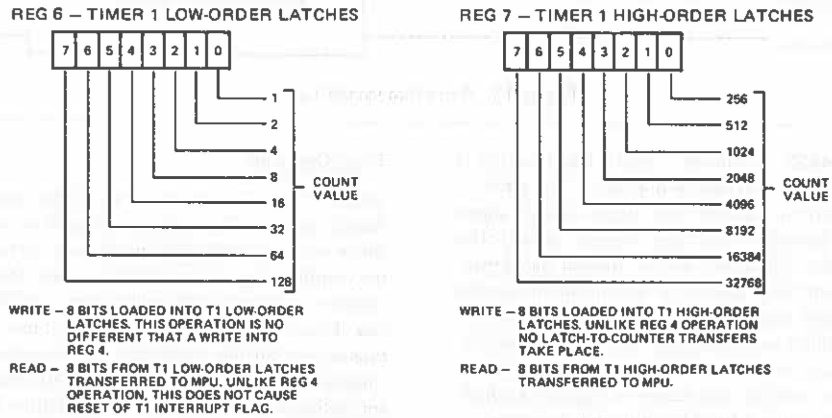


Figure 16. T1 Latch Registers

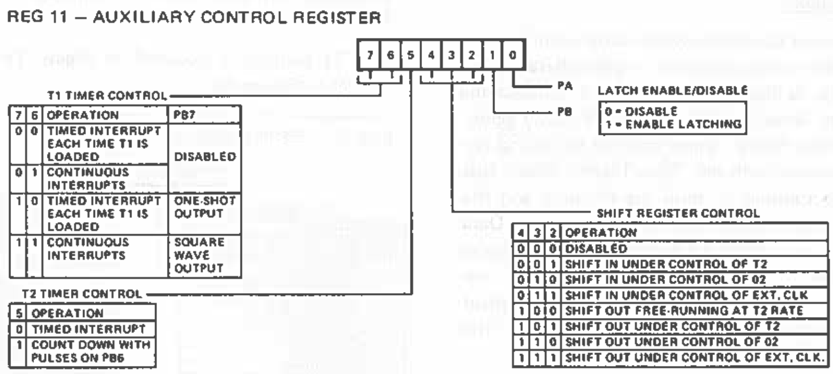


Figure 17. Auxiliary Control Register

Note: The processor does not write directly into the low order counter (T1C-L). Instead, this half of the counter is loaded automatically from the low order latch when the processor writes into the high order counter. In fact, it may not be necessary to write to the low order counter in some applications since the timing operation is triggered by writing to the high order counter.

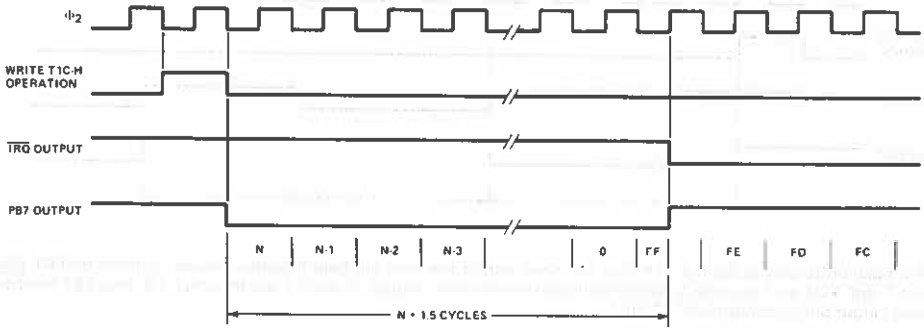


Figure 18. Timer 1 and Timer 2 One-Shot Mode Timing

Timer 1 One-Shot Mode

The interval timer one-shot mode allows generation of a single interrupt for each timer load operation. As with any interval timer, the delay between the "write T1C-H" operation and generation of the processor interrupt is a direct function of the data loaded into the timing counter. In addition to generating a single interrupt, Timer 1 can be programmed to produce a single negative pulse on the PB7 peripheral pin. With the output enabled (ACR7=1) a "write T1C-H" operation will cause PB7 to go low. PB7 will return high when Timer 1 times out. The result is a single programmable width pulse.

In the one-shot mode, writing into the high order latch has no effect on the operation of Timer 1. However, it will be necessary to assure that the low order latch contains the proper data before initiating the count-down with a "write T1C-H" operation. When the processor writes into the high order counter, the T1 interrupt flag will be cleared, the contents of the low order latch will be transferred into the low order counter, and the timer will begin to decrement at system clock rate. If the PB7 output is enabled, this signal will go low on the phase two following the write operation. When the counter reaches zero, the T1 interrupt flag will be set, the $\overline{IRQ}$ pin will go low (interrupt enabled), and the signal on PB7 will go high. At this time the counter will continue to decrement at system clock rate. This allows the system processor to read the contents of the counter to determine the time since interrupt. However, the T1 interrupt flag cannot be set again unless it has been cleared as described in this specification.

Timing for the SY6522 interval timer one-shot modes is shown in Figure 18.

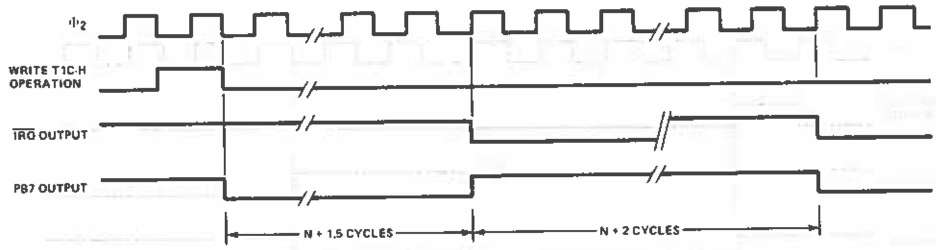
Timer 1 Free-Run Mode

The most important advantage associated with the latches in T1 is the ability to produce a continuous series of evenly spaced interrupts and the ability to produce a square wave on PB7 whose frequency is not affected by variations in the processor interrupt response time. This is accomplished in the "free-running" mode.

In the free-running mode, the interrupt flag is set and the signal on PB7 is inverted each time the counter reaches zero. However, instead of continuing to decrement from zero after a time-out, the timer automatically transfers the contents of the latch into the counter (16 bits) and continues to decrement from there. The interrupt flag can be cleared by writing T1C-H, by reading T1C-L, or by writing directly into the flag as described later. However, it is not necessary to rewrite the timer to enable setting the interrupt flag on the next time-out.

All interval timers in the SY6522 are "re-triggerable". Rewriting the counter will always re-initialize the time-out period. In fact, the time-out can be prevented completely if the processor continues to rewrite the timer before it reaches zero. Timer 1 will operate in this manner if the processor writes into the high order counter (T1C-H). However, by loading the latches only, the processor can access the timer during each down-counting operation without affecting the time-out in process. Instead, the data loaded into the latches will determine the length of the next time-out period. This capability is particularly valuable in the free-running mode with the output enabled. In this mode, the signal on PB7 is inverted and the interrupt flag is set with each time-out. By responding to the interrupts with new data for the latches, the processor can determine the period of the next half cycle during each half cycle of the output signal on PB7. In this manner, very complex waveforms can be generated. Timing for the free-running mode is shown in Figure 19.

MICRO-PROCESSORS



Note: A precaution to take in the use of PB7 as the timer output concerns the Data Direction Register contents for PB7. Both DDRB bit 7 and ACR bit 7 must be 1 for PB7 to function as the timer output. If one is 1 and the other is 0, then PB7 functions as a normal output pin, controlled by ORB bit 7.

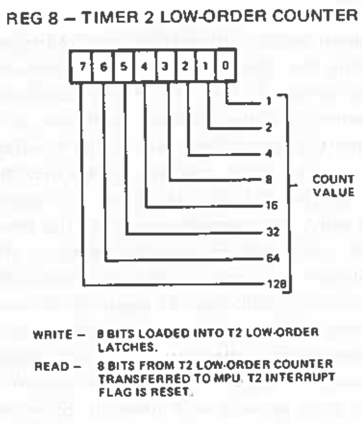
Figure 19. Timer 1 Free-Run Mode Timing

Timer 2 Operation

Timer 2 operates as an interval timer (in the "one-slot" mode only), or as a counter for counting negative pulses on the PB6 peripheral pin. A single control bit is provided in the Auxiliary Control Register to select between these two modes. This timer is comprised of a "write-only" low-order latch (T2L-L), a "read-only" low-order counter and a read/write high order counter. The counter registers act as a 16-bit counter which decrements at Φ2 rate. Figure 20 illustrates the T2 Counter Registers.

Timer 2 One-Shot Mode

As an interval timer, T2 operates in the "one-shot" mode similar to Timer 1. In this mode, T2 provides a single interrupt for each "write T2C-H" operation. After timing out, the counter will continue to decrement. However, setting of the interrupt flag will be disabled after initial time-out so that it will not be set by the counter continuing to decrement through zero. The processor must rewrite T2C-H to enable setting of the interrupt flag. The interrupt flag is cleared by reading T2C-L or by writing T2C-H. Timing for this operation is shown in Figure 18.



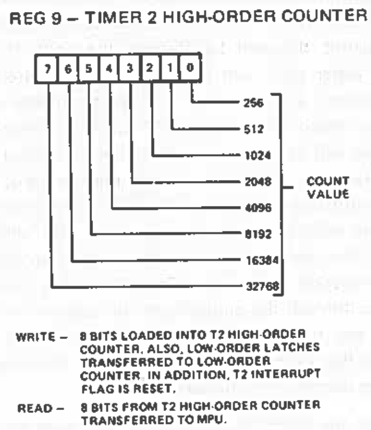


Figure 20. T2 Counter Registers

MICRO-PROCESSORS

Timer 2 Pulse Counting Mode

In the pulse counting mode, T2 serves primarily to count a predetermined number of negative-going pulses on PB6. This is accomplished by first loading a number into T2. Writing into T2C-H clears the interrupt flag and allows the counter to decrement each time a pulse is applied to PB6. The interrupt flag will be set when T2 reaches zero. At this time the counter will continue to decrement with each pulse on PB6. However, it is necessary to rewrite T2C-H to allow the interrupt flag to set on subsequent down-counting operations. Timing for this mode is shown in Figure 21. The pulse must be low on the leading edge of $\Phi 2$.

Shift Register Operation

The Shift Register (SR) performs serial data transfers into and out of the CB2 pin under control of an internal modulo-8 counter. Shift pulses can be applied to the CB1 pin from an external source or, with the proper mode selection, shift pulses generated internally will appear on the CB1 pin for controlling external devices.

The control bits which select the various shift register operating modes are located in the Auxiliary Control Register. Figure 22 illustrates the configuration of the SR data bits and the SR control bits of the ACR. Figures 23 and 24 illustrate the operation of the various shift register modes.

Interrupt Operation

Controlling interrupts within the SY6522 involves three principal operations. These are flagging the interrupts, enabling interrupts and signaling to the processor that an active interrupt exists within the chip. Interrupt flags are set by interrupting conditions which exist within the chip or on inputs to the chip. These flags normally remain set until the interrupt has been serviced. To determine the source of an interrupt, the microprocessor must examine these flags in order from highest to lowest priority. This is accomplished by reading the flag register into the processor accumulator, shifting this register either right or left and then using conditional branch instructions to detect an active interrupt.

Associated with each interrupt flag is an interrupt enable bit. This can be set or cleared by the processor to enable interrupting the processor from the corresponding interrupt flag. If an interrupt flag is set to a logic 1 by an interrupting condition, and the corresponding interrupt enable bit is set to a 1, the Interrupt Request Output ($\overline{IRQ}$) will go low. $\overline{IRQ}$ is an "open-collector" output which can be "wire-or'ed" with other devices in the system to interrupt the processor.

In the SY6522, all the interrupt flags are contained in one register. In addition, bit 7 of this register will be read as a logic 1 when an interrupt exists within the chip. This allows very convenient polling of several devices within a system to locate the source of an interrupt.

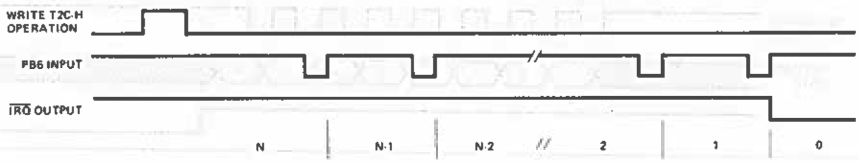
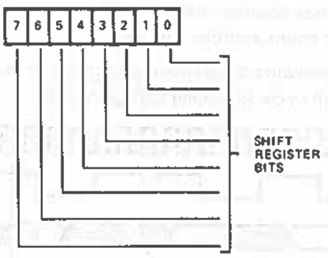


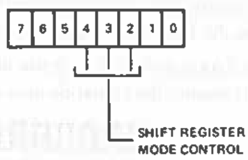
Figure 21. Timer 2 Pulse Counting Mode

REG 10 – SHIFT REGISTER



NOTES:
1. WHEN SHIFTING OUT, BIT 7 IS THE FIRST BIT OUT AND SIMULTANEOUSLY IS ROTATED BACK INTO BIT 0.
2. WHEN SHIFTING IN, BITS INITIALLY ENTER BIT 0 AND ARE SHIFTED TOWARDS BIT 7.

REG 11 – AUXILIARY CONTROL REGISTER



4	3	2	OPERATION
0	0	0	DISABLED
0	0	1	SHIFT IN UNDER CONTROL OF T2
0	1	0	SHIFT IN UNDER CONTROL OF Φ_2
0	1	1	SHIFT IN UNDER CONTROL OF EXT CLK
1	0	0	SHIFT OUT FREE-RUNNING AT T2 RATE
1	0	1	SHIFT OUT UNDER CONTROL OF T2
1	1	0	SHIFT OUT UNDER CONTROL OF Φ_2
1	1	1	SHIFT OUT UNDER CONTROL OF EXT CLK

Figure 22. SR and ACR Control Bits

MICRO-PROCESSORS

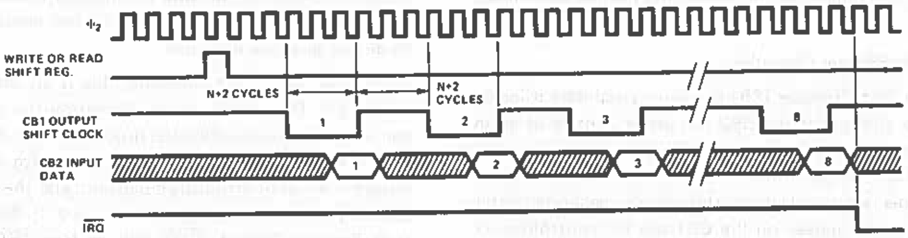
SR Disabled (000)

The 000 mode is used to disable the Shift Register. In this mode the microprocessor can write or read the SR, but the shifting operation is disabled and operation of CB1 and CB2 is controlled by the appropriate bits in the Peripheral Control Register (PCR). In this mode the SR Interrupt Flag is disabled (held to a logic 0).

Shift in Under Control of T2 (001)

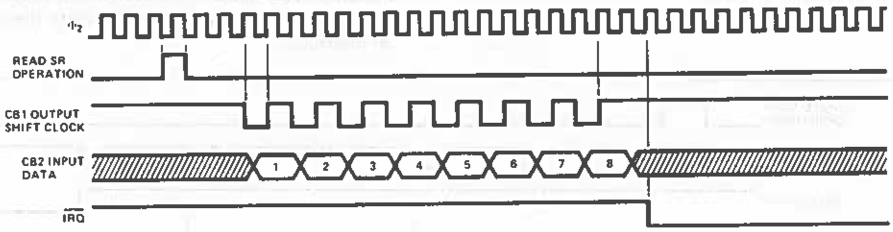
In the 001 mode the shifting rate is controlled by the low order 8 bits of T2. Shift pulses are generated on the CB1 pin to control shifting in external devices. The time between transitions of this output clock is a function of the system clock period and the contents of the low order T2 latch (N).

The shifting operation is triggered by writing or reading the shift register. Data is shifted first into the low order bit of SR and is then shifted into the next higher order bit of the shift register on the negative-going edge of each clock pulse. The input data should change before the positive-going edge of the CB1 clock pulse. This data is shifted into the shift register during the ϕ_2 clock cycle following the positive-going edge of the CB1 clock pulse. After 8 CB1 clock pulses, the shift register interrupt flag will be set and $\overline{IRQ}$ will go low.



Shift in Under Control of ϕ_2 (010)

In mode 010 the shift rate is a direct function of the system clock frequency. CB1 becomes an output which generates shift pulses for controlling external devices. Timer 2 operates as an independent interval timer and has no effect on SR. The shifting operation is triggered by reading or writing the Shift Register. Data is shifted first into bit 0 and is then shifted into the next higher order bit of the shift register on the trailing edge of each ϕ_2 clock pulse. After 8 clock pulses, the shift register interrupt flag will be set, and the output clock pulses on CB1 will stop.



Shift in Under Control of External CB1 Clock (011)

In mode 011 CB1 becomes an input. This allows an external device to load the shift register at its own pace. The shift register counter will interrupt the processor each time 8 bits have been shifted in. However, the shift register counter does not stop the shifting operation; it acts simply as a pulse counter. Reading or writing the Shift Register resets the Interrupt flag and initializes the SR counter to count another 8 pulses.

Note that the data is shifted during the first system clock cycle following the positive-going edge of the CB1 shift pulse. For this reason, data must be held stable during the first full cycle following CB1 going high.

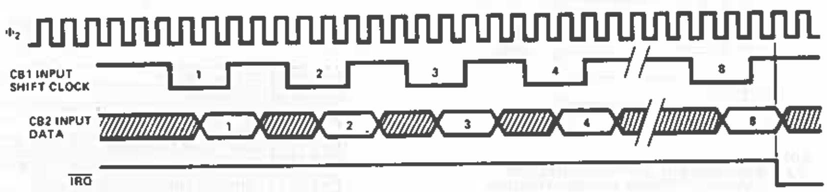
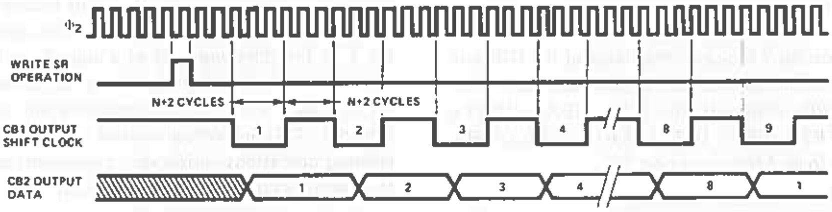


Figure 23. Shift Register Input Modes

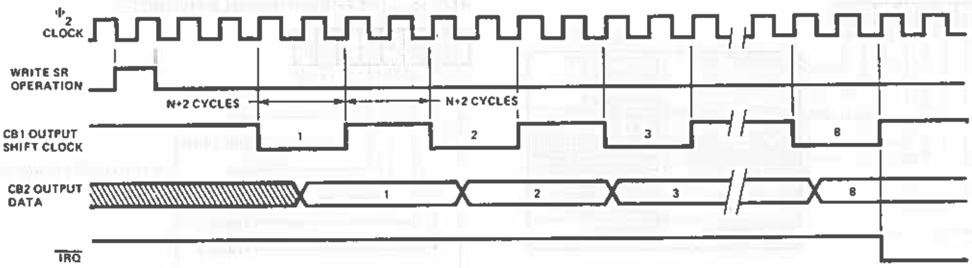
Shift Out Free-Running at T2 Rate (100)

Mode 100 is very similar to mode 101 in which the shifting rate is set by T2. However, in mode 100 the SR Counter does not stop the shifting operation. Since the Shift Register bit 7 (SR7) is recirculated back into bit 0, the 8 bits loaded into the shift register will be clocked onto CB2 repetitively. In this mode the shift register counter is disabled.



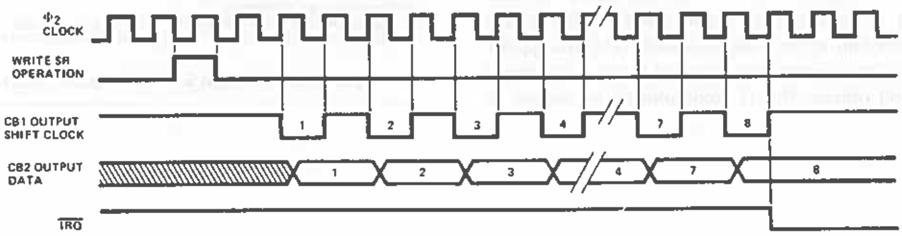
Shift Out Under Control of T2 (101)

In mode 101 the shift rate is controlled by T2 (as in the previous mode). However, with each read or write of the shift register the SR Counter is reset and 8 bits are shifted onto CB2. At the same time, 8 shift pulses are generated on CB1 to control shifting in external devices. After the 8 shift pulses, the shifting is disabled, the SR Interrupt Flag is set and CB2 remains at the last data level.



Shift Out Under Control of ϕ_2 (110)

In mode 110, the shift rate is controlled by the ϕ_2 system clock.



Shift Out Under Control of External CB1 Clock (111)

In mode 111 shifting is controlled by pulses applied to the CB1 pin by an external device. The SR counter sets the SR Interrupt flag each time it counts 8 pulses but it does not disable the shifting function. Each time the microprocessor writes or reads the shift register, the SR Interrupt flag is reset and the SR counter is initialized to begin counting the next 8 shift pulses on pin CB1. After 8 shift pulses, the interrupt flag is set. The microprocessor can then load the shift register with the next byte of data.

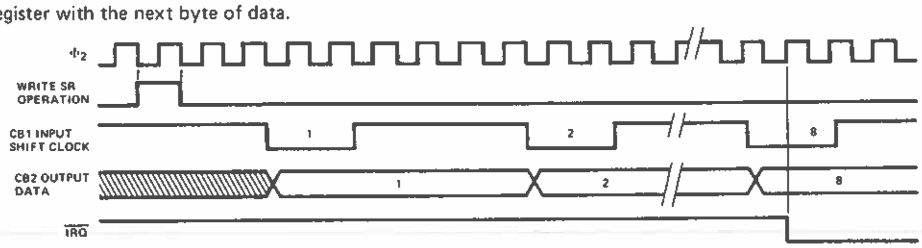


Figure 24. Shift Register Output Modes

MICRO-PROCESSORS

The Interrupt Flag Register (IFR) and Interrupt Enable Register (IER) are depicted in Figures 25 and 26, respectively.

The IFR may be read directly by the processor. In addition, individual flag bits may be cleared by writing a "1" into the appropriate bit of the IFR. When the proper chip select and register signals are applied to the chip, the contents of this register are placed on the data bus. Bit 7 indicates the status of the IRQ output. This bit corresponds to the logic function: IRQ = IFR6 x IER6 + IFR5 x IER5 + IFR4 x IER4 + IFR3 x IER3 + IFR2 x IER2 + IFR1 x IER1 + IFR0 x IER0. Note: X = logic AND, + = Logic OR.

The IFR bit 7 is not a flag. Therefore, this bit is not directly cleared by writing a logic 1 into it. It can only be cleared by clearing all the flags in the register or by disabling all the active interrupts as discussed in the next section.

REG 13 — INTERRUPT FLAG REGISTER

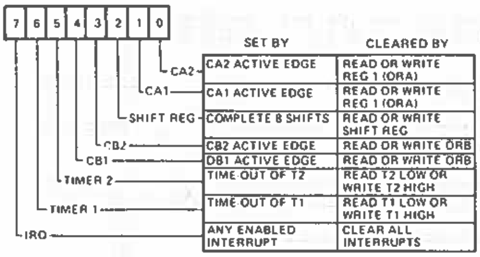


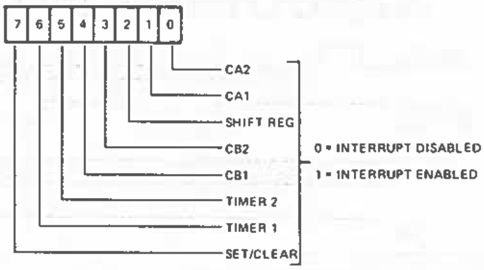
Figure 25. Interrupt Flag Register (IFR)

For each interrupt flag in IFR, there is a corresponding bit in the Interrupt Enable Register. The system processor can set or clear selected bits in this register to facilitate controlling individual interrupts without affecting others. This is accomplished by writing to address 1110 (IER address). If bit 7 of the data placed on the system data bus during this write operation is a 0, each 1 in bits 6 through 0 clears the corresponding bit in the Interrupt Enable Register. For each zero in bits 6 through 0, the corresponding bit is unaffected.

Setting selected bits in the Interrupt Enable Register is accomplished by writing to the same address with bit 7 in the data word set to a logic 1. In this case, each 1 in bits 6 through 0 will set the corresponding bit. For each zero, the corresponding bit will be unaffected. This individual control of the setting and clearing operations allows very convenient control of the interrupts during system operation.

In addition to setting and clearing IER bits, the processor can read the contents of this register by placing the proper address on the register select and chip select inputs with the R/W line high. Bit 7 will be read as a logic 0.

REG 14 — INTERRUPT ENABLE REGISTER

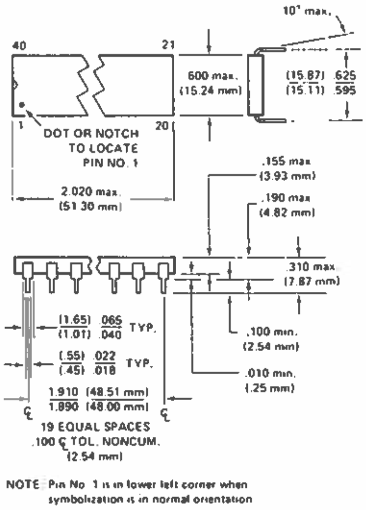


NOTES:
1 IF BIT 7 IS A "0", THEN EACH "1" IN BITS 0 - 6 DISABLES THE CORRESPONDING INTERRUPT.
2. IF BIT 7 IS A "1", THEN EACH "1" IN BITS 0 - 6 ENABLES THE CORRESPONDING INTERRUPT.
3. IF A READ OF THIS REGISTER IS DONE, BIT 7 WILL BE "0" AND ALL OTHER BITS WILL REFLECT THEIR ENABLE/DISABLE STATE.

Figure 26. Interrupt Enable Register (IER)

SY6522/SY6522A

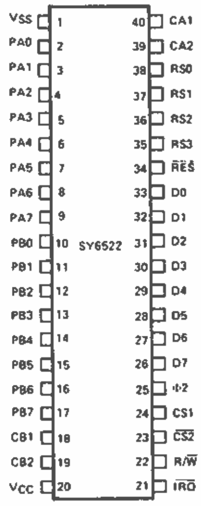
PACKAGE OUTLINE



ORDERING INFORMATION

Order Number	Package Type	Frequency Option
SYP 6522	Plastic	1 MHz
SYP 6522A	Plastic	2 MHz
SYC 6522	Ceramic	1 MHz
SYC 6522A	Ceramic	2 MHz

PIN CONFIGURATION



MICRO-PROCESSORS

Fiche technique :

ENCODEUR DE CLAVIER

intel®

PRELIMINARY
Notice: This is not a final specification. Some parametric limits are subject to change.

8279/8279-5
PROGRAMMABLE KEYBOARD/DISPLAY INTERFACE

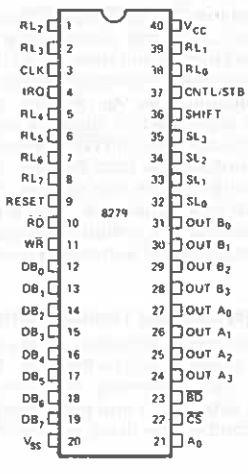
- MCS-85™ Compatible 8279-5
- Simultaneous Keyboard Display Operations
- Scanned Keyboard Mode
- Scanned Sensor Mode
- Strobed Input Entry Mode
- 8-Character Keyboard FIFO
- 2-Key Lockout or N-Key Rollover with Contact Debounce
- Dual 8- or 16-Numerical Display
- Single 16-Character Display
- Right or Left Entry 16-Byte Display RAM
- Mode Programmable from CPU
- Programmable Scan Timing
- Interrupt Output on Key Entry

The Intel® 8279 is a general purpose programmable keyboard and display I/O interface device designed for use with Intel® microprocessors. The keyboard portion can provide a scanned interface to a 64-contact key matrix. The keyboard portion will also interface to an array of sensors or a strobed interface keyboard, such as the hall effect and ferrite variety. Key depressions can be 2-key lockout or N-key rollover. Keyboard entries are debounced and strobed in an 8-character FIFO. If more than 8 characters are entered, overrun status is set. Key entries set the interrupt output line to the CPU.

The display portion provides a scanned display interface for LED, incandescent, and other popular display technologies. Both numeric and alphanumeric segment displays may be used as well as simple indicators. The 8279 has 16X8 display RAM which can be organized into dual 16X4. The RAM can be loaded or interrogated by the CPU. Both right entry, calculator and left entry typewriter display formats are possible. Both read and write of the display RAM can be done with auto-increment of the display RAM address.

MPU PERIPHERALS

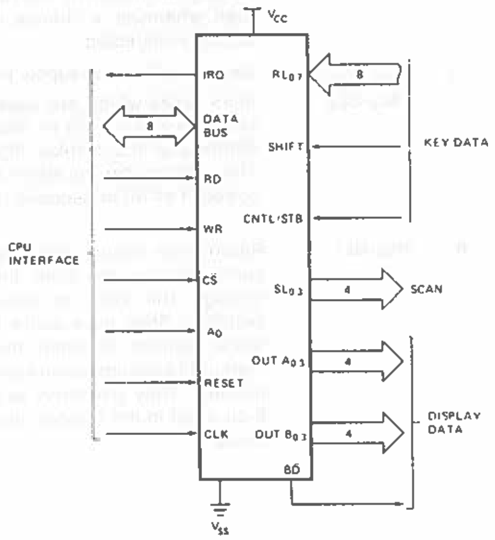
PIN CONFIGURATION



PIN NAMES

DB_{0-7}	I/O	DATA BUS (BI-DIRECTIONAL)
CLK	I	CLOCK INPUT
RESET	I	RESET INPUT
$\overline{CS}$	I	CHIP SELECT
$\overline{RD}$	I	READ INPUT
$\overline{WR}$	I	WRITE INPUT
A_0	I	BUFFER ADDRESS
IRQ	O	INTERRUPT REQUEST OUTPUT
SL_{0-3}	O	SCAN LINES
RL_{0-7}	I	RETURN LINES
SHIFT	I	SHIFT INPUT
CNTL/STB	I	CONTROL/STROBE INPUT
OUT A_{0-3}	O	DISPLAY (A) OUTPUTS
OUT B_{0-3}	O	DISPLAY (B) OUTPUTS
$\overline{BD}$	O	BLANK DISPLAY OUTPUT

LOGIC SYMBOL



HARDWARE DESCRIPTION

The 8279 is packaged in a 40 pin DIP. The following is a functional description of each pin.

No. Of Pins	Designation	Function
8	DB_0-DB_7	Bi-directional data bus. All data and commands between the CPU and the 8279 are transmitted on these lines.
1	CLK	Clock from system used to generate internal timing.
1	RESET	A high signal on this pin resets the 8279. After being reset the 8279 is placed in the following mode: 1) 16 8-bit character display —left entry. 2) Encoded scan keyboard—2 key lockout. Along with this the program clock prescaler is set to 31.
1	$\overline{CS}$	Chip Select. A low on this pin enables the interface functions to receive or transmit.
1	A_0	Buffer Address. A high on this line indicates the signals in or out are interpreted as a command or status. A low indicates that they are data.
2	$\overline{RD}$, $\overline{WR}$	Input/Output read and write. These signals enable the data buffers to either send data to the external bus or receive it from the external bus.
1	IRQ	Interrupt Request. In a keyboard mode, the interrupt line is high when there is data in the FIFO/Sensor RAM. The interrupt line goes low with each FIFO/Sensor RAM read and returns high if there is still information in the RAM. In a sensor mode, the interrupt line goes high whenever a change in a sensor is detected.
2	V_{SS}, V_{CC}	Ground and power supply pins.
4	SL_0-SL_3	Scan Lines which are used to scan the key switch or sensor matrix and the display digits. These lines can be either encoded (1 of 16) or decoded (1 of 4).
8	RL_0-RL_7	Return line inputs which are connected to the scan lines through the keys or sensor switches. They have active internal pullups to keep them high until a switch closure pulls one low. They also serve as an 8-bit input in the Strobed Input mode.
1	SHIFT	The shift input status is stored along with the key position on key closure in the Scanned Keyboard modes. It has an active internal pullup to keep it high until a switch closure pulls it low.
1	CNTL/STB	For keyboard modes this line is used as a control input and stored like status on a key closure. The line is also the strobe line that enters the data into the FIFO in the Strobed Input mode. (Rising Edge). It has an active internal pullup to keep it high until a switch closure pulls it low.
4 4	OUT A_0-OUT A_3 OUT B_0-OUT B_3	These two ports are the outputs for the 16 x 4 display refresh registers. The data from these outputs is synchronized to the scan lines (SL_0-SL_3) for multiplexed digit displays. The two 4 bit ports may be blanked independently. These two ports may also be considered as one 8 bit port.
1	$\overline{BD}$	Blank Display. This output is used to blank the display during digit switching or by a display blanking command.

PRINCIPLES OF OPERATION

The following is a description of the major elements of the 8279 Programmable Keyboard/Display interface device. Refer to the block diagram in Figure 1.

I/O Control and Data Buffers

The I/O control section uses the $\overline{CS}$, A_0, $\overline{RD}$ and $\overline{WR}$ lines to control data flow to and from the various internal registers and buffers. All data flow to and from the 8279 is enabled by $\overline{CS}$. The character of the information, given or desired by the CPU, is identified by A_0. A logic one means the information is a command or status. A logic zero means the information is data. $\overline{RD}$ and $\overline{WR}$ determine the direction of data flow through the Data Buffers. The Data Buffers are bi-directional buffers that connect the internal bus to the external bus. When the chip is not selected ($\overline{CS}$ = 1), the devices are in a high impedance state. The drivers input during $\overline{WR} \bullet \overline{CS}$ and output during $\overline{RD} \bullet \overline{CS}$.

Control and Timing Registers and Timing Control

These registers store the keyboard and display modes and other operating conditions programmed by the CPU. The modes are programmed by presenting the proper command on the data lines with A_0 = 1 and then sending a $\overline{WR}$. The command is latched on the rising edge of $\overline{WR}$.

MPU PERIPHERALS

FUNCTIONAL DESCRIPTION

Since data input and display are an integral part of many microprocessor designs, the system designer needs an interface that can control these functions without placing a large load on the CPU. The 8279 provides this function for 8-bit microprocessors.

The 8279 has two sections: keyboard and display. The keyboard section can interface to regular typewriter style keyboards or random toggle or thumb switches. The display section drives alphanumeric displays or a bank of indicator lights. Thus the CPU is relieved from scanning the keyboard or refreshing the display.

The 8279 is designed to directly connect to the microprocessor bus. The CPU can program all operating modes for the 8279. These modes include:

Input Modes

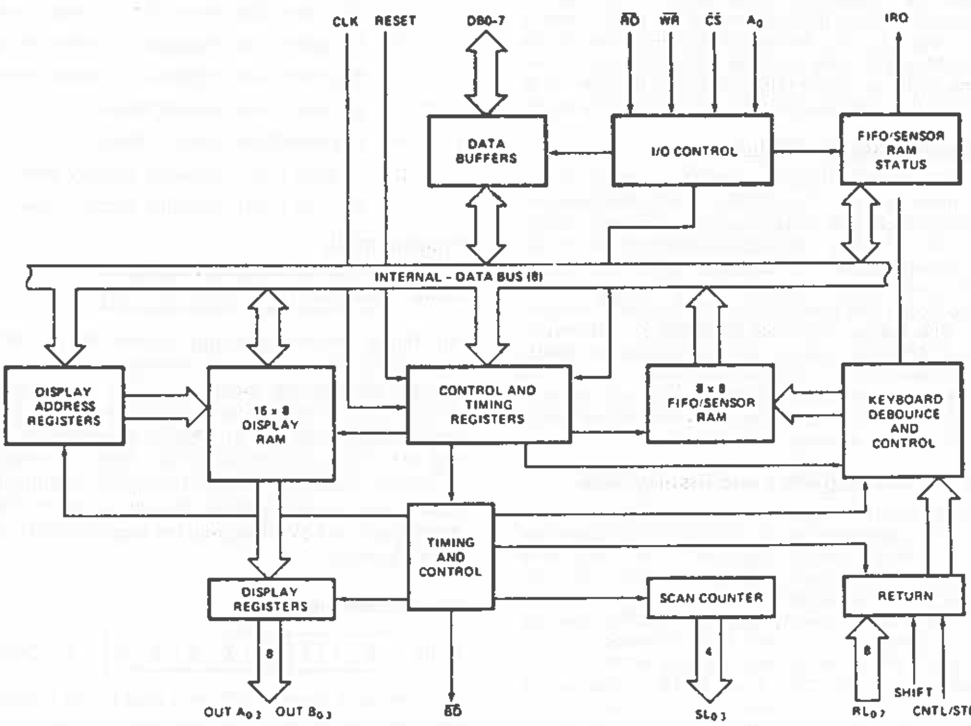
- Scanned Keyboard — with encoded (8 x 8 key keyboard) or decoded (4 x 8 key keyboard) scan lines. A key depression generates a 6-bit encoding of key position. Position and shift and control status are stored in the FIFO. Keys are automatically debounced with 2-key lockout or N-key rollover.
- Scanned Sensor Matrix — with encoded (8 x 8 matrix switches) or decoded (4 x 8 matrix switches) scan lines. Key status (open or closed) stored in RAM addressable by CPU.
- Strobed Input -- Data on return lines during control line strobe is transferred to FIFO.

Output Modes

- 8 or 16 character multiplexed displays that can be organized as dual 4-bit or single 8-bit.
- Right entry or left entry display formats.

Other features of the 8279 include:

- Mode programming from the CPU.
- Clock Prescaler
- Interrupt output to signal CPU when there is keyboard or sensor data available.
- An 8 byte FIFO to store keyboard information.
- 16 byte internal Display RAM for display refresh. This RAM can also be read by the CPU.



MPU PERIPHERALS

The command is then decoded and the appropriate function is set. The timing control contains the basic timing counter chain. The first counter is a ÷ N prescaler that can be programmed to yield an internal frequency of 100 kHz which gives a 5.1 ms keyboard scan time and a 10.3 ms debounce time. The other counters divide down the basic internal frequency to provide the proper key scan, row scan, keyboard matrix scan, and display scan times.

Scan Counter

The scan counter has two modes. In the encoded mode, the counter provides a binary count that must be externally decoded to provide the scan lines for the keyboard and display. In the decoded mode, the scan counter decodes the least significant 2 bits and provides a decoded 1 of 4 scan. Note than when the keyboard is in decoded scan, so is the display. This means that only the first 4 characters in the Display RAM are displayed.

In the encoded mode, the scan lines are active high outputs. In the decoded mode, the scan lines are active low outputs.

Return Buffers and Keyboard Debounce and Control

The 8 return lines are buffered and latched by the Return Buffers. In the keyboard mode, these lines are scanned, looking for key closures in that row. If the debounce circuit detects a closed switch, it waits about 10 msec to check if the switch remains closed. If it does, the address of the switch in the matrix plus the status of SHIFT and CONTROL are transferred to the FIFO. In the scanned Sensor Matrix modes, the contents of the return lines is directly transferred to the corresponding row of the Sensor RAM (FIFO) each key scan time. In Strobed Input mode, the contents of the return lines are transferred to the FIFO on the rising edge of the CNTL/STB line pulse.

FIFO/Sensor RAM and Status

This block is a dual function 8 x 8 RAM. In Keyboard or Strobed Input modes, it is a FIFO. Each new entry is written into successive RAM positions and each is then read in order of entry. FIFO status keeps track of the number of characters in the FIFO and whether it is full or empty. Too many reads or writes will be recognized as an error. The status can be read by an $\overline{RD}$ with $\overline{CS}$ low and A_0 high. The status logic also provides an IRQ signal when the FIFO is not empty. In Scanned Sensor Matrix mode, the memory is a Sensor RAM. Each row of the Sensor RAM is loaded with the status of the corresponding row of sensor in the sensor matrix. In this mode, IRQ is high if a change in a sensor is detected.

Display Address Registers and Display RAM

The Display Address Registers hold the address of the word currently being written or read by the CPU and the two 4-bit nibbles being displayed. The read/write addresses are programmed by CPU command. They also can be set to auto increment after each read or write. The Display RAM can be directly read by the CPU after the correct mode and address is set. The addresses for the A and B nibbles are automatically updated by the 8279 to match data entry by the CPU. The A and B nibbles can be entered independently or as one word, according to the mode that is set by the CPU. Data entry to the display can be set to either left or right entry. See Interface Considerations for details.

SOFTWARE OPERATION

8279 commands

The following commands program the 8279 operating modes. The commands are sent on the Data Bus with $\overline{CS}$ low and A_0 high and are loaded to the 8279 on the rising edge of $\overline{WR}$.

Keyboard/Display Mode Set

	MSB							LSB
Code:	0	0	0	D	D	K	K	K

Where DD is the Display Mode and KKK is the Keyboard Mode.

DD

0 0 8 8-bit character display — Left entry
0 1 16 8-bit character display — Left entry*
1 0 8 8-bit character display — Right entry
1 1 16 8-bit character display — Right entry

For description of right and left entry, see Interface Considerations. Note that when decoded scan is set in keyboard mode, the display is reduced to 4 characters independent of display mode set.

KKK

0 0 0 Encoded Scan Keyboard — 2 Key Lockout*
0 0 1 Decoded Scan Keyboard — 2-Key Lockout
0 1 0 Encoded Scan Keyboard — N-Key Rollover
0 1 1 Decoded Scan Keyboard — N-Key Rollover
1 0 0 Encoded Scan Sensor Matrix
1 0 1 Decoded Scan Sensor Matrix
1 1 0 Strobed Input, Encoded Display Scan
1 1 1 Strobed Input, Decoded Display Scan

Program Clock

Code:	0	0	1	P	P	P	P	P

All timing and multiplexing signals for the 8279 are generated by an internal prescaler. This prescaler divides the external clock (pin 3) by a programmable integer. Bits PPPPP determine the value of this integer which ranges from 2 to 31. Choosing a divisor that yields 100 kHz will give the specified scan and debounce times. For instance, if Pin 3 of the 8279 is being clocked by a 2 MHz signal, PPPPP should be set to 10100 to divide the clock by 20 to yield the proper 100 kHz operating frequency.

Read FIFO/Sensor RAM

Code:	0	1	0	AI	X	A	A	A	X = Don't Care

The CPU sets up the 8279 for a read of the FIFO/Sensor RAM by first writing this command. In the Scan Key-

*Default after reset

MPU PERIPHERALS

board Mode, the Auto-Increment flag (AI) and the RAM address bits (AAA) are irrelevant. The 8279 will automatically drive the data bus for each subsequent read ($A_0=0$) in the same sequence in which the data first entered the FIFO. All subsequent reads will be from the FIFO until another command is issued.

In the Sensor Matrix Mode, the RAM address bits AAA select one of the 8 rows of the Sensor RAM. If the AI flag is set (AI = 1), each successive read will be from the subsequent row of the sensor RAM.

Read Display RAM

Code:

0	1	1	AI	A	A	A	A

The CPU sets up the 8279 for a read of the Display RAM by first writing this command. The address bits AAAA select one of the 16 rows of the Display RAM. If the AI flag is set (AI = 1), this row address will be incremented after each following read *or write* to the Display RAM. Since the same counter is used for both reading and writing, this command sets the next read *or write* address and the sense of the Auto-Increment mode for both operations.

Write Display RAM

Code:

1	0	0	AI	A	A	A	A

The CPU sets up the 8279 for a write to the Display RAM by first writing this command. After writing the command with $A_0=1$, all subsequent writes with $A_0=0$ will be to the Display RAM. The addressing and Auto-Increment functions are identical to those for the Read Display RAM. However, this command does not affect the source of subsequent Data Reads; the CPU will read from whichever RAM (Display or FIFO/Sensor) which was last specified. If, indeed, the Display RAM was last specified, the Write Display RAM will, nevertheless, change the next Read location.

Display Write Inhibit/Blanking

The IW Bits can be used to mask nibble A and nibble B in applications requiring separate 4-bit display ports. By setting the IW flag (IW = 1) for one of the ports, the port becomes marked so that entries to the Display RAM from the CPU do not affect that port. Thus, if each nibble is input to a BCD decoder, the CPU may write a digit to the Display RAM without affecting the other digit being displayed. It is important to note that bit B_0 corresponds to bit D_0 on the CPU bus, and that bit A_3 corresponds to bit D_7.

If the user wishes to blank the display, the BL flags are available for each nibble. The last Clear command issued determines the code to be used as a "blank." This code defaults to all zeros after a reset. Note that both BL flags must be set to blank a display formatted with a single 8-bit port.

Clear

The C_D bits are available in this command to clear all rows of the Display RAM to a selectable blanking code as follows:

C_D	C_D	C_D	
	0	X	All Zeros (X = Don't Care)
	1	0	AB = Hex 20 (0010 0000)
	1	1	All Ones

Enable clear display when = 1 (or by C_A = 1)

During the time the Display RAM is being cleared (~160 μs), it may not be written to. The most significant bit of the FIFO status word is set during this time. When the Display RAM becomes available again, it automatically resets.

If the C_F bit is asserted ($C_F=1$), the FIFO status is cleared and the interrupt output line is reset. Also, the Sensor RAM pointer is set to row 0.

C_A, the Clear All bit, has the combined effect of C_D and C_F; it uses the C_D clearing code on the Display RAM and also clears FIFO status. Furthermore, it resynchronizes the internal timing chain.

End Interrupt/Error Mode Set

Code:

1	1	1	E	X	X	X	X

X = Don't care.

For the sensor matrix modes this command lowers the IRQ line and enables further writing into RAM. (The IRQ line would have been raised upon the detection of a change in a sensor value. This would have also inhibited further writing into the RAM until reset).

For the N-key rollover mode — if the E bit is programmed to "1" the chip will operate in the special Error mode. (For further details, see Interface Considerations Section.)

Status Word

The status word contains the FIFO status, error, and display unavailable signals. This word is read by the CPU when A_0 is high and $\overline{CS}$ and $\overline{RD}$ are low. See Interface Considerations for more detail on status word.

Data Read

Data is read when A_0, $\overline{CS}$ and $\overline{RD}$ are all low. The source of the data is specified by the Read FIFO or Read Display commands. The trailing edge of $\overline{RD}$ will cause the address of the RAM being read to be incremented if the Auto-Increment flag is set. FIFO reads always increment (if no error occurs) independent of AI.

Data Write

Data that is written with A_0, $\overline{CS}$ and $\overline{WR}$ low is always written to the Display RAM. The address is specified by the latest Read Display or Write Display command. Auto-Incrementing on the rising edge of $\overline{WR}$ occurs if AI set by the latest display command.

INTERFACE CONSIDERATIONS

Scanned Keyboard Mode, 2-Key Lockout

There are three possible combinations of conditions that can occur during debounce scanning. When a key is depressed, the debounce logic is set. Other depressed keys are looked for during the next two scans. If none are encountered, it is a single key depression and the key position is entered into the FIFO along with the status of CNTL and SHIFT lines. If the FIFO was empty, IRQ will be set to signal the CPU that there is an entry in the FIFO. If the FIFO was full, the key will not be entered and the error flag will be set. If another closed switch is encountered, no entry to the FIFO can occur. If all other keys are released before this one, then it will be entered to the FIFO. If this key is released before any other, it will be entirely ignored. A key is entered to the FIFO only once per depression, no matter how many keys were pressed along with it or in what order they were released. If two keys are depressed within the debounce cycle, it is a simultaneous depression. Neither key will be recognized until one key remains depressed alone. The last key will be treated as a single key depression.

Scanned Keyboard Mode, N-Key Rollover

With N-key Rollover each key depression is treated independently from all others. When a key is depressed, the debounce circuit waits 2 keyboard scans and then checks to see if the key is still down. If it is, the key is entered into the FIFO. Any number of keys can be depressed and another can be recognized and entered into the FIFO. If a simultaneous depression occurs, the keys are recognized and entered according to the order the keyboard scan found them.

Scanned Keyboard — Special Error Modes

For N-key rollover mode the user can program a special error mode. This is done by the "End Interrupt/Error Mode Set" command. The debounce cycle and key-validity check are as in normal N-key mode. If during a single debounce cycle, two keys are found depressed, this is considered a simultaneous multiple depression, and sets an error flag. This flag will prevent any further writing into the FIFO and will set interrupt (if not yet set). The error flag could be read in this mode by reading the FIFO STATUS word. (See "FIFO STATUS" for further details.) The error flag is reset by sending the normal CLEAR command with C_F = 1.

Sensor Matrix Mode

In Sensor Matrix mode, the debounce logic is inhibited. The status of the sensor switch is inputted directly to the Sensor RAM. In this way the Sensor RAM keeps an image of the state of the switches in the sensor matrix. Although debouncing is not provided, this mode has the advantage that the CPU knows how long the sensor was closed and when it was released. A keyboard mode can only indicate a validated closure. To make the software easier, the designer should functionally group the sensors by row since this is the format in which the CPU will read them.

The IRQ line goes high if any sensor value change is detected at the end of a sensor matrix scan. The IRQ line is cleared by the first data read operation if the Auto-Increment flag is set to zero, or by the End Interrupt command if the Auto-Increment flag is set to one.

Note: Multiple changes in the matrix Addressed by (SL_{0-3} = 0) may cause multiple interrupts. (SL_0 = 0 in the Decoded Mode). Reset may cause the 8279 to see multiple changes.

Data Format

In the Scanned Keyboard mode, the character entered into the FIFO corresponds to the position of the switch in the keyboard plus the status of the CNTL and SHIFT lines (non-inverted). CNTL is the MSB of the character and SHIFT is the next most significant bit. The next three bits are from the scan counter and indicate the row the key was found in. The last three bits are from the column counter and indicate to which return line the key was connected.

MSB							LSB
CNTL	SHIFT	SCAN			RETURN		

SCANNED KEYBOARD DATA FORMAT

In Sensor Matrix mode, the data on the return lines is entered directly in the row of the Sensor RAM that corresponds to the row in the matrix being scanned. Therefore, each switch postion maps directly to a Sensor RAM position. The SHIFT and CNTL inputs are ignored in this mode. Note that switches are not necessarily the only thing that can be connected to the return lines in this mode. Any logic that can be triggered by the scan lines can enter data to the return line inputs. Eight multiplexed input ports could be tied to the return lines and scanned by the 8279.

MSB							LSB
RL_7	RL_6	RL_5	RL_4	RL_3	RL_2	RL_1	RL_0

In Strobed Input mode, the data is also entered to the FIFO from the return lines. The data is entered by the rising edge of a CNTL/STB line pulse. Data can come from another encoded keyboard or simple switch matrix. The return lines can also be used as a general purpose strobed input.

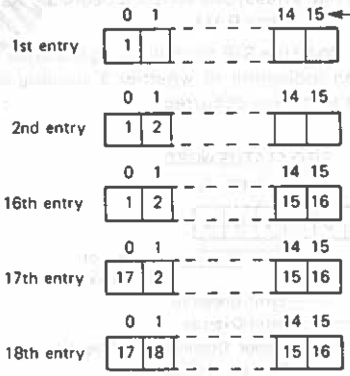
MSB							LSB
RL_7	RL_6	RL_5	RL_4	RL_3	RL_2	RL_1	RL_0

Display

Left Entry

Left Entry mode is the simplest display format in that each display position directly corresponds to a byte (or nibble) in the Display RAM. Address 0 in the RAM is the left-most display character and address 15 (or address 7 in 8 character display) is the right most display character. Entering characters from position zero causes the display to fill from the left. The 17th (9th) character is entered back in the left most position and filling again proceeds from there.

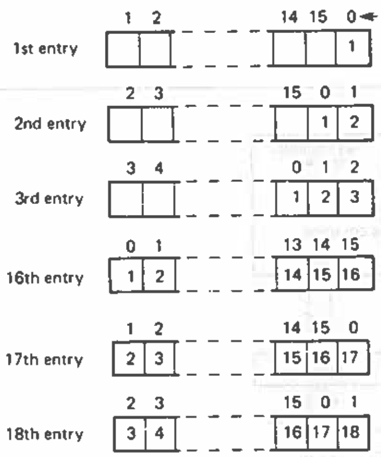
MPU PERIPHERALS



LEFT ENTRY MODE
(AUTO INCREMENT)

Right Entry

Right entry is the method used by most electronic calculators. The first entry is placed in the right most display character. The next entry is also placed in the right most character after the display is shifted left one character. The left most character is shifted off the end and is lost.

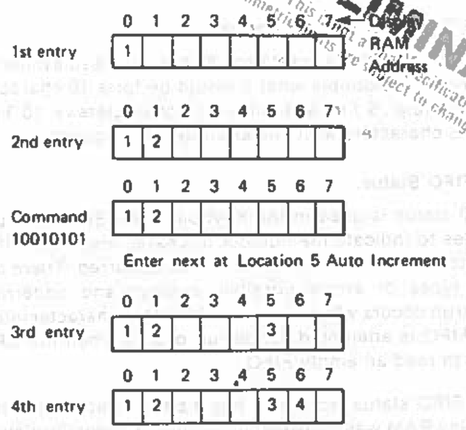


RIGHT ENTRY MODE
(AUTO INCREMENT)

Note that now the display position and register address do not correspond. Consequently, entering a character to an arbitrary position in the Auto Increment mode may have unexpected results. Entry starting at Display RAM address 0 with sequential entry is recommended.

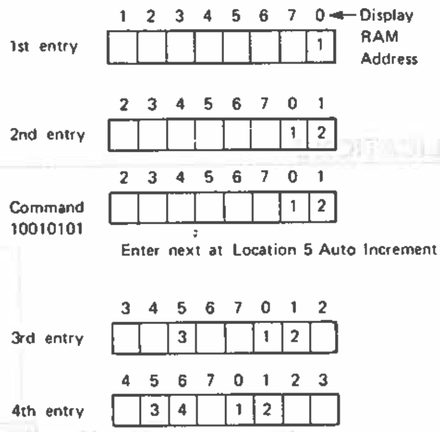
Auto Increment

In the Left Entry mode, Auto Incrementing causes the address where the CPU will next write to be incremented by one and the character appears in the next location. With non-Auto Incrementing the entry is both to the same RAM address and display position. Entry to an arbitrary address in the Auto Increment mode has no undesirable side effects and the result is predictable:



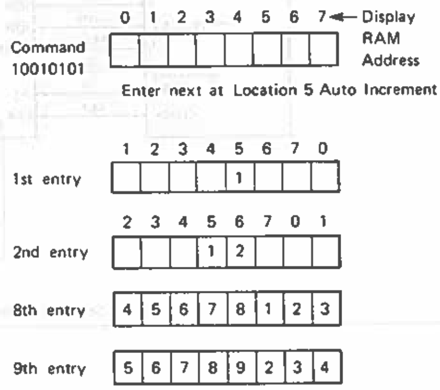
LEFT ENTRY MODE
(AUTO INCREMENT)

In the Right Entry mode, Auto Incrementing and non Incrementing have the same effect as in the Left Entry except if the address sequence is interrupted:



RIGHT ENTRY MODE
(AUTO INCREMENT)

Starting at an arbitrary location operates as shown below:



RIGHT ENTRY MODE
(AUTO INCREMENT)

Entry appears to be from the initial entry point.

8/16 Character Display Formats

If the display mode is set to an 8 character display: the on duty-cycle is double what it would be for a 16 character display (e.g., 5.1 ms scan time for 8 characters vs. 10.3 ms for 16 characters with 100 kHz internal frequency).

G. FIFO Status

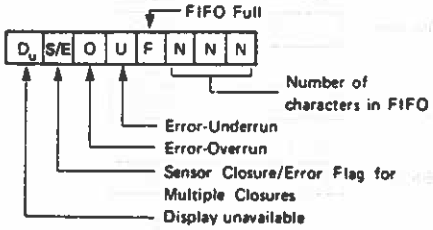
FIFO status is used in the Keyboard and Strobed Input modes to indicate the number of characters in the FIFO and to indicate whether an error has occurred. There are two types of errors possible: overrun and underrun. Overrun occurs when the entry of another character into a full FIFO is attempted. Underrun occurs when the CPU tries to read an empty FIFO.

The FIFO status word also has a bit to indicate that the Display RAM was unavailable because a Clear Display or Clear All command had not completed its clearing operation.

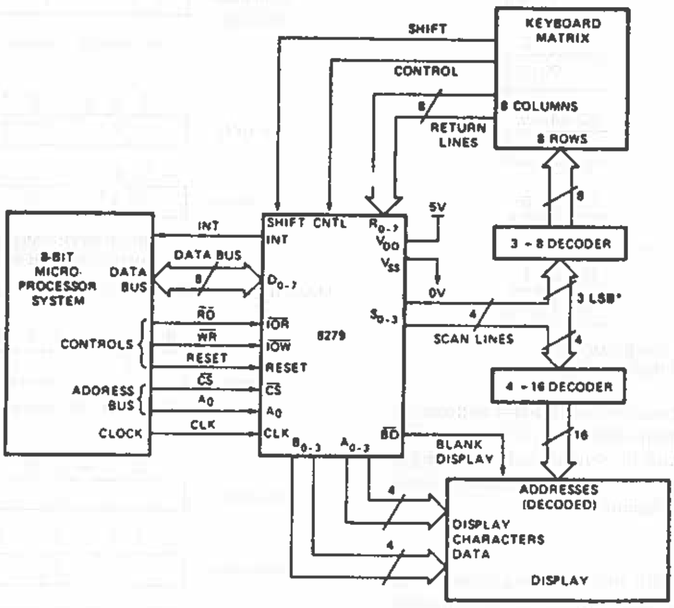
In a Sensor Matrix mode, a bit is set in the FIFO status word to indicate that at least one sensor closure indication is contained in the Sensor RAM.

In Special Error Mode the S/E bit is showing the error flag and serves as an indication to whether a simultaneous multiple closure error has occurred.

FIFO STATUS WORD



APPLICATIONS



*Do not drive the keyboard decoder with the MSB of the scan lines.

MPU PERIPHERALS

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature .... 0°C to 70°C
Storage Temperature .... -65°C to 125°C
Voltage on any Pin with
Respect to Ground .... -0.5V to +7V
Power Dissipation .... 1 Watt

COMMENT: Stresses above 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 device reliability.

PRELIMINARY

D.C. CHARACTERISTICS

T_A = 0°C to 70°C, V_{SS} = 0V, Note 1

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
V_{IL1}	Input Low Voltage for Return Lines	-0.5	1.4	V	
V_{IL2}	Input Low Voltage for All Others	-0.5	0.8	V	
V_{IH1}	Input High Voltage for Return Lines	2.2		V	
V_{IH2}	Input High Voltage for All Others	2.0		V	
V_{OL}	Output Low Voltage		0.45	V	Note 2
V_{OH}	Output High Voltage on Interrupt Line	3.5		V	Note 3
I_{IL1}	Input Current on Shift, Control and Return Lines		+10 -100	μA μA	$V_{IN} = V_{CC}$ V_{IN} = 0V
I_{IL2}	Input Leakage Current on All Others		±10	μA	$V_{IN} = V_{CC}$ to 0V
I_{OFL}	Output Float Leakage		±10	μA	$V_{OUT} = V_{CC}$ to 0V
I_{CC}	Power Supply Current		120	mA	

Notes:
1. 8279, V_{CC} = +5V ±5%; 8279-5, V_{CC} = +5V ±10%.
2. 8279, I_{OL} = 1.6mA; 8279-5, I_{OL} = 2.2mA.
3. 8279, I_{OH} = -100μA; 8279-5, I_{OH} = -400μA.

CAPACITANCE

SYMBOL	TEST	TYP.	MAX.	UNIT	TEST CONDITIONS
C_{in}	Input Capacitance	5	10	pF	$V_{in}=V_{CC}$
C_{out}	Output Capacitance	10	20	pF	$V_{out}=V_{CC}$

MPU PERIPHERALS

A.C. CHARACTERISTICS

$T_A = 0°C$ to $70°C$, $V_{SS} = 0V$, (Note 1)

Bus Parameters

Read Cycle:

Symbol	Parameter	8279 Min.	8279 Max.	8279-5 Min.	8279-5 Max.	Unit
t_{AR}	Address Stable Before $\overline{\text{READ}}$	50		0		ns
t_{RA}	Address Hold Time for $\overline{\text{READ}}$	5		0		ns
t_{RR}	$\overline{\text{READ}}$ Pulse Width	420		250		ns
t_{RD}[2]	Data Delay from $\overline{\text{READ}}$		300		150	ns
t_{AD}[2]	Address to Data Valid		450		250	ns
t_{DF}	$\overline{\text{READ}}$ to Data Floating	10	100	10	100	ns
t_{RCY}	Read Cycle Time	1		1		μs

Write Cycle:

Symbol	Parameter	8279 Min.	8279 Max.	8279-5 Min.	8279-5 Max.	Unit
t_{AW}	Address Stable Before $\overline{\text{WRITE}}$	50		0		ns
t_{WA}	Address Hold Time for $\overline{\text{WRITE}}$	20		0		ns
t_{WW}	$\overline{\text{WRITE}}$ Pulse Width	400		250		ns
t_{DW}	Data Set Up Time for $\overline{\text{WRITE}}$	300		150		ns
t_{WD}	Data Hold Time for $\overline{\text{WRITE}}$	40		0		ns

Notes:
1. 8279, $V_{CC} = +5V \pm 5\%$; 8279-5, $V_{CC} = +5V \pm 10\%$.
2. 8279, $C_L = 100pF$; 8279-5, $C_L = 150pF$.

Other Timings:

Symbol	Parameter	8279 Min.	8279 Max.	8279-5 Min.	8279-5 Max.	Unit
$t_{\phi W}$	Clock Pulse Width	230		120		nsec
t_{CY}	Clock Period	500		320		nsec

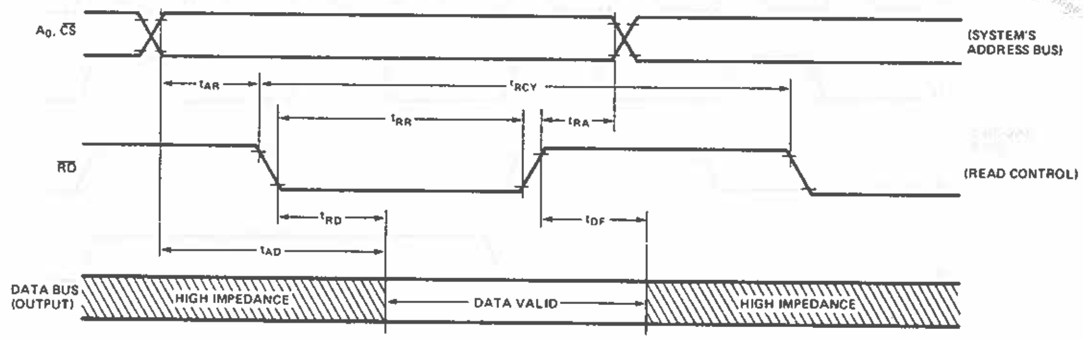
Keyboard Scan Time:	5.1 msec	Digit-on Time:	480 μsec
Keyboard Debounce Time:	10.3 msec	Blanking Time:	160 μsec
Key Scan Time:	80 μsec	Internal Clock Cycle:	10 μsec
Display Scan Time:	10.3 msec		

MPU PERIPHERALS

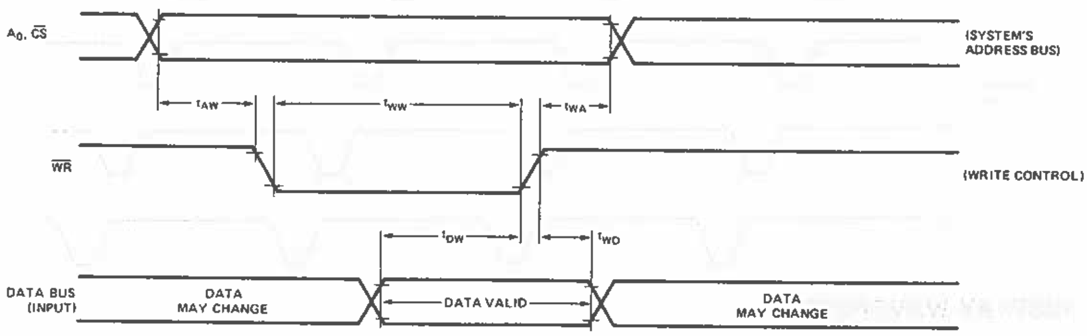
Input Waveforms For A.C. Tests

WAVEFORMS

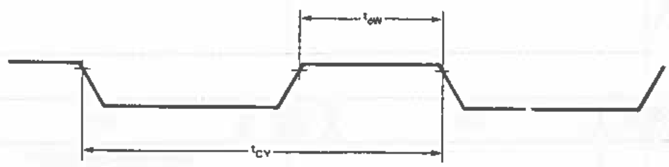
Read Operation



Write Operation



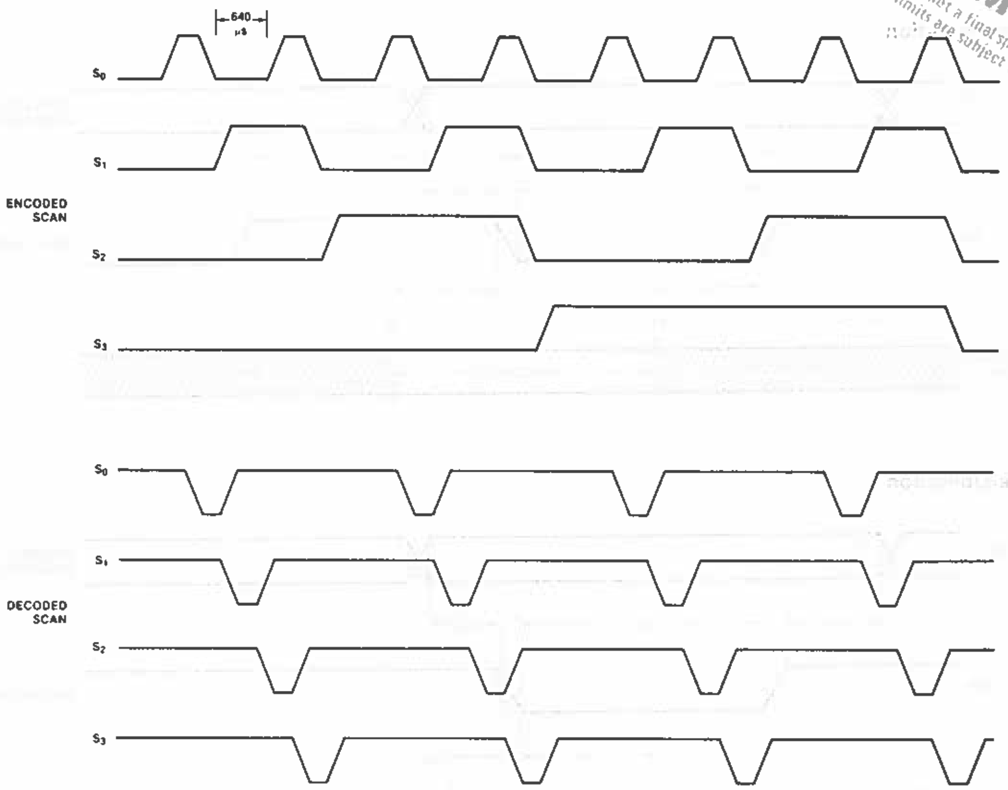
Clock Input



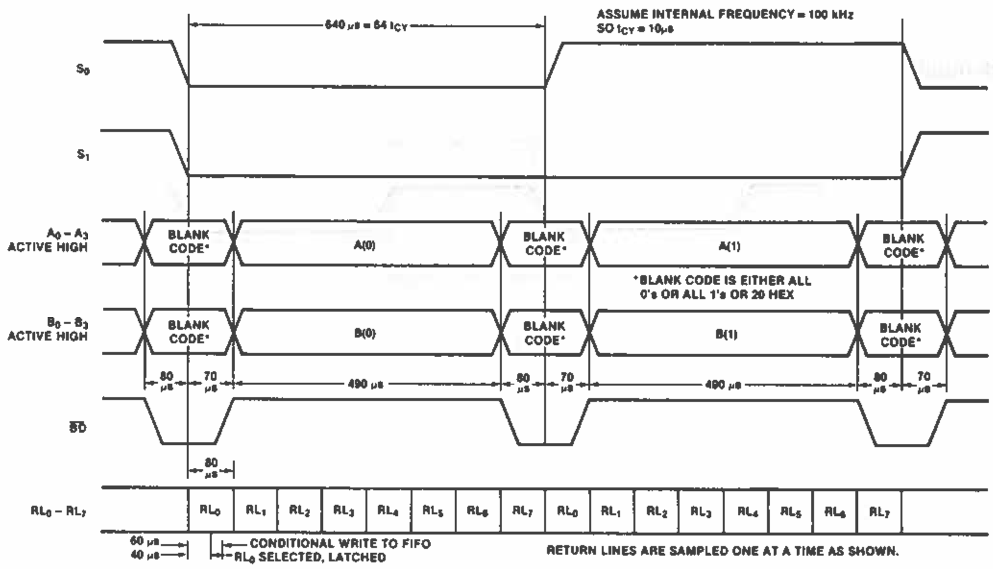
MPU PERIPHERALS

8279 SCAN TIMING

SCAN WAVEFORMS



DISPLAY WAVEFORMS



NOTE: SHOWN IS ENCODED SCAN LEFT ENTRY
S_2-S_3 ARE NOT SHOWN BUT THEY ARE SIMPLY S_1 DIVIDED BY 2 AND 4

MPU PERIPHERALS

Fiche technique :

EPROM 2708

PROM/ROM

intel®

2708/8708*
8K AND 4K UV ERASABLE PROM

	Max. Power	Max. Access	Organization
2708	800 mW	450 ns	1K × 8
2708L	425 mW	450 ns	1K × 8
2708-1	800 mW	350 ns	1K × 8
2704	800 mW	450 ns	512 × 8

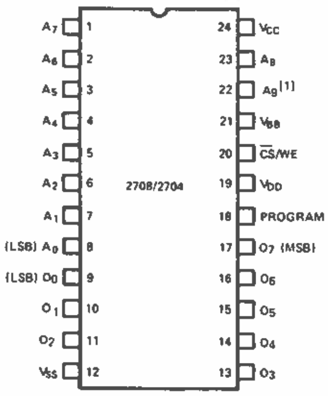
- **Low Power Dissipation — 425 mW Max. (2708L)**
- **Fast Access Time — 350 ns Max. (2708-1)**
- **Static — No Clocks Required**
- **Data Inputs and Outputs TTL Compatible during both Read and Program Modes**
- **Three-State Outputs — OR-Tie Capability**

The Intel® 2708 is a 8192-bit ultraviolet light erasable and electrically reprogrammable EPROM, ideally suited where fast turnaround and pattern experimentation are important requirements. All data inputs and outputs are TTL compatible during both the read and program modes. The outputs are three-state, allowing direct interface with common system bus structures.

The 2708L at 425 mW is available for systems requiring lower power dissipation than from the 2708. A power dissipation savings of over 50%, without any sacrifice in speed, is obtained with the 2708L. The 2708L has high input noise immunity and is specified at 10% power supply tolerance. A high-speed 2708-1 is also available at 350 ns for microprocessors requiring fast access times. For smaller size systems there is the 4096-bit 2704 which is organized as 512 words by 8 bits. All these devices have the same programming and erasing specifications of the 2708. The 2704 electrical specifications are the same as the 2708.

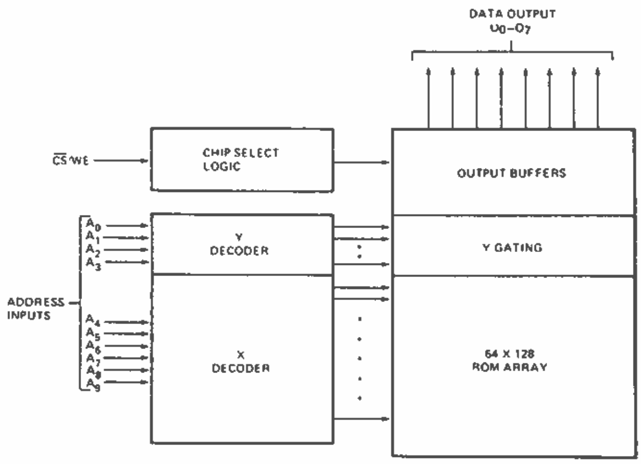
The 2708 family is fabricated with the N-channel silicon gate FAMOS technology and is available in a 24-pin dual in-line package.

PIN CONFIGURATION



NOTE 1: PIN 22 MUST BE CONNECTED TO V_{SS} FOR THE 2704.

BLOCK DIAGRAM



PIN NAMES

A_0-A_9	ADDRESS INPUTS
O_1-O_8	DATA OUTPUTS/INPUTS
$\overline{CS}$/WE	CHIP SELECT/WRITE ENABLE INPUT

PIN CONNECTION DURING READ OR PROGRAM

MODE	DATA I/O 9-11, 13-17	ADDRESS INPUTS 1-8, 22, 23	V_{SS} 12	PROGRAM 18	V_{DD} 19	$\overline{CS}$/WE 20	V_{BB} 21	V_{CC} 24
READ	D_{OUT}	A_{IN}	GND	GND	+12	V_{IL}	-5	+5
DESELECT	HIGH IMPEDANCE	DON'T CARE	GND	GND	+12	V_{IH}	-5	+5
PROGRAM	D_{IN}	A_{IN}	GND	PULSED 26V	+12	V_{IHW}	-5	+5

(The column headings are grouped under "PIN NUMBER".)

*All 8708 specifications are identical to the 2708 specifications.

2708 FAMILY

PROGRAMMING

The programming specifications are described in the Data Catalog PROM/ROM Programming Instructions Section.

Absolute Maximum Ratings*

Temperature Under Bias . . . . −25°C to +85°C
Storage Temperature . . . . −65°C to +125°C
V_{DD} With Respect to V_{BB} . . . . +20V to -0.3V
V_{CC} and V_{SS} With Respect to V_{BB} . . . . +15V to -0.3V
All Input or Output Voltages With Respect
to V_{BB} During Read . . . . +15V to -0.3V
$\overline{CS}$/WE Input With Respect to V_{BB}
During Programming . . . . +20V to -0.3V
Program Input With Respect to V_{BB} . . . . +35V to -0 3V
Power Dissipation 1.5W

*COMMENT
Stresses above 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 device reliability.

PROM/ROM

DC and AC Operating Conditions During Read

	2708	2708-1	2708L
Temperature Range	0°C – 70°C	0°C – 70°C	0°C – 70°C
V_{CC} Power Supply	5V ± 5%	5V ± 5%	5V ± 10%
V_{DD} Power Supply	12V ± 5%	12V ± 5%	12V ± 10%
V_{BB} Power Supply	-5V ± 5%	-5V ± 5%	-5V ± 10%

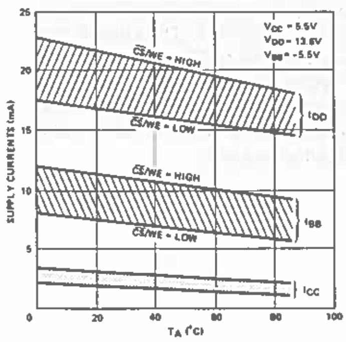
READ OPERATION

D.C. and Operating Characteristics

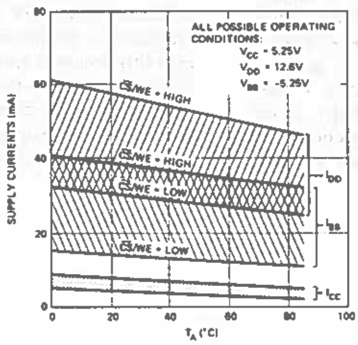
Symbol	Parameter	2708, 2708-1 Limits Min.	Typ.[2]	Max.	2708L Limits Min.	Typ.[2]	Max.	Units	Test Conditions
I_{LI}	Address and Chip Select Input Sink Current		1	10		1	10	μA	V_{IN} = 5.25V or V_{IN} = V_{IL}
I_{LO}	Output Leakage Current		1	10		1	10	μA	V_{OUT} = 5.5V, CS/WE = 5V
I_{DD}[3]	V_{DD} Supply Current		50	65		21	28	mA	Worst Case Supply Currents[4]
I_{CC}[3]	V_{CC} Supply Current		6	10		2	4	mA	All Inputs High;
I_{BB}[3]	V_{BB} Supply Current		30	45		10	14	mA	$\overline{CS}$/WE = 5V; T_A = 0°C
V_{IL}	Input Low Voltage	V_{SS}		0.65	V_{SS}		0.65	V	
V_{IH}	Input High Voltage	3.0		V_{CC}+1	2.2		V_{CC}+1	V	
V_{OL}	Output Low Voltage			0.45			0.4	V	I_{OL} = 1.6mA (2708, 2708-1) / I_{OL} = 2mA (2708L)
V_{OH1}	Output High Voltage	3.7			3.7			V	I_{OH} = −100 μA
V_{OH2}	Output High Voltage	2.4			2.4			V	I_{OH} = −1 mA
PD	Power Dissipation			800			325	mW	T_A = 70°C
							425	mW	T_A = 0°C

NOTES:
1. V_{BB} must be applied prior to V_{CC} and V_{DD}. V_{BB} must also be the last power supply switched off
2. Typical values are for T_A = 25°C and nominal supply voltages.
3. The total power disiipation is not calculated by summing the various currents (I_{DD}, I_{CC}, and I_{BB}) multiplied by their respective voltages since current paths exist between the various power supplies and V_{SS}. The I_{DD}, I_{CC}, and I_{BB} currents should be used to determine power supply capacity only.
4. I_{BB} for the 2708L is specified in the programmed state and is 18 mA maximum in the unprogrammed state.

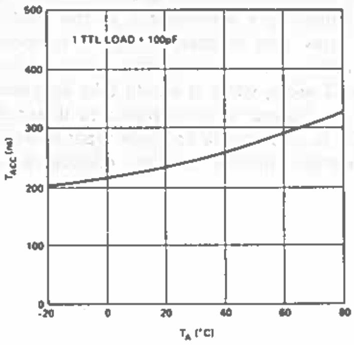
2708L
RANGE OF SUPPLY CURRENTS VS. TEMPERATURE



2708 AND 2708-1
RANGE OF SUPPLY CURRENTS VS. TEMPERATURE



ACCESS TIME VS. TEMPERATURE



A. C. Characteristics

Symbol	Parameter	2708-1 Limits Min.	Typ.	Max.	2708, 2708L, Limits Min.	Typ.	Max.	Units
t_{ACC}	Address to Output Delay		280	350		280	450	ns
t_{CO}	Chip Select to Output Delay		60	120		60	120	ns
t_{DF}	Chip Deselect to Output Float	0		120	0		120	ns
t_{OH}	Address to Output Hold	0			0			ns

CAPACITANCE[1] $T_A = 25°C$, $f = 1$ MHz

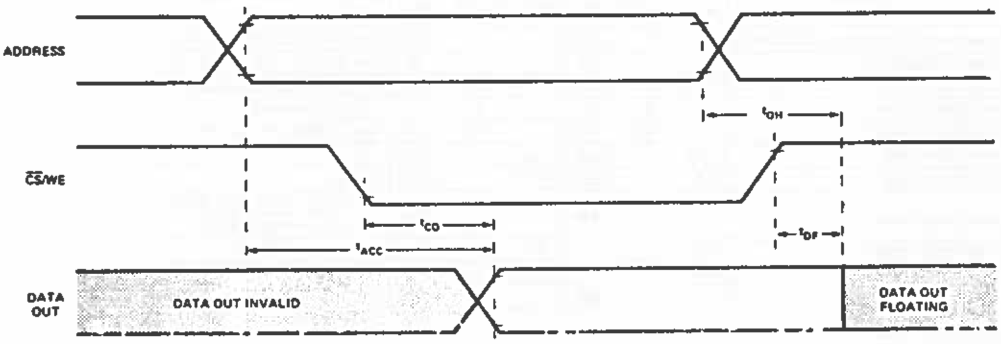
Symbol	Parameter	Typ.	Max.	Unit.	Conditions
C_{IN}	Input Capacitance	4	6	pF	$V_{IN} = 0V$
C_{OUT}	Output Capacitance	8	12	pF	$V_{OUT} = 0V$

Note: 1. This parameter is periodically sampled and is not 100% tested

A.C. TEST CONDITIONS:

Output Load: 1 TTL gate and $C_L = 100$ pF
Input Rise and Fall Times: ≤20 ns
Timing Measurement Reference Levels: 0.8V and 2.8V for inputs; 0.8V and 2.4V for outputs.
Input Pulse Levels: 0.65V to 3.0V

Waveforms



ERASURE CHARACTERISTICS

The erasure characteristics of the 2708 family are such that erasure begins to occur when exposed to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000–4000Å range. Data show that constant exposure to room level fluorescent lighting could erase the typical device in approximately 3 years, while it would take approximately 1 week to cause erasure when exposed to direct sunlight. If the 2708 is to be exposed to these types of lighting conditions for extended periods of time, opaque labels are available form Intel which should be placed over the 2708 window to prevent unintentional erasure.

The recommended erasure procedure (see Data Catalog PROM/ROM Programming Instructions Section) for the 2708 family is exposure to shortwave ultraviolet light which has a wavelength of 2537 Angstroms (Å). The integrated dose (i.e., UV intensity X exposure time) for erasure should be a minimum of 15 W-sec/cm^2. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with a 12000 μW/cm^2 power rating. The device should be placed within 1 inch of the lamp tubes during erasure. Some lamps have a filter on their tubes which should be removed before erasure.

Fiche technique :

RAM 2114

Synertek® INCORPORATED

1024x4 Static Random Access Memory

SY2114

MEMORY PRODUCTS

SEPTEMBER 1978

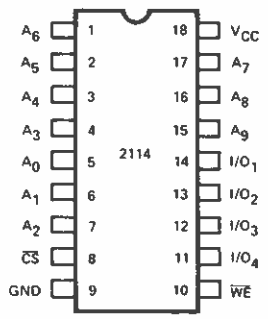
- 200 ns Maximum Access
- Low Operating Power Dissipation 0.1 mW/Bit
- No Clocks or Strobes Required
- Identical Cycle and Access Times
- Single +5V Supply
- Totally TTL Compatible: All Inputs, Outputs, and Power Supply
- Common Data I/O
- 400 mv Noise Immunity
- High Density 18 Pin Package

The SY2114 is a 4096-Bit static Random Access Memory organized 1024 words by 4-bits and is fabricated using Synertek's N-channel Silicon-Gate MOS technology. It is designed using fully DC stable (static) circuitry in both the memory array and the decoding and therefore requires no clock or refreshing to operate. Address setup times are not required and the data is read out nondestructively with the same polarity as the input data. Common Input/Output pins are provided to simplify design of the bus oriented systems, and can drive 1 TTL load.

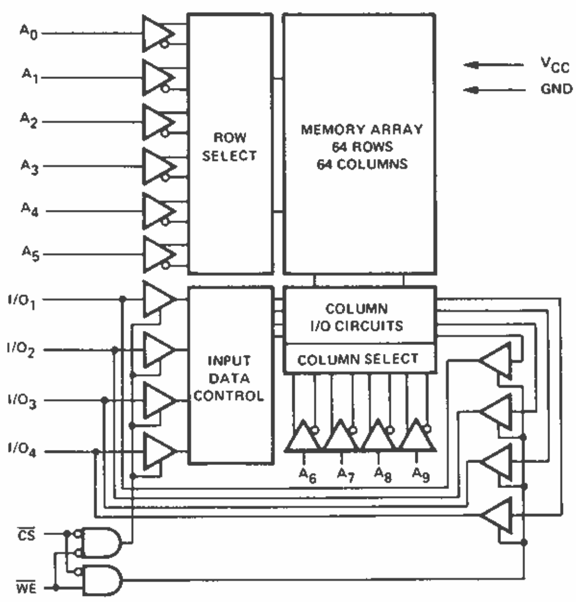
The SY2114 is designed for memory applications where high performance, low cost, large bit storage, and simple interfacing are important design objectives. It is totally TTL compatible in all respects: inputs, outputs, and the single +5V supply. A separate Chip Select ($\overline{CS}$) input allows easy selection of an individual device when outputs are or-tied.

The SY2114 is packaged in an 18-pin DIP for the highest possible density and is fabricated with N-channel, Ion Implanted, Silicon-Gate technology — a technology providing excellent performance characteristics as well as improved protection against contamination.

PIN CONFIGURATION



BLOCK DIAGRAM



ORDERING INFORMATION

Order Number	Package Type	Access Time	Supply Current (Max)	Temperature Range
SYC2114	Ceramic	450nsec	100mA	0°C to 70°C
SYP2114	Molded	450nsec	100mA	0°C to 70°C
SYC2114-3	Ceramic	300nsec	100mA	0°C to 70°C
SYP2114-3	Molded	300nsec	100mA	0°C to 70°C
SYC2114L	Ceramic	450nsec	70mA	0°C to 70°C
SYP2114L	Molded	450nsec	70mA	0°C to 70°C
SYC2114L-3	Ceramic	300nsec	70mA	0°C to 70°C
SYP2114L-3	Molded	300nsec	70mA	0°C to 70°C
SYC2114-2	Ceramic	200nsec	100mA	0°C to 70°C
SYP2114-2	Molded	200nsec	100mA	0°C to 70°C
SYC2114L-2	Ceramic	200nsec	70mA	0°C to 70°C
SYP2114L-2	Molded	200nsec	70mA	0°C to 70°C

SYNERTEK® INC. • P.O. BOX 552 • SANTA CLARA, CALIFORNIA 95052 • TEL. (408) 988-5600 • TWX: 910-338-0135

ABSOLUTE MAXIMUM RATINGS

Temperature Under Bias	$-10°C$ to $80°C$
Storage Temperature	$-65°C$ to $150°C$
Voltage on Any Pin with Respect to Ground	$-0.5V$ to $+7$ V
Power Dissipation	1.0W

COMMENT

Stresses above 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.

D.C. CHARACTERISTICS $T_A = 0°C$ to $+70°C$, $V_{CC} = 5V \pm 5\%$ (Unless otherwise specified)

Symbol	Parameter	2114-2 2114-3,2114 Min	2114-2 2114-3,2114 Max	2114L-2 2114L,2114L-3 Min	2114L-2 2114L,2114L-3 Max	Unit	Conditions
I_{LI}	Input Load Current (All input pins)		10		10	μA	$V_{IN} = 0$ to 5.25V
I_{LO}	I/O Leakage Current		10		10	μA	$\overline{CS} = 2.0V$, $V_{I/O} = 0.4V$ to V_{CC}
I_{CC1}	Power Supply Current		95		65	mA	$V_{CC} = 5.25V$, $I_{I/O} = 0$ mA, $T_A = 25°C$
I_{CC2}	Power Supply Current		100		70	mA	$V_{CC} = 5.25V$, $I_{I/O} = 0$ mA, $T_A = 0°C$
V_{IL}	Input Low Voltage	-0.5	0.8	-0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC}	2.0	V_{CC}	V	
V_{OL}	Output Low Voltage		0.4		0.4	V	$I_{OL} = 3.2$ mA
V_{OH}	Output High Voltage	2.4	V_{CC}	2.4	V_{CC}	V	$I_{OH} = -1.0$ mA

CAPACITANCE $T_A = 25°C$, f = 1.0 MHz

Symbol	Test	Typ	Max	Units
$C_{I/O}$	Input/Output Capacitance		5	pF
C_{IN}	Input Capacitance		5	pF

NOTE: This parameter is periodically sampled and not 100% tested.

A.C. CHARACTERISTICS $T_A = 0°C$ to $70°C$, $V_{CC} = 5V \pm 5\%$ (Unless Otherwise Specified)

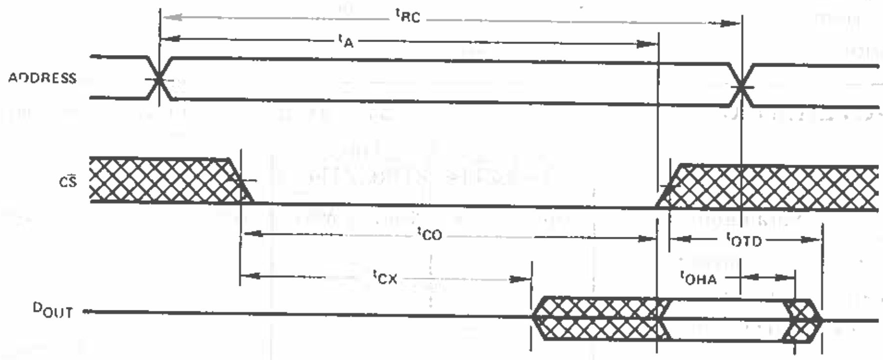
Symbol	Parameter	2114-2,2114L-2 Min.	2114-2,2114L-2 Max.	2114-3,2114L-3 Min.	2114-3,2114L-3 Max.	2114,2114L Min.	2114,2114L Max.	Unit
Read Cycle								
t_{RC}	Read Cycle Time	200		300		450		nsec
t_A	Access Time		200		300		450	nsec
t_{CO}	Chip Select to Output Valid		70		100		120	nsec
t_{CX}	Chip Select to Output Enabled	20		20		20		nsec
t_{OTD}	Chip Deselect to Output Off	0	60	0	80	0	100	nsec
t_{OHA}	Output Hold From Address Change	50		50		50		nsec
Write Cycle								
t_{WC}	Write Cycle Time	200		300		450		nsec
t_{AW}	Address to Write Setup Time	0		0		0		nsec
t_W	Write Pulse Width	120		150		200		nsec
t_{WR}	Write Release Time	0		0		0		nsec
t_{OTW}	Write to Output Off	0	60	0	80	0	100	nsec
t_{DW}	Data to Write Overlap	120		150		200		nsec
t_{DH}	Data Hold	0		0		0		nsec

A.C. Test Conditions

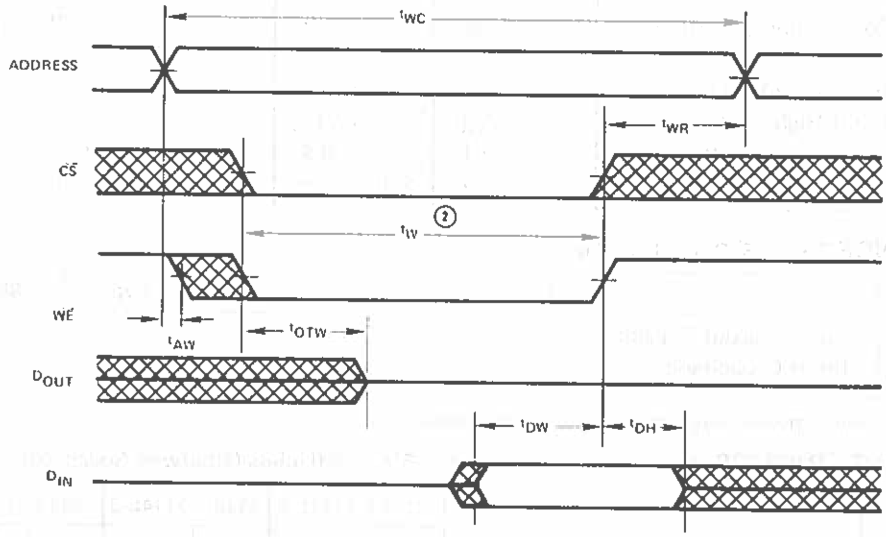
Input Pulse Levels . . . . 0.8V to 2.0V
Input Rise and Fall Time . . . . 10 n sec
Timing Measurement Levels: Input 1.5V
Output . . . . 0.8 and 2.0V
Output Load . . . . 1TTL Gate and 100pF

TIMING DIAGRAMS

Read Cycle ①



Write Cycle



NOTES:

① $\overline{WE}$ is high for a Read Cycle
② t_W is measured from the latter of $\overline{CS}$ or $\overline{WE}$ going low to the earlier of $\overline{CS}$ or $\overline{WE}$ going high.

DATA STORAGE

When $\overline{WE}$ is high, the data input buffers are inhibited to prevent erroneous data from being written into the array. As long as $\overline{WE}$ remains high, the data stored cannot be affected by the Address, Chip Select, or Data I/O logic levels or timing transitions.

Data storage also cannot be affected by $\overline{WE}$, Addresses, or the I/O ports as long as $\overline{CS}$ is high. Either $\overline{CS}$ or $\overline{WE}$ or both can prevent extraneous writing due to signal transitions.

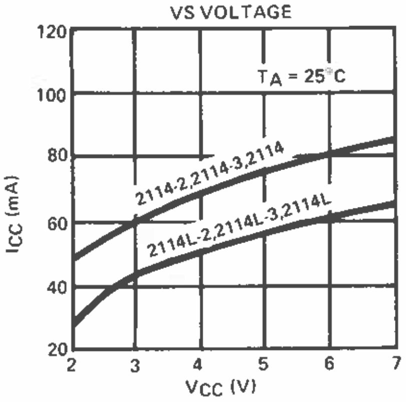
Data within the array can only be changed during Write time — defined as the overlap of CS low and $\overline{WE}$ low. The addresses must be properly established during the entire Write time plus t_{WR}.

Internal delays are such that address decoding propagates ahead of data inputs and therefore no address setup time is required. If the Write time precedes the addresses, the data in previously addressed locations, or some other location, may be changed. Addresses must remain stable for the entire Write cycle but the Data Inputs may change. The data which is stable for t_{DW} at the end of the Write time will be written into the addressed location.

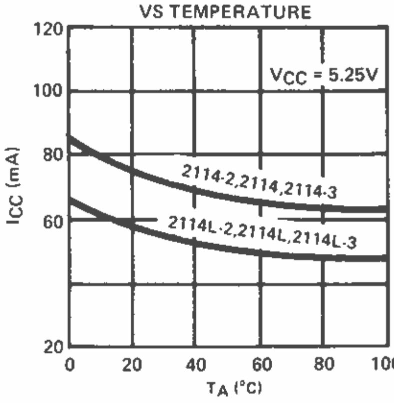
SY2114

TYPICAL CHARACTERISTICS

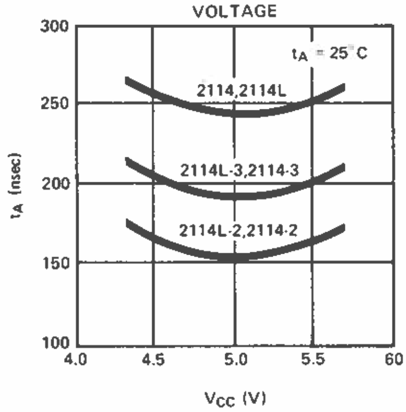
SUPPLY CURRENT VS VOLTAGE



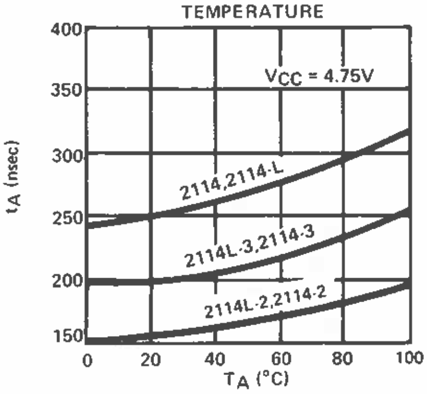
SUPPLY CURRENT VS TEMPERATURE



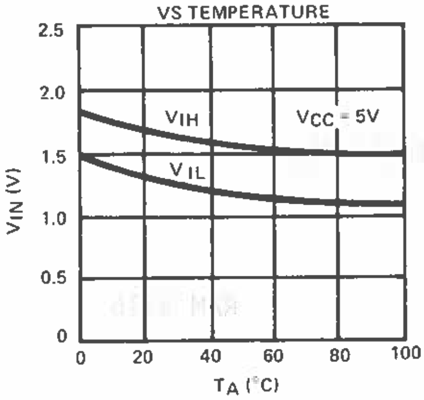
ACCESS TIME VS VOLTAGE



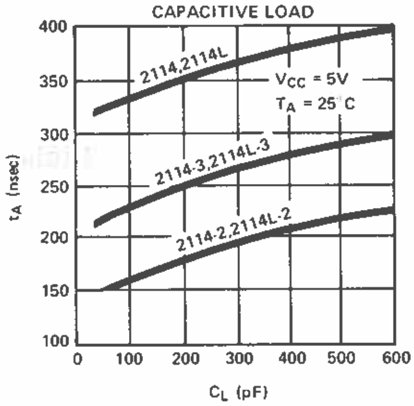
ACCESS TIME VS TEMPERATURE



INPUT VOLTAGE LIMITS VS TEMPERATURE

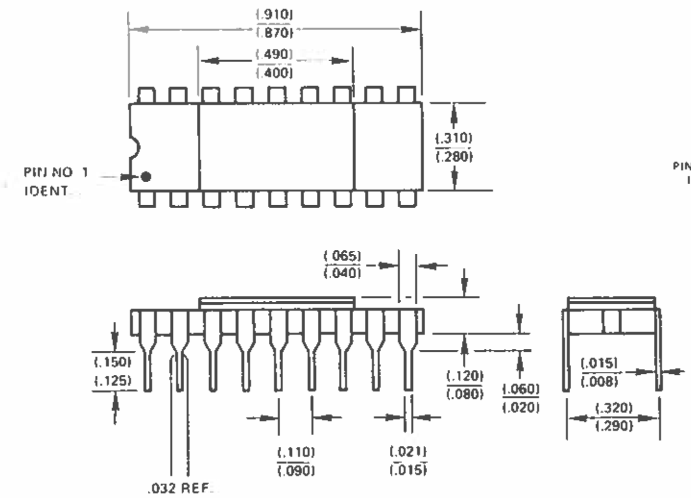


ACCESS TIME VS CAPACITIVE LOAD

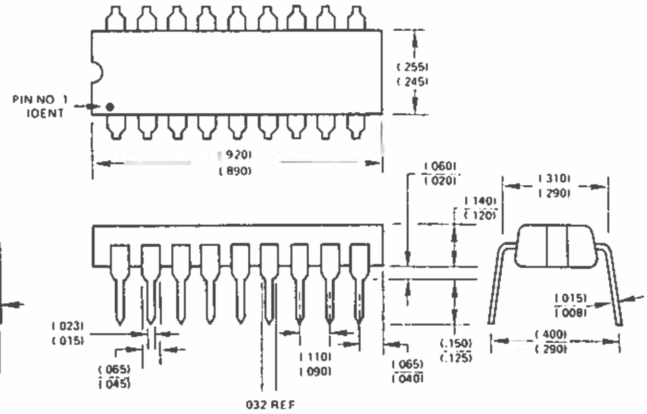


PACKAGE DIAGRAM

CERAMIC PACKAGE



MOLDED PACKAGE



SYNERTEK® INC. • P.O. BOX 552 • SANTA CLARA, CALIFORNIA 95052 • TEL. (408) 988-5600 • TWX: 910-338-0135

Fiche technique :

RAM 4116

MOSTEK

16,384 X 1-BIT DYNAMIC RAM

MK4116(P/N)-2/3

16,384x1-BIT DYN 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)

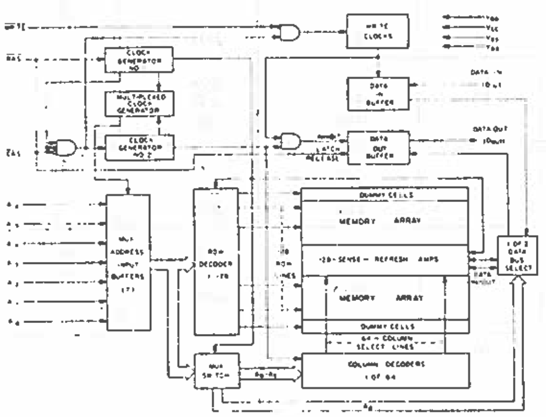
DESCRIPTION

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).

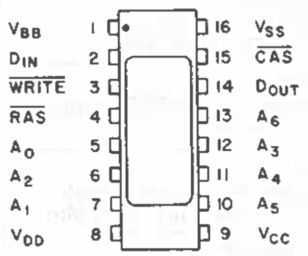
The technology used to fabricate the MK 4116 is MOSTEK's double-poly, N-channel silicon gate, POLY II® process. This process, coupled with the use of a single transistor dynamic storage cell, provides the maximum possible circuit density and reliability, while maintaining high performance capability. 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]
($0°C \leq T_A \leq 70°C$)

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
($0°C \leq TA \leq 70°C$) ($VDD = 12.0V \pm 10\%$; $VCC = 5.0V \pm 10\%$; $-5.7V \leq VBB \leq -4.5V$; $VSS = 0V$)

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, $0V \leq V_{IN} \leq +7.0V$, all other pins not under test = 0 volts)	$I_{I(L)}$	−10	10	µA	
OUTPUT LEAKAGE Output leakage current (D_{OUT} is disabled, $0V \leq V_{OUT} \leq +5.5V$)	$I_{O(L)}$	−10	10	µA	
OUTPUT LEVELS Output high (Logic 1) voltage (I_{OUT} = −5mA)	V_{OH}	2.4		Volts	3
Output low (Logic 0) voltage (I_{OUT} = 4.2 mA)	V_{OL}		0.4	Volts	

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 MIN	MK 4116-2 MAX	MK 4116-3 MIN	MK 4116-3 MAX	UNITS	NOTES
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 $t_T = 5$ns.
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 tRCD ≤ tRCD (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 tWCS ≥ tWCS (min), the cycle is an early write cycle and the data out pin will remain open circuit (high impedance) throughout the entire cycle. If tCWD ≥ tCWD (min) and tRWD ≥ tRWD (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 $C = \frac{I\Delta t}{\Delta V}$ with $\Delta V = 3$ 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

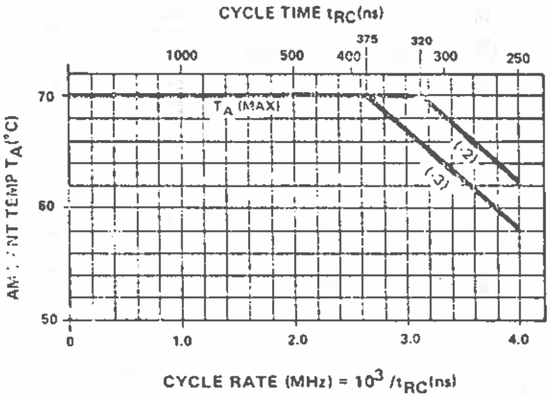


Fig. 1 Maximum ambient temperature versus cycle rate for extended frequency operation. T_A (max) for operation at cycling rates greater than 2.66 MHz (t_{CYC} < 375ns) is determined by T_A (max)° C = 70 − 9.0 × (cycle rate MHz − 2.66) for -3. T_A (max) °C = 70 − 9.0 × cycle rate MHz − 3.125MHz) for -2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

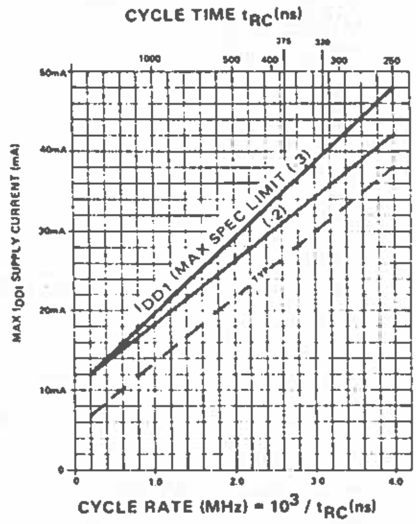


Fig. 2 Maximum I_{DD1} versus cycle rate for device operation at extended frequencies. I_{DD1} (max) curve is defined by the equation:

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

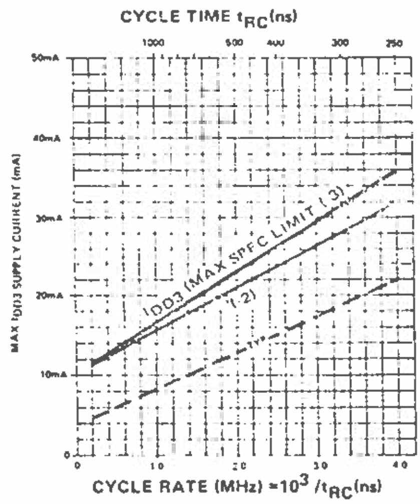


Fig. 3 Maximum I_{DD3} versus cycle rate for device operation at extended frequencies. I_{DD3} (max) curve is defined by the equation:

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

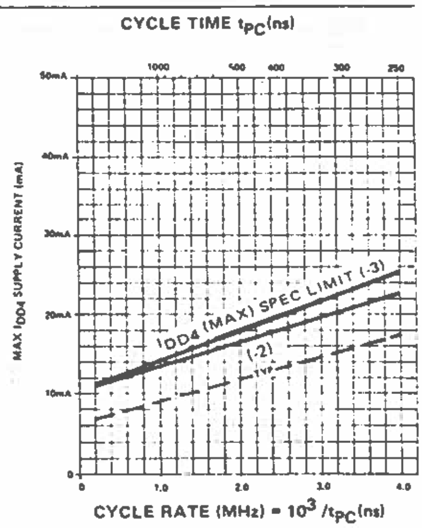
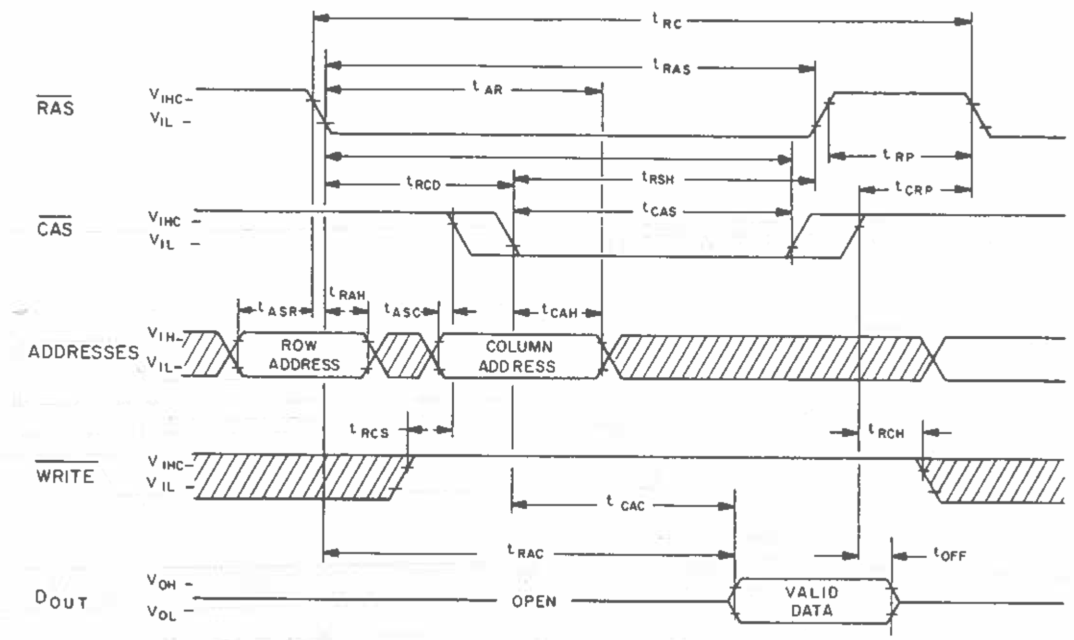


Fig. 4 Maximum I_{DD4} versus cycle rate for device operation in page mode. I_{DD4} (max) curve is defined by the equation:

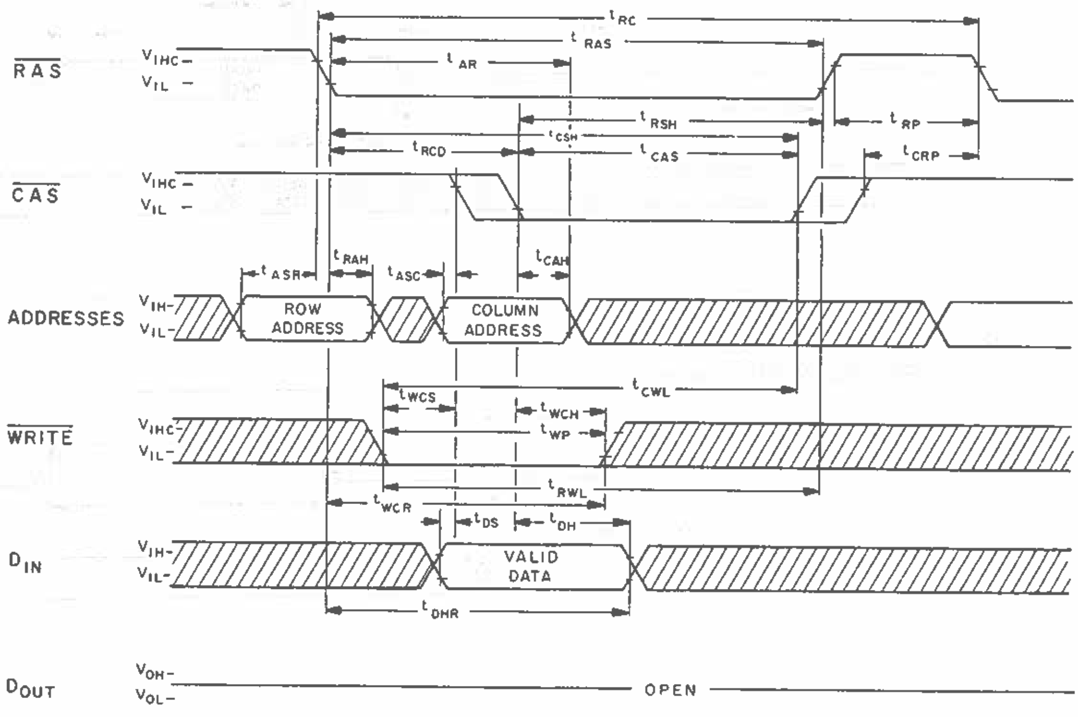
I_{DD4} (max) mA = 10 + 3.75 x cycle rate [MHz] for -3
I_{DD4} (max) mA = 10 + 3.2 x cycle rate [MHz] for -2

READ CYCLE

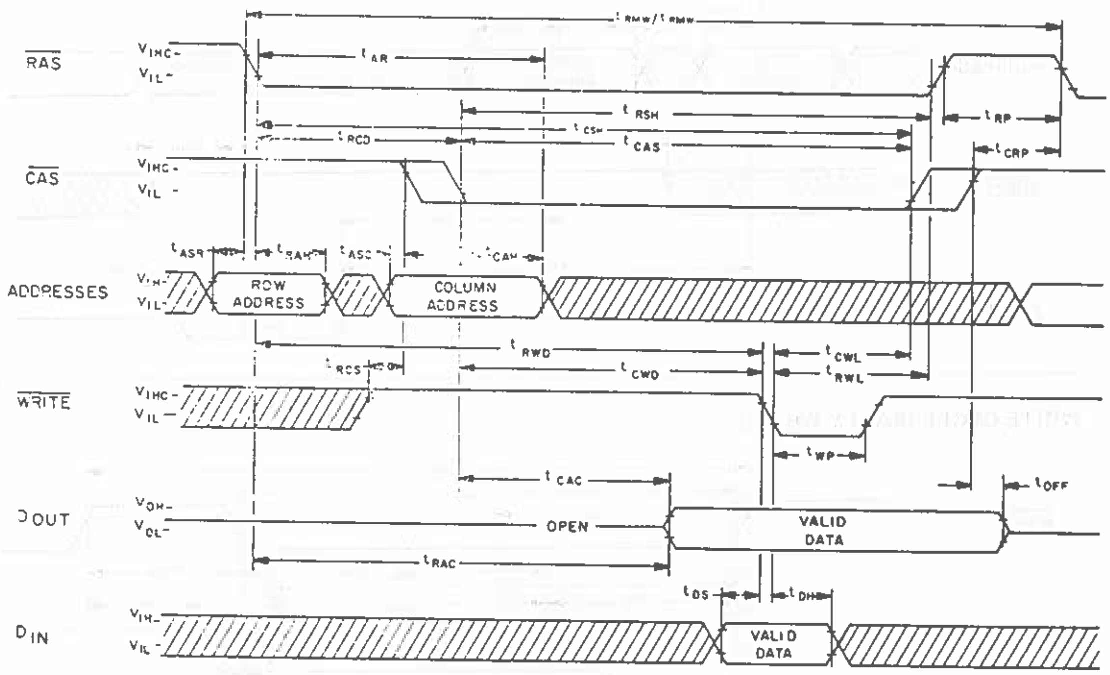


16.384x1-BIT
DYN RAM
MK4116(P/N)-2,3

WRITE CYCLE (EARLY WRITE)

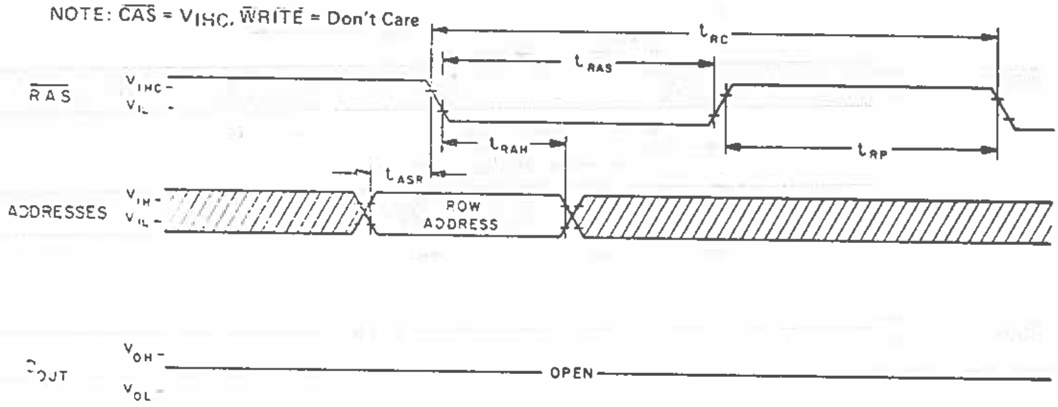


READ–WRITE/READ–MODIFY–WRITE CYCLE

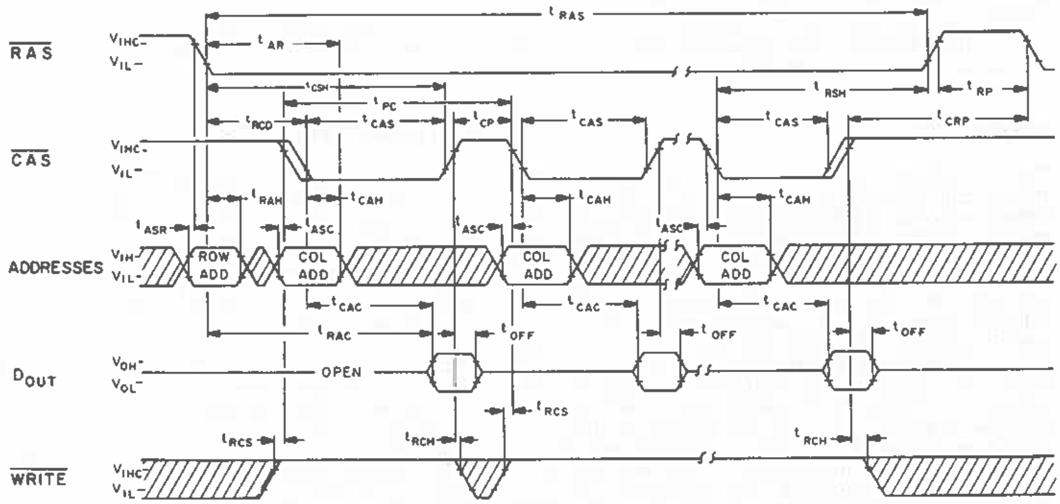


"RAS–ONLY" REFRESH CYCLE

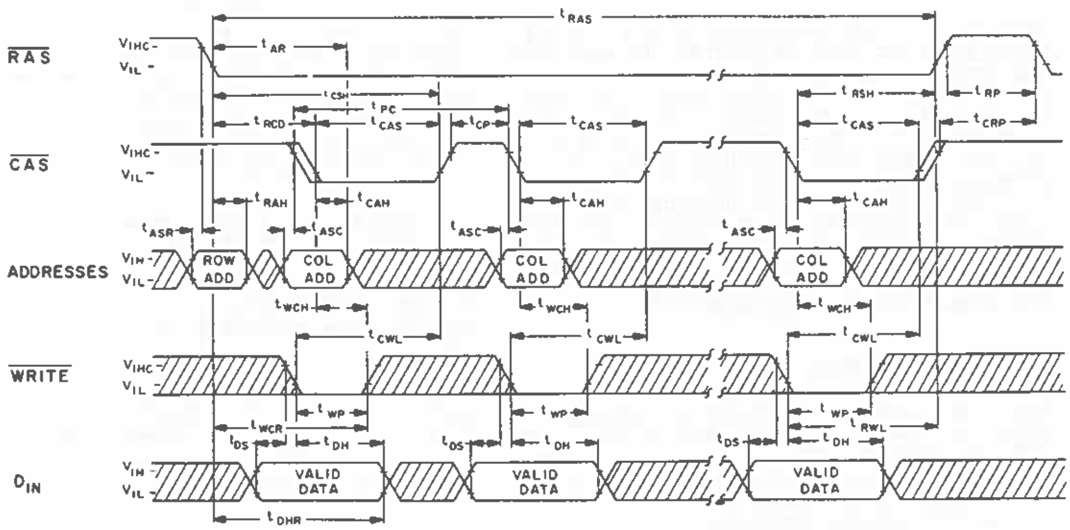
NOTE: CAS = VIHC, WRITE = Don't Care



PAGE MODE READ CYCLE



PAGE MODE WRITE CYCLE



16,384x1-BIT DYN RAM MK4116(P/N)-2/3

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 $\overline{RAS}$-only refresh. Proper control of the clock inputs($\overline{RAS}$, $\overline{CAS}$ and $\overline{WRITE}$) 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 ($\overline{RAS}$), latches the 7 row address bits into the chip. The second clock, the Column Address Strobe ($\overline{CAS}$), subsequently latches the 7 column address bits into the chip. Each of these signals, $\overline{RAS}$ and $\overline{CAS}$, 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 $\overline{CAS}$ clock sequence are inhibited until the occurence of a delayed signal derived from the $\overline{RAS}$ clock chain. This "gated $\overline{CAS}$" feature allows the $\overline{CAS}$ 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 $\overline{CAS}$ 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 $\overline{CAS}$ which are called t_{RCD} (min) and t_{RCD} (max). No data storage or reading errors will result if $\overline{CAS}$ 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 $\overline{CAS}$ (t_{CAC}) rather than from $\overline{RAS}$ (t_{RAC}), and access time from $\overline{RAS}$ 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 $\overline{WRITE}$ and $\overline{CAS}$ while $\overline{RAS}$ is active. The later of the signals ($\overline{WRITE}$ or $\overline{CAS}$) 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 $\overline{WRITE}$ input is brought low (active) prior to $\overline{CAS}$, the D_{IN} is strobed by $\overline{CAS}$, and the set-up and hold times are referenced to $\overline{CAS}$. If the input data is not available at $\overline{CAS}$ time or if it is desired that the cycle be a read-write cycle, the $\overline{WRITE}$ signal will be delayed until after $\overline{CAS}$ 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 $\overline{WRITE}$ rather than $\overline{CAS}$. (To illustrate this feature, D_{IN} is referenced to $\overline{WRITE}$ in the timing diagrams depicting the read-write and page-mode write cycles while the "early write" cycle diagram shows D_{IN} referenced to $\overline{CAS}$).

Data is retrieved from the memory in a read cycle by maintaining $\overline{WRITE}$ in the inactive or high state throughout the portion of the memory cycle in which $\overline{CAS}$ 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 $\overline{CAS}$ 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 $\overline{CAS}$ 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 $\overline{CAS}$ is taken to the precharge (logic 1) state, whether or not $\overline{RAS}$ goes into precharge.

If the cycle in progress is an "early-write" cycle ($\overline{WRITE}$ active before $\overline{CAS}$ 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 $\overline{WRITE}$ 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 $\overline{CAS}$ 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 $\overline{RAS}/\overline{CAS}$ 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

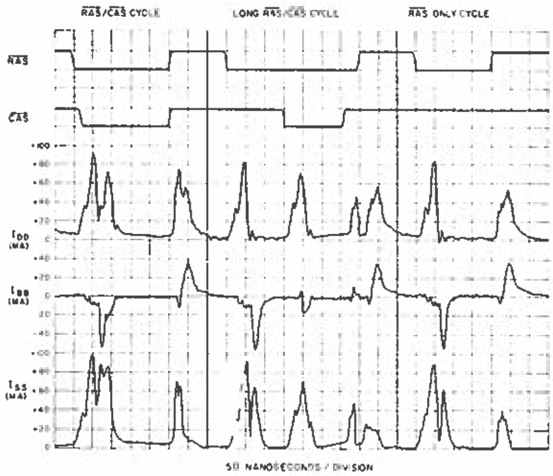


Fig. 5 Typical Current Waveforms

16,384x1-BIT DYN RAM MK4116(P/N)-2/3

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 $\overline{CAS}$.

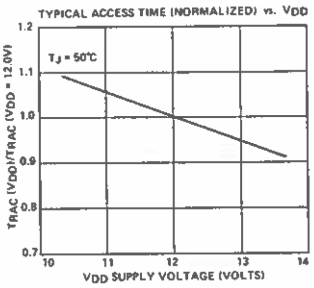
POWER UP

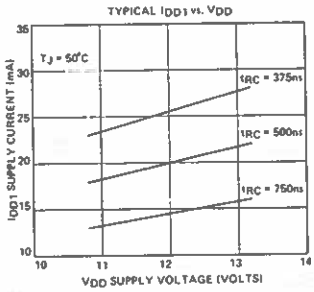
The MK 4116 requires no particular power supply sequencing so long as the Absolute Maximum Rating Conditions are observed. However, in order to insure compliance with the Absolute Maximum Ratings, MOSTEK recommends sequencing of power supplies such that V_{BB} is applied first and removed last. V_{BB}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

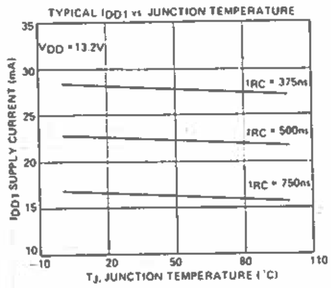
Under system failure conditions in which one or more supplies exceed the specified limits significant additional margin against catastrophic device failure may be achieved by forcing $\overline{RAS}$ and $\overline{CAS}$ to the inactive state (high level).

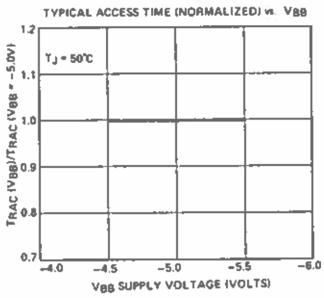
After power is applied to the device, the MK 4116 requires several cycles before proper device operation is achieved. Any 8 cycles which perform refresh are adequate for this purpose.

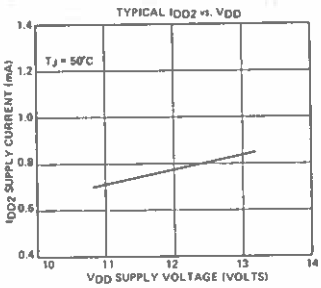
TYPICAL CHARACTERISTICS

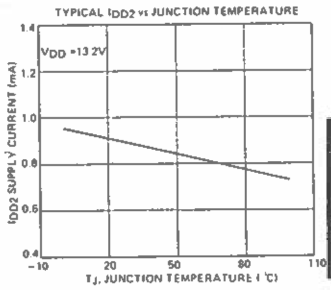




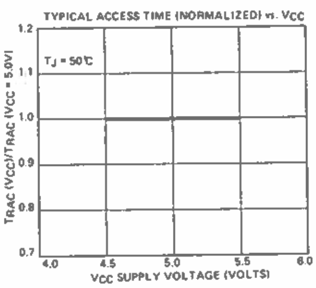


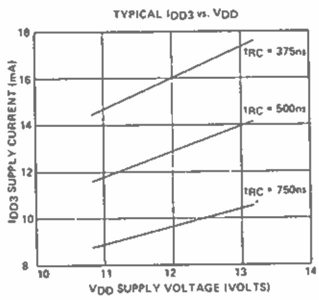


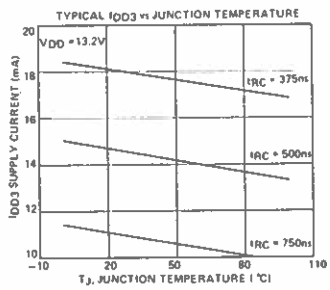


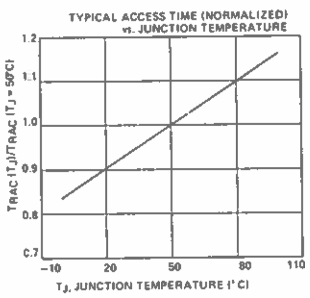


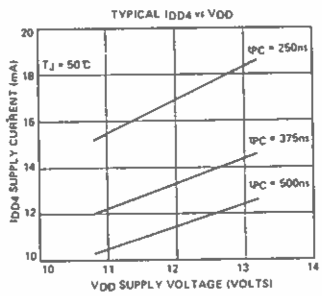
16.384x1-BIT
DYN RAM
MK4116/P N/2 3

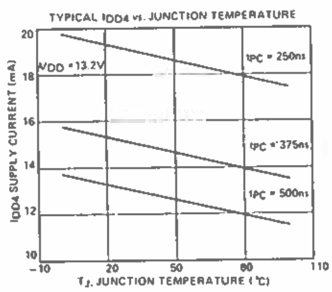




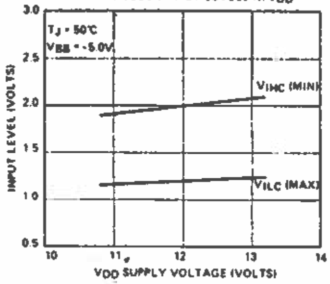




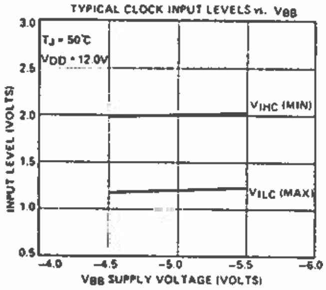




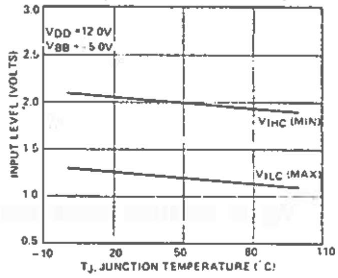
TYPICAL CLOCK INPUT LEVELS vs. VDD



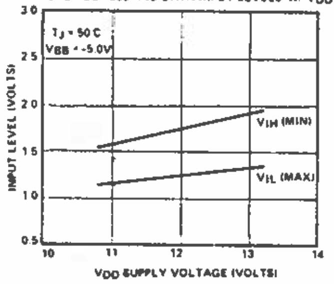
TYPICAL CLOCK INPUT LEVELS vs. VBB



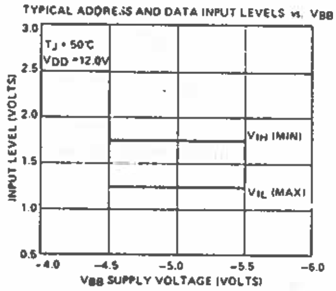
TYPICAL CLOCK INPUT LEVELS vs. TJ



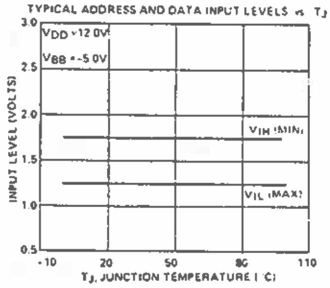
TYPICAL ADDRESS AND DATA INPUT LEVELS vs. VDD



TYPICAL ADDRESS AND DATA INPUT LEVELS vs. VBB



TYPICAL ADDRESS AND DATA INPUT LEVELS vs. TJ



Fiche technique :

EPROM 2716

National Semiconductor

MOS EPROMs

MM2716 16,384-Bit (2048 × 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.

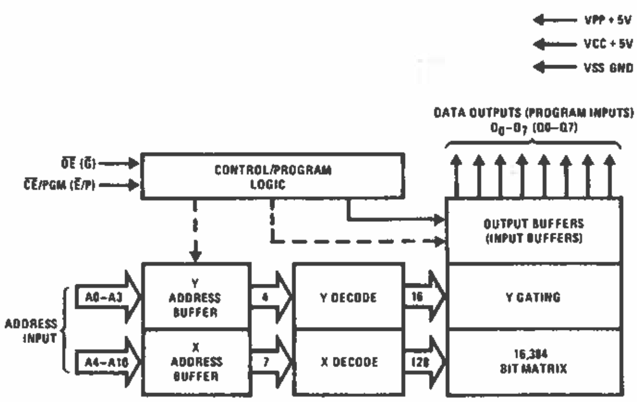
The MM2716 is packaged in a 24-pin dual-in-line package with transparent lid. The transparent lid allows the user to expose the chip to ultraviolet light to erase the bit pattern. A new pattern can then be written into the device by following the programming procedure.

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

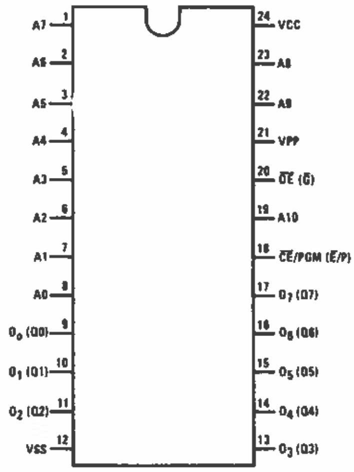
Features

- 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



TOP VIEW

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

Pin Connection During Read or Program

MODE	PIN NAME/NUMBER				
	$\overline{CE}$/PGM ($\overline{E}$/P) 18	$\overline{OE}$ ($\overline{G}$) 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

*Symbols in parentheses are proposed industry standard

Pin Names

A0–A10	Address Inputs
O_0–O_7 (Q0–Q7)	Data Outputs
$\overline{CE}$/PGM ($\overline{E}$/P)	Chip Enable/Program
$\overline{OE}$ ($\overline{G}$)	Output Enable
VPP	Read 5V, Program 25V
VCC	Power (5V)
VSS	Ground

Absolute Maximum Ratings (Note 1)

Temperature Under Bias	−25°C to +85°C
Storage Temperature	−65°C to +125°C
VPP Supply Voltage with Respect to VSS	26.5V to −0.3V
All Input or Output Voltages with Respect to VSS (except VPP)	6V to −0.3V
Power Dissipation	1.5 W
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		VCC + 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}	TAXOX	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.

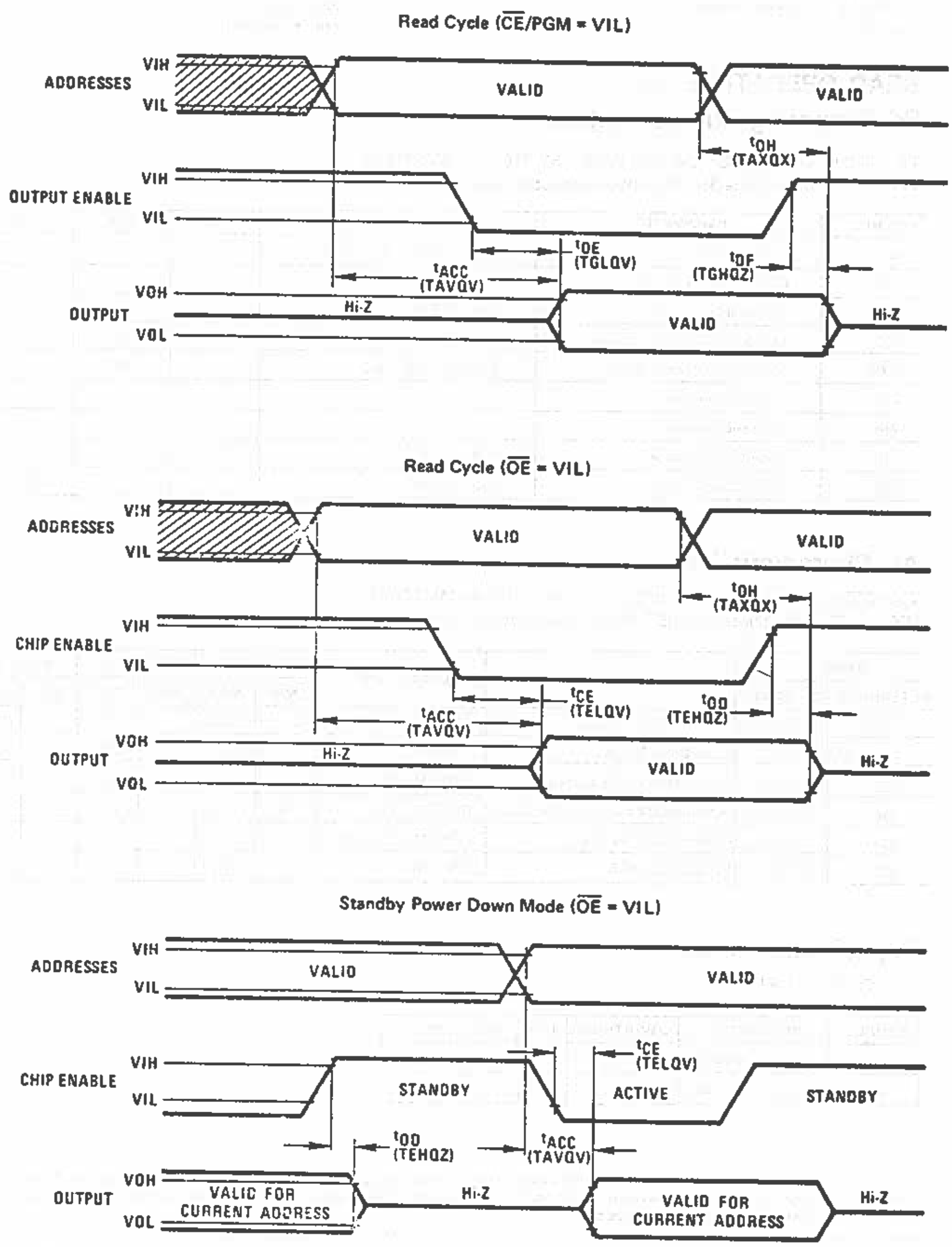
Note 2: Typical conditions are for operation at: T_A = 25°C, VCC = 5V, VPP = VCC, and VSS = 0V.

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 *



*Symbols in parentheses are proposed industry standard

MM2716

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					
t_{AS}	TAVPH	Address Setup Time	2			μs
t_{OS}	TGHPH	$\overline{OE}$ Setup Time	2			μs
t_{DS}	TDVPH	Data Setup Time	2			μs
t_{AH}	TPLAX	Address Hold Time	2			μs
t_{OH}	TPLGX	$\overline{OE}$ Hold Time	2			μs
t_{DH}	TPLDX	Data Hold Time	2			μs
t_{DF}	TGHQZ	Chip Disable to Output Float Delay (Note 4)	0		100	ns
t_{CE}	TGLQV	Chip Enable to Output Delay (Note 4)			120	ns
t_{PW}	TPHPL	Program Pulse Width	45	50	55	ms
t_{PR}	TPH1PH2	Program Pulse Rise Time	5			ns
t_{PF}	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 ≤ VIN ≤ 5.25V.

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

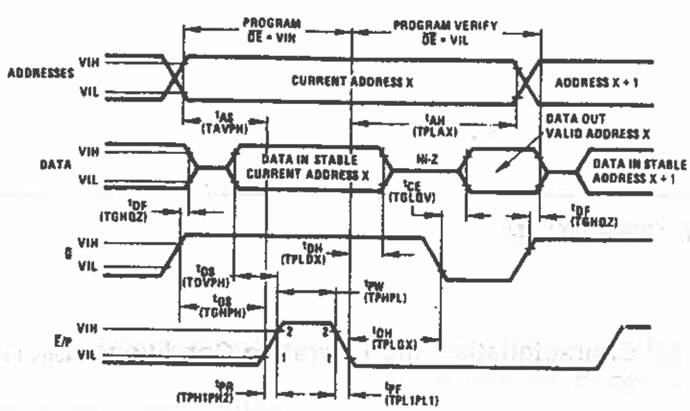
Note 5: VPP = 26V.

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

5

MM2716

Timing Diagram *



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 Q 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 μW/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

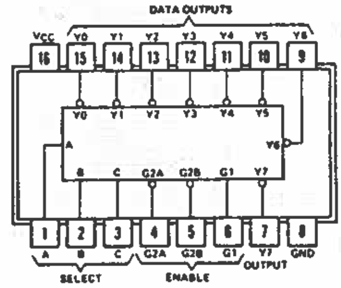
TYPES SN54LS138, SN54LS139, SN54S138, SN54S139, SN74LS138, SN74LS139, SN74S138, SN74S139 DECODERS/DEMULTIPLEXERS

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**

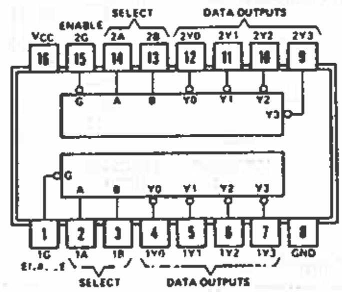
TYPE	TYPICAL PROPAGATION DELAY (3 LEVELS OF LOGIC)	TYPICAL POWER DISSIPATION
'LS138	22 ns	32 mW
'S138	8 ns	245 mW
'LS139	22 ns	34 mW
'S139	7.5 ns	300 mW

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.

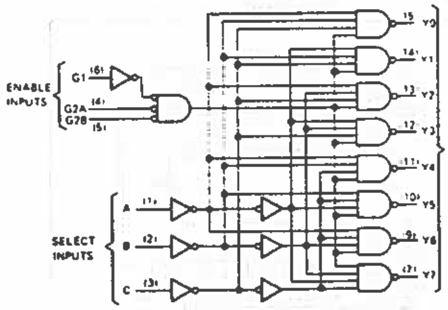
The 'LS138 and 'S138 decode one-of-eight lines dependent on the conditions at the three binary select inputs and the three enable inputs. Two active-low and one active-high enable inputs reduce the need for external gates or inverters when expanding. A 24-line decoder can be implemented without external 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

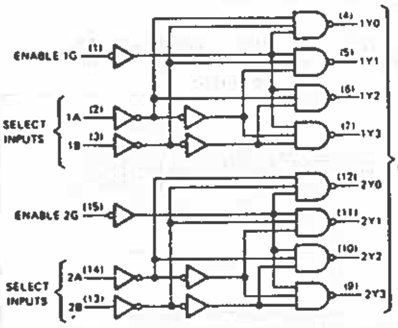


'LS138, 'S138
FUNCTION TABLE

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

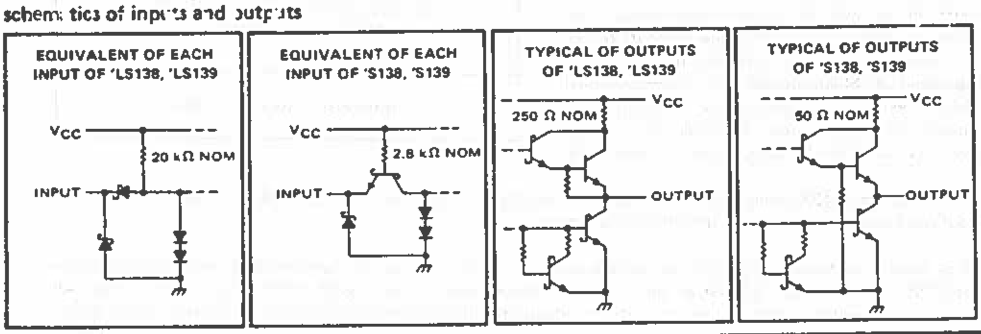


'LS139, 'S139
(EACH DECODER/DEMULTIPLEXER)
FUNCTION TABLE

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	V
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	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.

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}			2			27	41		22	33	ns
t_{PLH}			3			18	27		18	29	ns
t_{PHL}			3			26	39		25	38	ns
t_{PLH}	Enable	Any	2			12	18		16	24	ns
t_{PHL}			2			21	32		21	32	ns
t_{PLH}			3			17	26				ns
t_{PHL}			3			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.

TYPES SN54S138, SN54S139, SN74S138, SN74S139
DECODERS/DEMULTIPLEXERS

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

HEX BUS DRIVERS WITH 3-STATE OUTPUTS

recommended operating conditions

	54 FAMILY / 74 FAMILY	SERIES 54 / SERIES 74 '365A, '366A '367A, '368A MIN	NOM	MAX	SERIES 54LS / SERIES 74LS 'LS365, 'LS366 'LS367, 'LS368 MIN	NOM	MAX	UNIT
Supply voltage, V_{CC}	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	
High-level output current, I_{OH}	54 Family			−2			−1	mA
	74 Family			−5.2			−2.6	
Low-level output current, I_{OL}	54 Family			32			12	mA
	74 Family			32			24	
Operating free-air temperature, T_A	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 MIN	TYP‡	MAX	SERIES 54LS / SERIES 74LS 'LS365, 'LS366 'LS367, 'LS368 MIN	TYP‡	MAX	UNIT
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, V_{IL} = V_{IL} max, I_{OH} = MAX	54 Family	2.4	3.3		2.4	3.3		V
				74 Family	2.4	3.1		2.4	3.1		
V_{OL}	Low-level output voltage	2	V_{CC} = MIN, V_{IH} = 2 V, V_{IL} = V_{IL} max, I_{OL} = MAX,	54 Family			0.4		0.25	0.4	V
				74 Family			0.4		0.35	0.5	
			I_{OL} = 12mA	Series 74LS					0.25	0.4	
I_{OZ}	Off-state (high-impedance state) output current	19	V_{CC} = MAX, V_{IH} = 2 V, V_{IL} = V_{IL} max	V_O = 2.4 V			40			20	µA
				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				mA
				V_I = 7 V						0.1	
I_{IH}	High-level input current	4	V_{CC} = MAX	V_{IH} = 2.4 V			40				µA
				V_{IH} = 2.7 V						20	
I_{IL}	Low-level input current, A inputs		V_{CC} = MAX, V_I = 0.5 V, Either $\bar{G}$ input at 2 V				−40			−20	µA
			V_{CC} = MAX, V_I = 0.4 V, Both $\bar{G}$ inputs at 0.4 V				−1.6			−0.4	mA
	$\bar{G}$ 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¶

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	59	77
'LS365, 'LS367	0 V	4.5 V	14	24
'LS366, 'LS368	0 V	4.5 V	12	21

¶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.

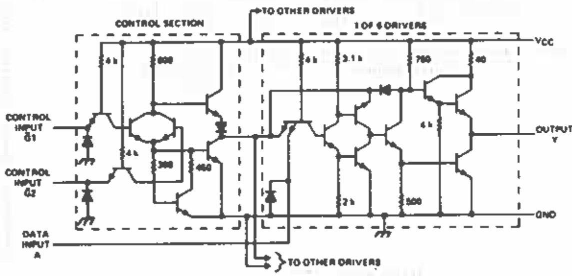
switching characteristics, V_{CC} = 5 V, T_A = 25°C, see note 1

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		19	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

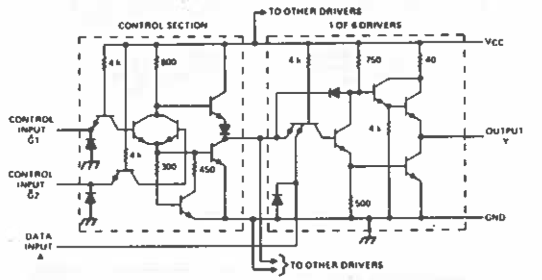
*t_{PLH} = Propagation delay time, low-to-high-level output
t_{PHL} = Propagation delay time, high-to-low-level output
t_{PZH} = Output enable time to high level
t_{PZL} = Output enable time to low level
t_{PHZ} = Output disable time from high level
t_{PLZ} = Output disable time from low level

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

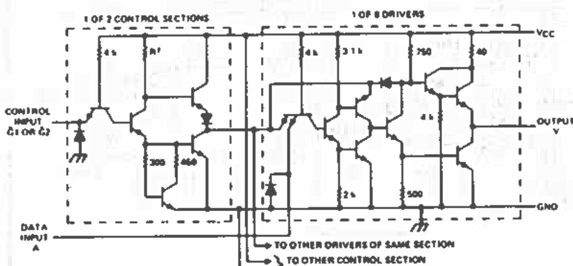
schematics



'365A CIRCUITS

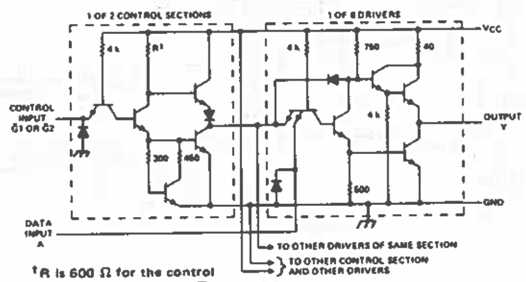


'366A CIRCUITS



†R is 600 Ω for the control section associated with $\bar{G}1$ and 900 Ω for the control section associated with $\bar{G}2$.

'367A CIRCUITS



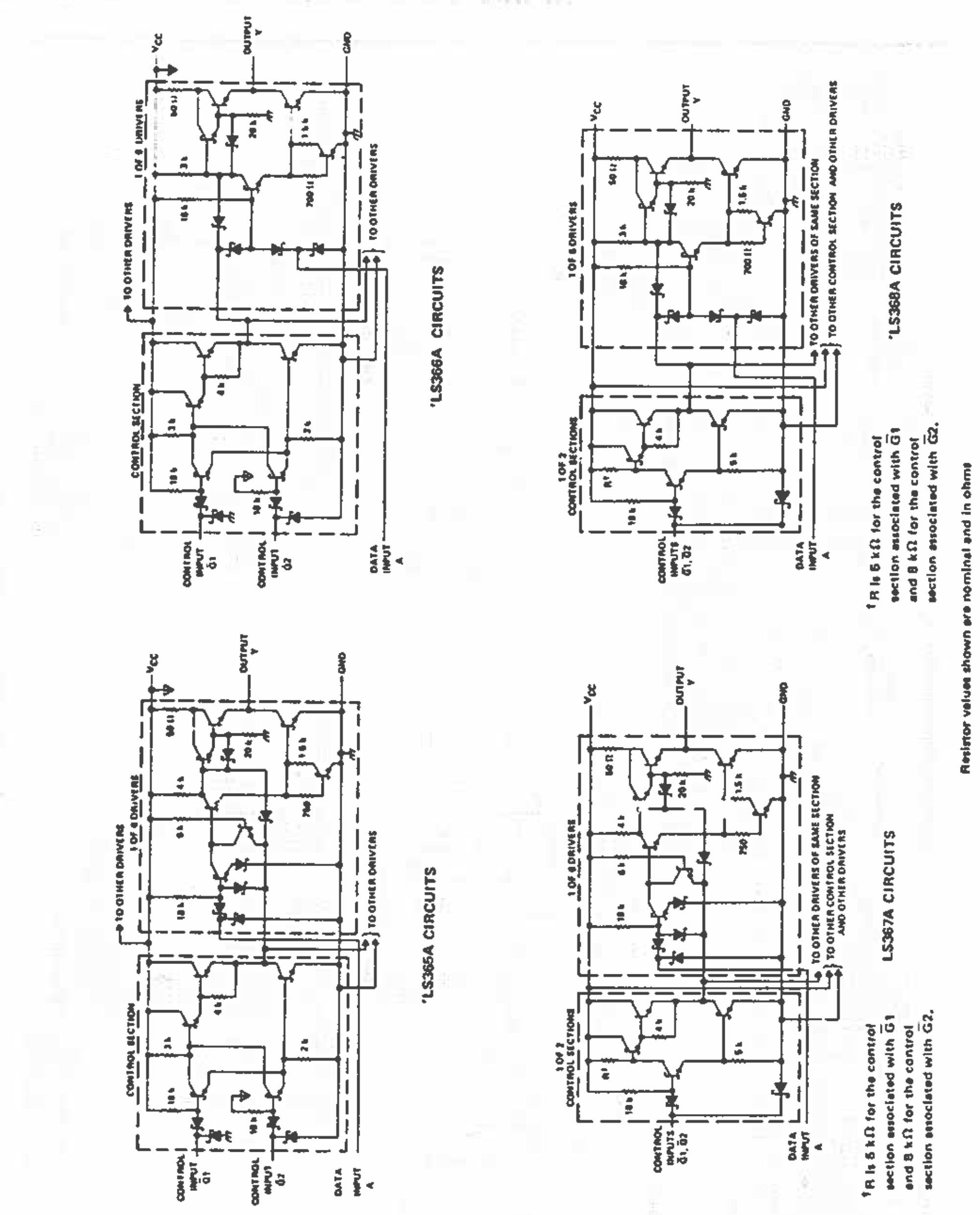
†R is 600 Ω for the control section associated with $\bar{G}1$ and 900 Ω for the control section associated with $\bar{G}2$.

'368A CIRCUITS

Resistor values shown are nominal and in ohms.

6

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

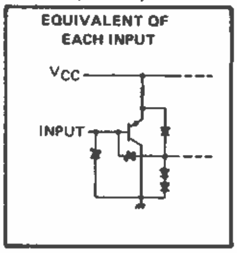
- 3-State Outputs Drive Bus Lines or Buffer Memory Address Registers
- P-N-P Inputs Reduce D-C Loading
- Hysteresis at Inputs Improves Noise Margins

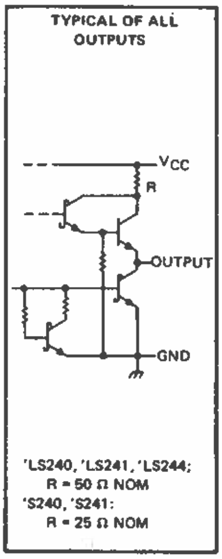
description

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 G̅ (active-low output control) inputs, and complementary G and G̅ 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.

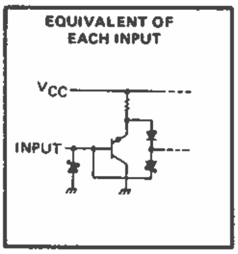
schematics of inputs and outputs

'LS240, 'LS241, 'LS244

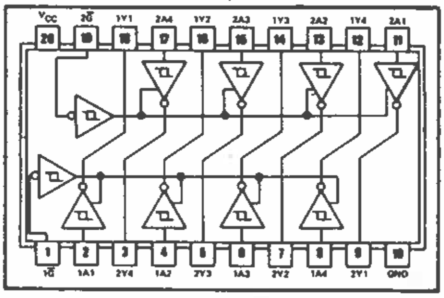




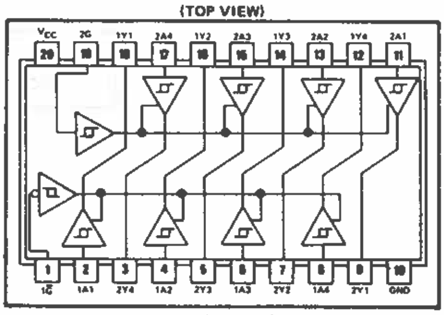
'S240 'S241



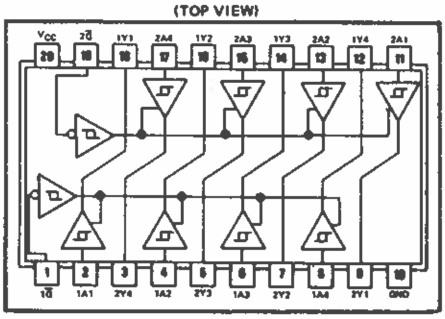
SN54LS240, SN54S240 . . . J
SN74LS240, SN74S240 . . . J OR N
(TOP VIEW)



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



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



6

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

recommended operating conditions

PARAMETER	SN54LS' MIN	SN54LS' NOM	SN54LS' MAX	SN74LS' MIN	SN74LS' NOM	SN74LS' MAX	UNIT
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' MIN	SN54LS' TYP‡	SN54LS' MAX	SN74LS' MIN	SN74LS' TYP‡	SN74LS' MAX	UNIT
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
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	V
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	µA
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, Outputs open	All		13	23		13	23	mA
	Outputs low		'LS240		26	44		26	44	mA
			'LS241, 'LS244		27	46		27	46	mA
	All outputs disabled		'LS240		29	50		29	50	mA
			'LS241, 'LS244		32	54		32	54	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.
♦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 MIN	'LS240 TYP	'LS240 MAX	'LS241, 'LS244 MIN	'LS241, 'LS244 TYP	'LS241, 'LS244 MAX	UNIT
t_{PLH}	Propagation delay time, low-to-high-level output	C_L = 45 pF, R_L = 667 Ω, See Note 2		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, R_L = 667 Ω, See Note 2		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

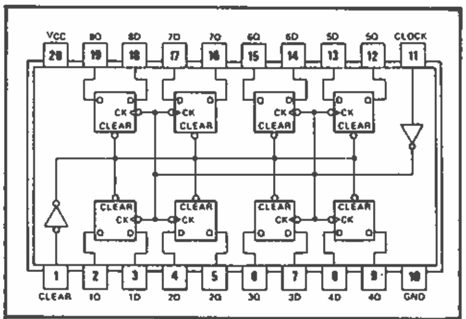
TTL
MSI

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)

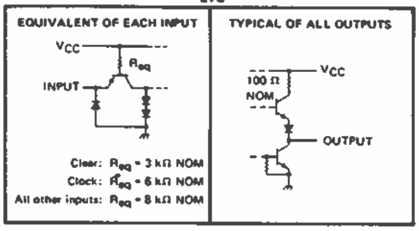
INPUTS			OUTPUT
CLEAR	CLOCK	D	Q
L	X	X	L
H	↑	H	H
H	↑	L	L
H	L	X	Q_0

See explanation of function tables on page 3-8.

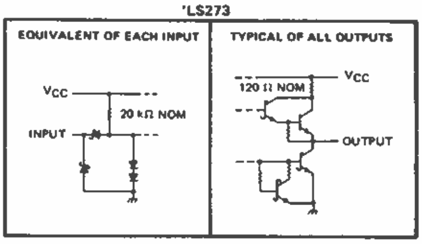
7

schematics of inputs and output

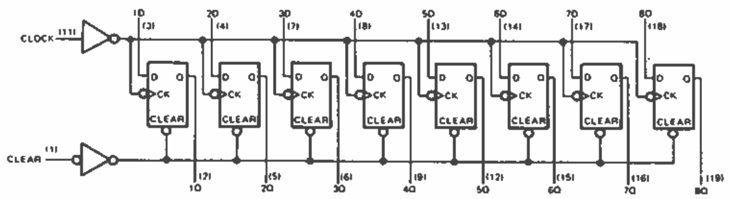
'273



'LS273



functional block diagram



Fiche technique :

6845

MOTOROLA

MC6845

Advance Information

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.

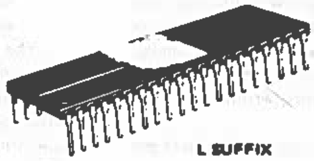
The CRTC is optimized for hardware/software balance in order to achieve integration of all key functions and maintain flexibility. For instance, all keyboard functions, R/W, cursor movements, and editing are under processor control; whereas the CRTC provides video timing and Refresh Memory Addressing.

- 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.

MOS

(N-Channel, Silicon-Gate)

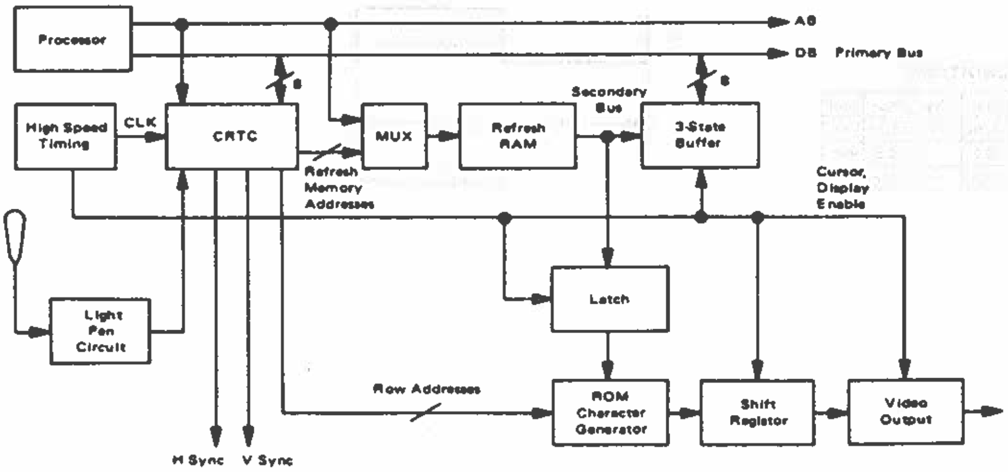
CRT CONTROLLER (CRTC)



L SUFFIX
CERAMIC PACKAGE
CASE 715

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 provids 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

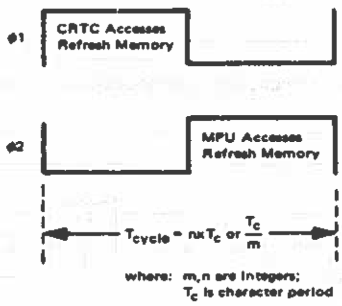
Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}*	-0.3 to +7.0	Vdc
Input Voltage	V_{in}*	-0.3 to +7.0	Vdc
Operating Temperature Range	T_A	0 to +70	°C
Storage Temperature Range	T_{stg}	-55 to +150	°C

*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		V_{OH}				Vdc
(I_{load} = -205 μA)	D0-D7		2.4	–	–	
(I_{load} = -100 μA)	Other Outputs		2.4	–	–	
Output Low Voltage (I_{load} = 1.6 mA)		V_{OL}	–	–	0.4	Vdc
Power Dissipation		P_D	–	600	–	mW
Input Capacitance		C_{in}				pF
	D0-D7		–	–	12.5	
	All others		–	–	10	
Output Capacitance	All Outputs	C_{out}	–	–	10	pF
Minimum Clock Pulse Width, Low		PW_{CL}	160	–	–	ns
Minimum Clock Pulse Width, High		PW_{CH}	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, $\overline{CS}$ 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

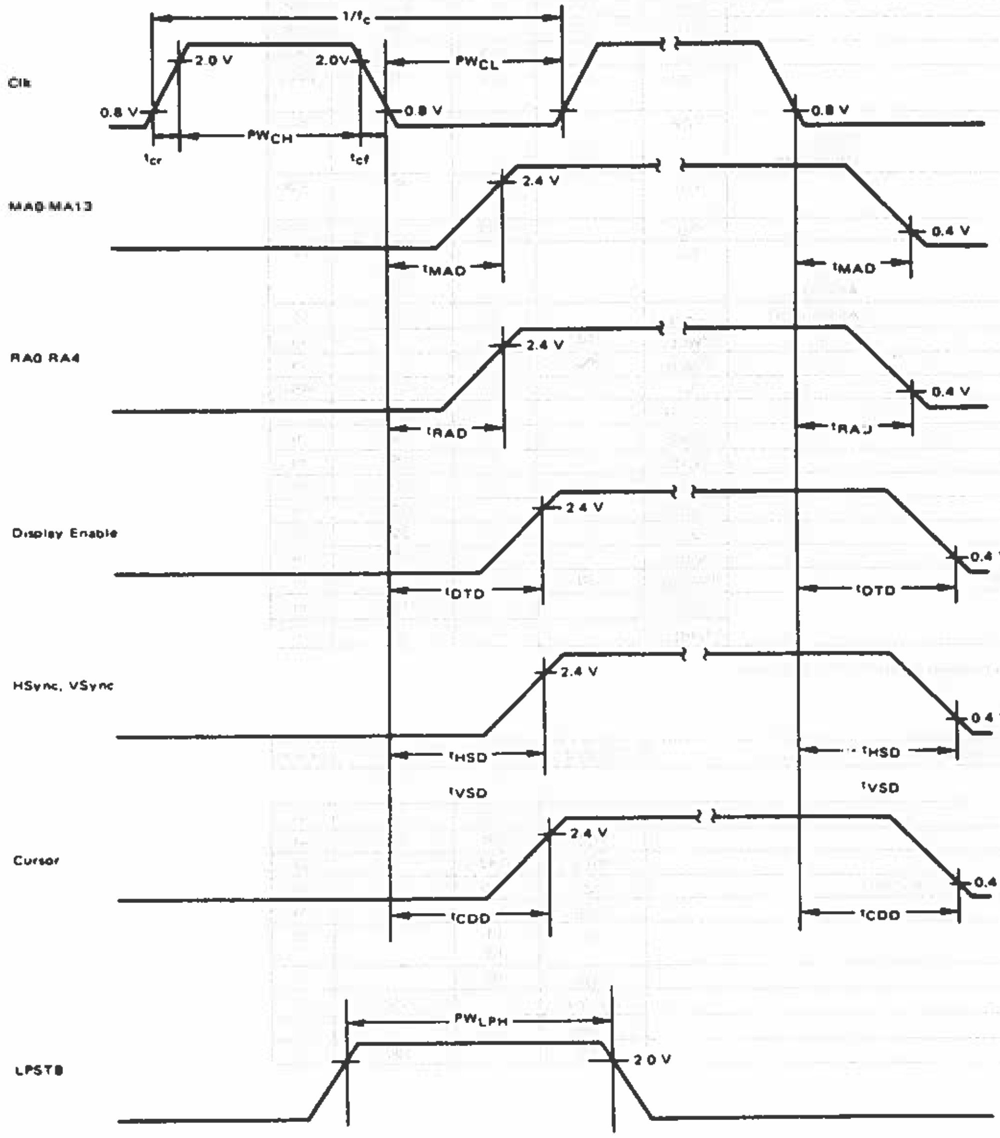
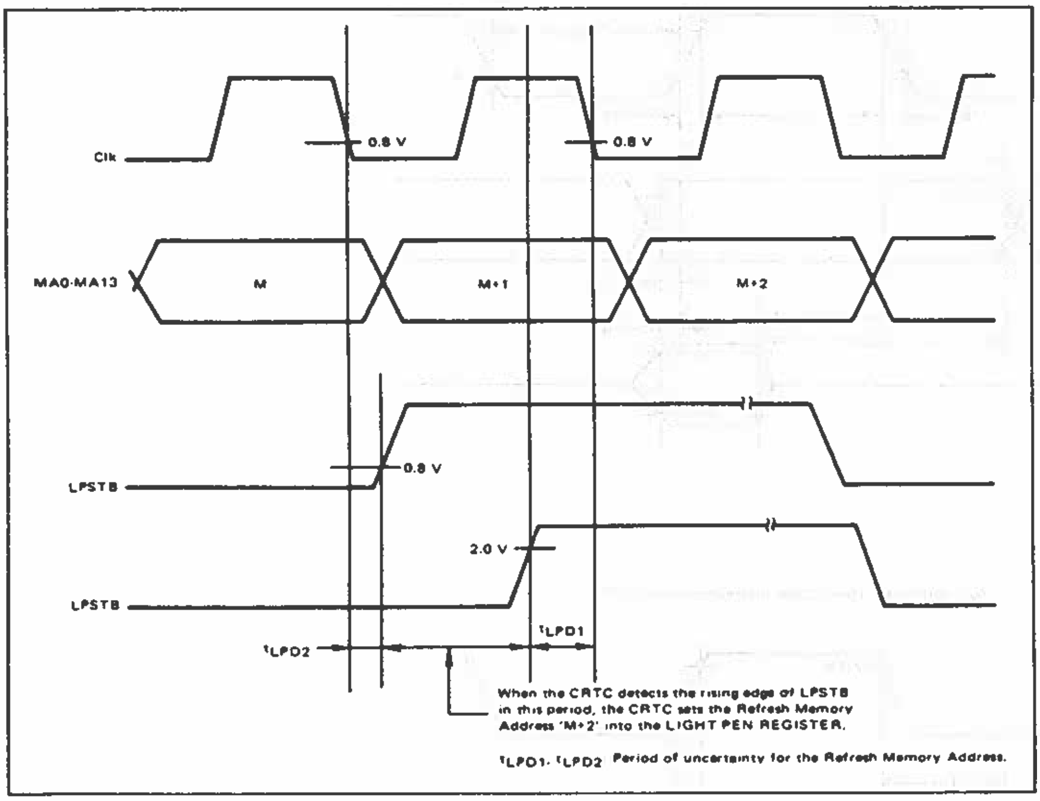


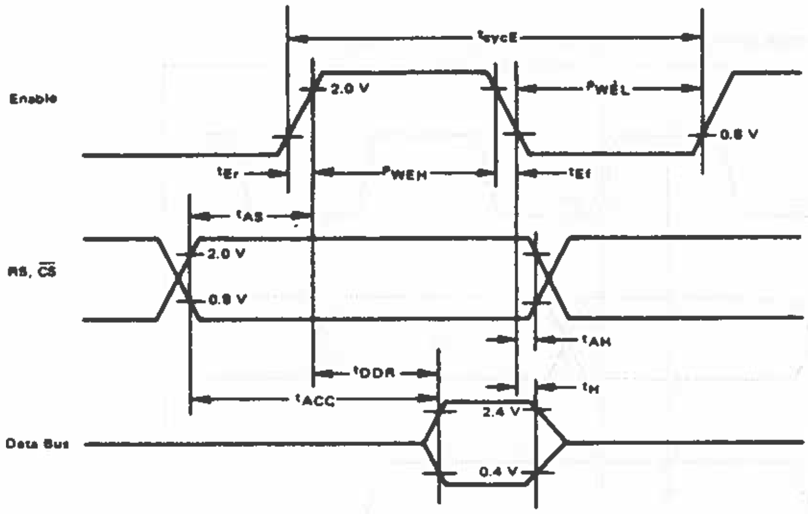
FIGURE 4 — RELATION BETWEEN LPSTB AND REFRESH MEMORY ADDRESS



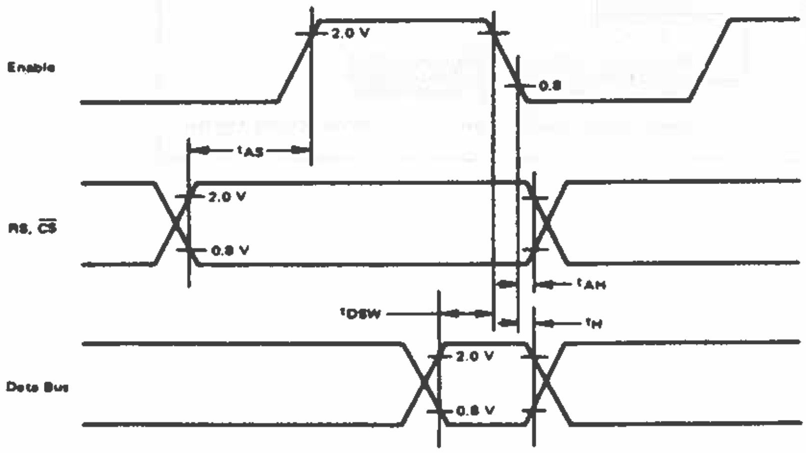
MC6845

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

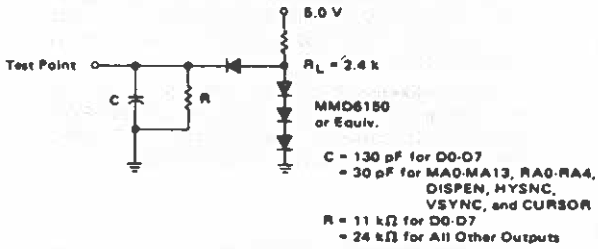
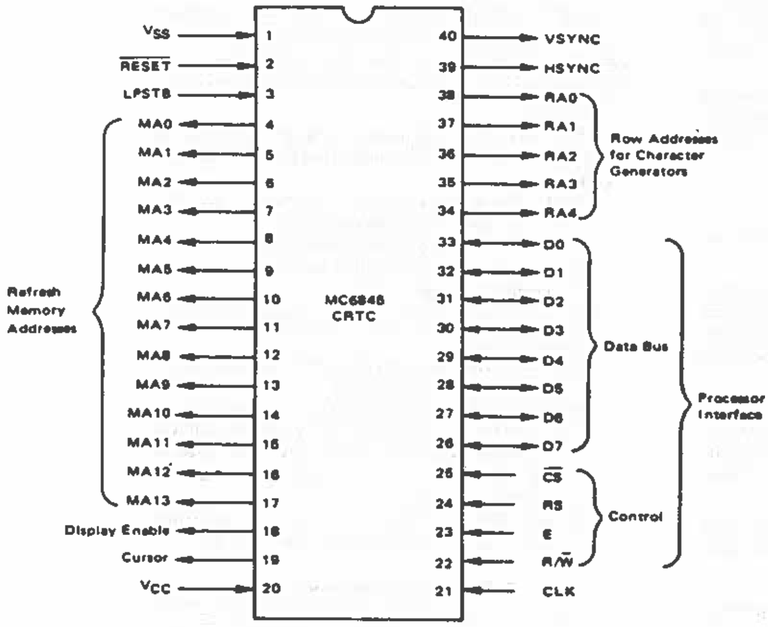


FIGURE 7 – PIN ASSIGNMENT



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:

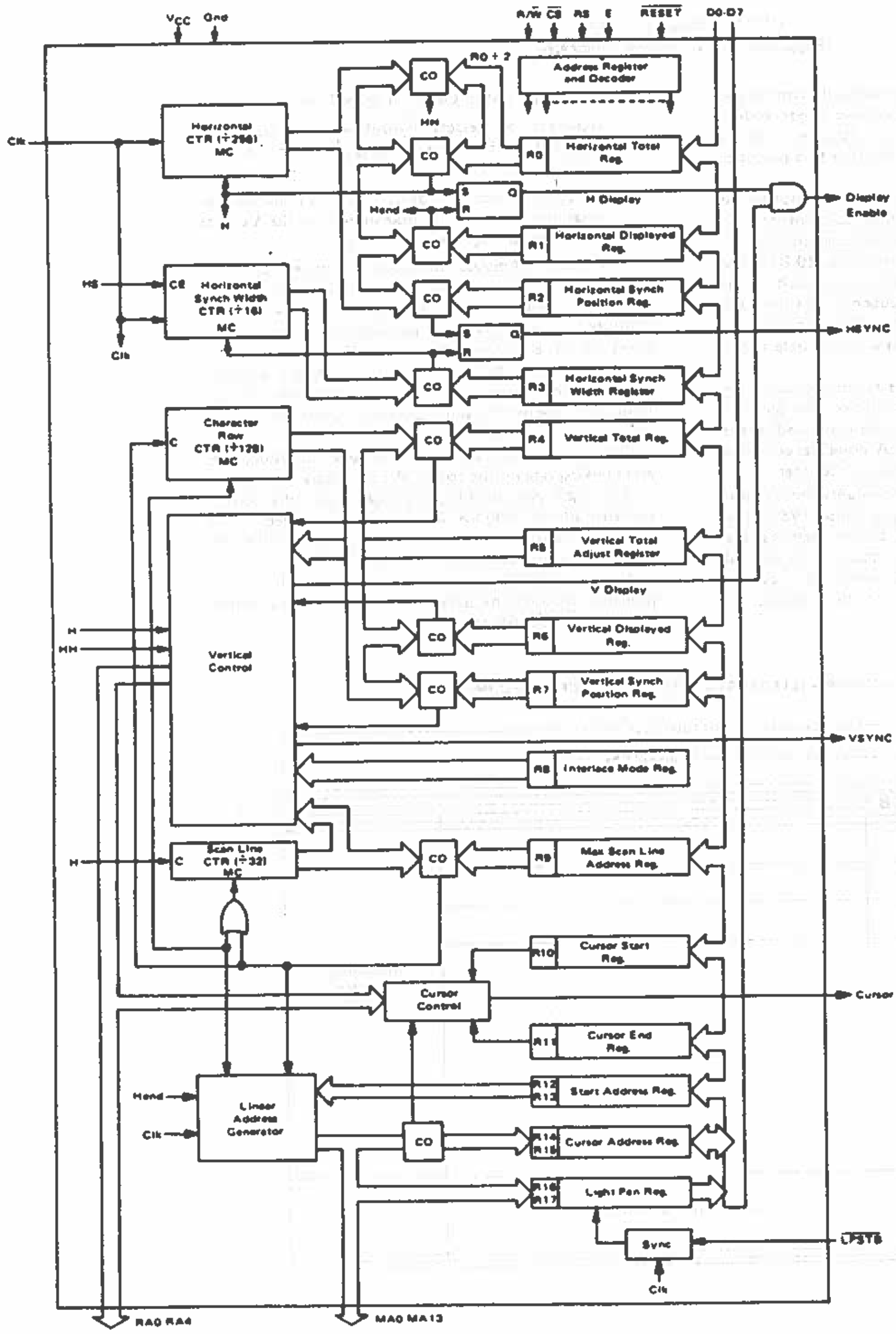
(A) $\overline{RES}$ signal has capability of reset function only when LPSTB is at low level.
(B) After $\overline{RES}$ has gone down to low level, output signals of MA0-MA13 and RA0-RA4, synchronizing with CLK low level, goes down to low level. (At least 1 cycle CLK signal is necessary for reset.)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

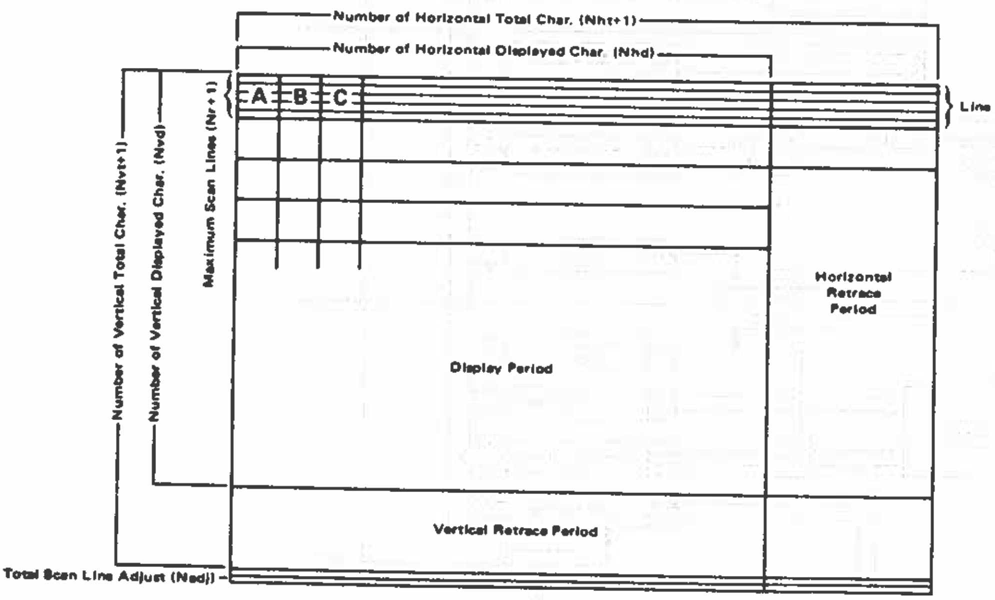
The Linear Address Generator is driven by CLK and locates the relative positions of characters in memory with their positions on the screen. Fourteen lines, MA0-MA13, are available for addressing up to four pages of 4K characters, 8 pages of 2K characters, etc. Using the Start Address Register, hardware 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



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 postiion 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.

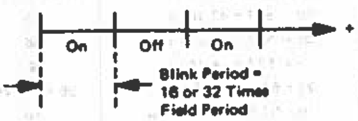
Hardware Scrolling — Registers R12/R13 contents determine which memory location is the first displayed character on the screen. Since the CRTC Linear Address Generator counts from this beginning count, the displayed portion of the screen may be a window on any continuous string of characters within a 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

Bit 6 (B)	Bit 5 (P)	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

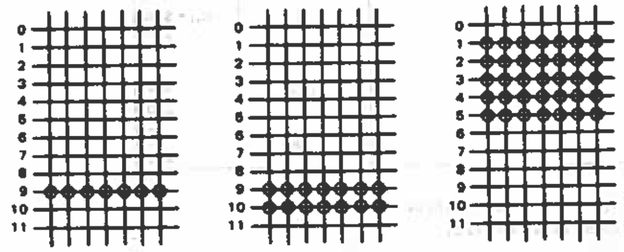


FIGURE 11 – INTERFACE CONTROL

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

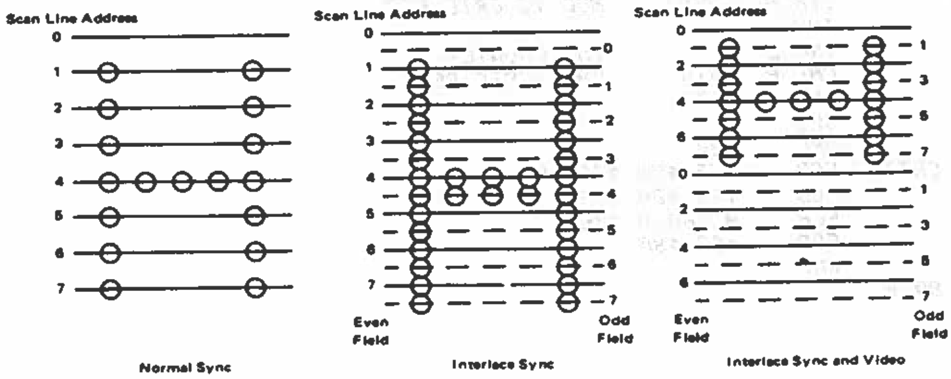


TABLE 3 – Typical 80 x 24 Screen Format Initialization of CRTC

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)	–			\$00
R13	Start Address (L)	–		128	\$80
R14	Cursor (H)	–			\$00
R15	Cursor (L)	–		128	\$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 CRTI1      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   CRTI1
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
CRTI1  0004 CRTTAB 0020

TOTAL ERRORS 00000
```

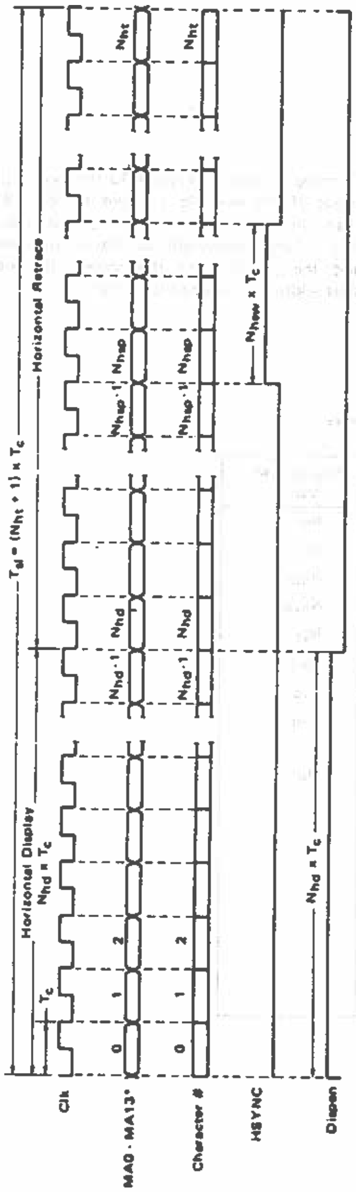
OPERATION OF THE CRTC

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)		

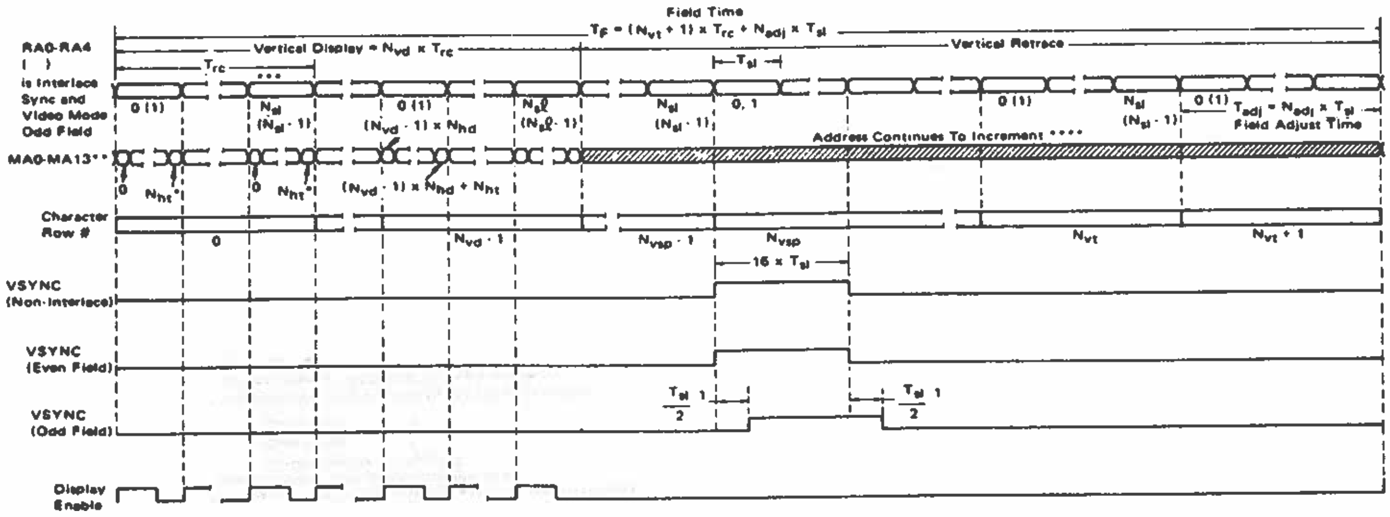
FIGURE 13 – CRTC HORIZONTAL TIMING



*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.

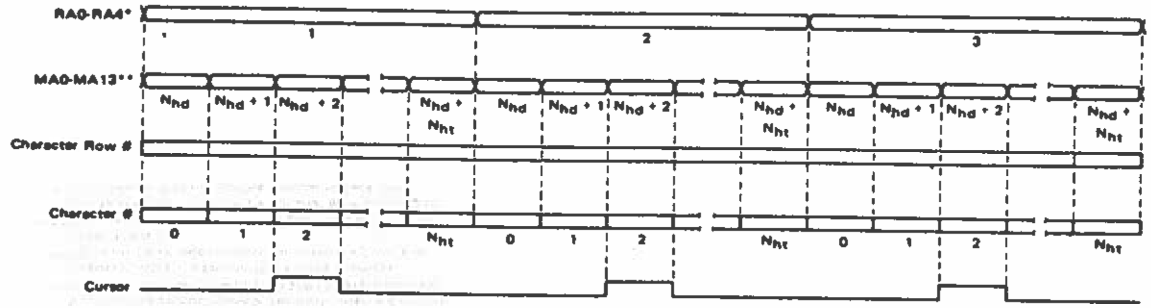
MC6845

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



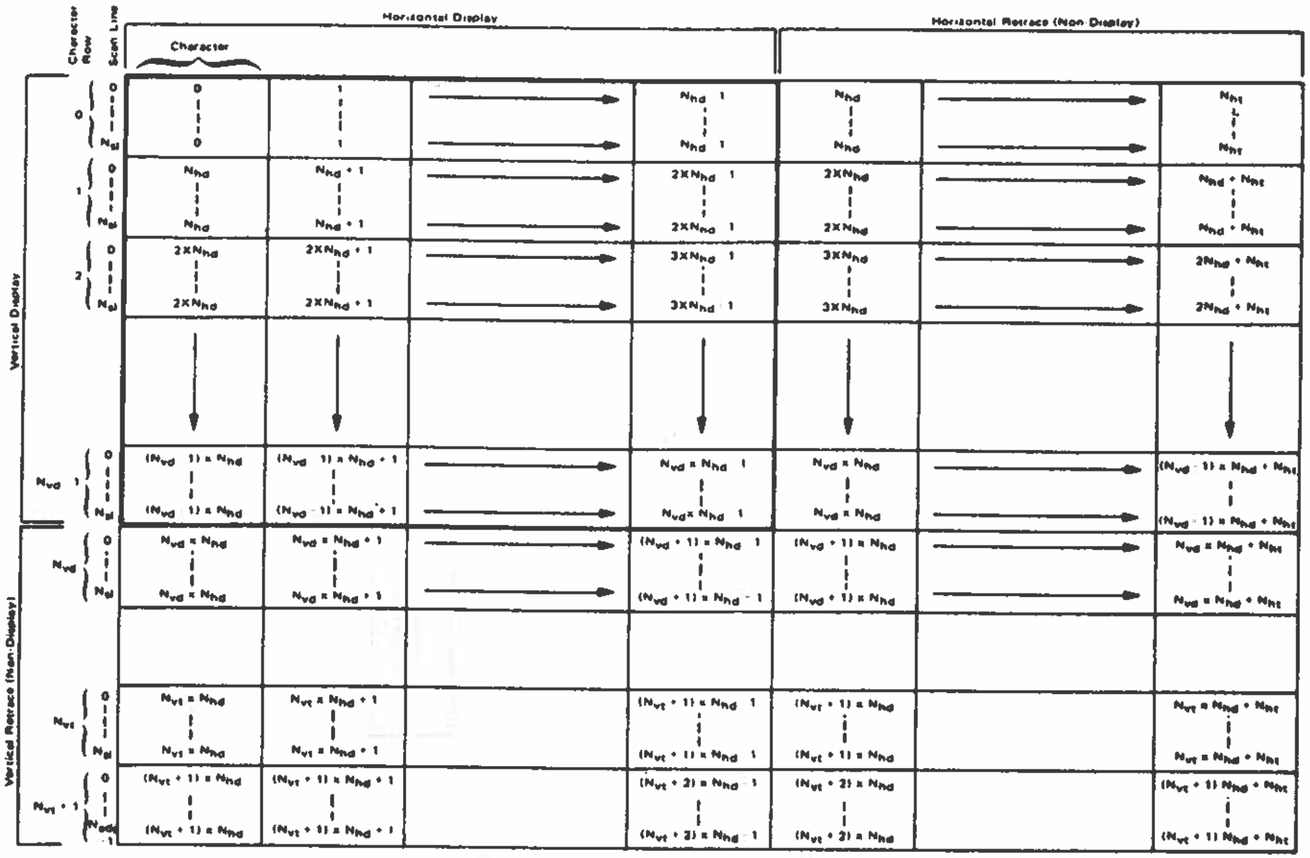
*Timing is shown for non-interlace and interlace sync modes.
Example shown has cursor programmed as:

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

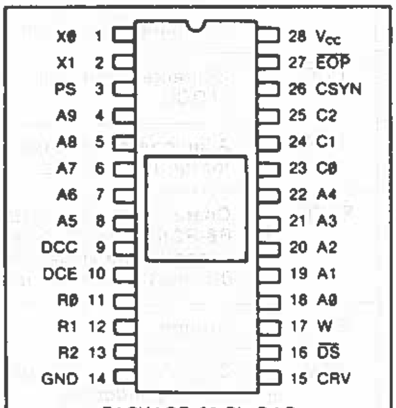
Preliminary Specifications

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

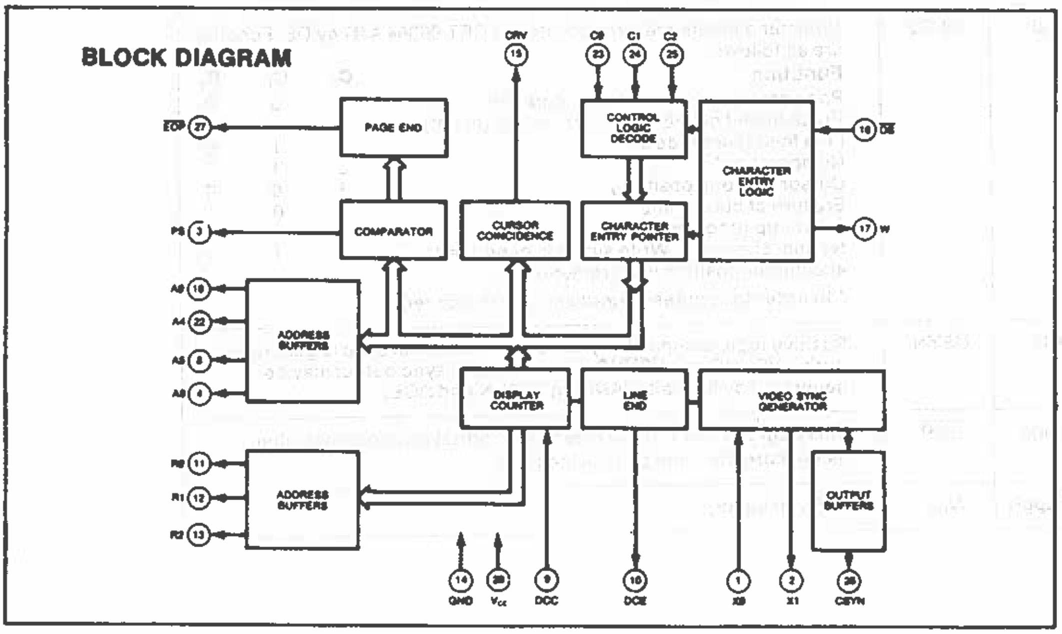
The CRT 96364 A/B is a CRT Controller which controls all of the functions associated with a 16 line x 64 character video display. Functions include CRT refresh, character entry, and cursor management.

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 96364 B). For 50 Hz line operation, the CRT 96364 A 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 96364 A/B.
17	Write	W	A positive going signal which indicates that the CRT 96364 A/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 96364 A/B by $\overline{DS}$. Functions are as follows: (see table below)
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.

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

SECTION IV

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.

PRELIMINARY

ELECTRICAL CHARACTERISTICS (T_A = 0°C to 70°C, Vcc = −5V ± 5%, unless otherwise noted)

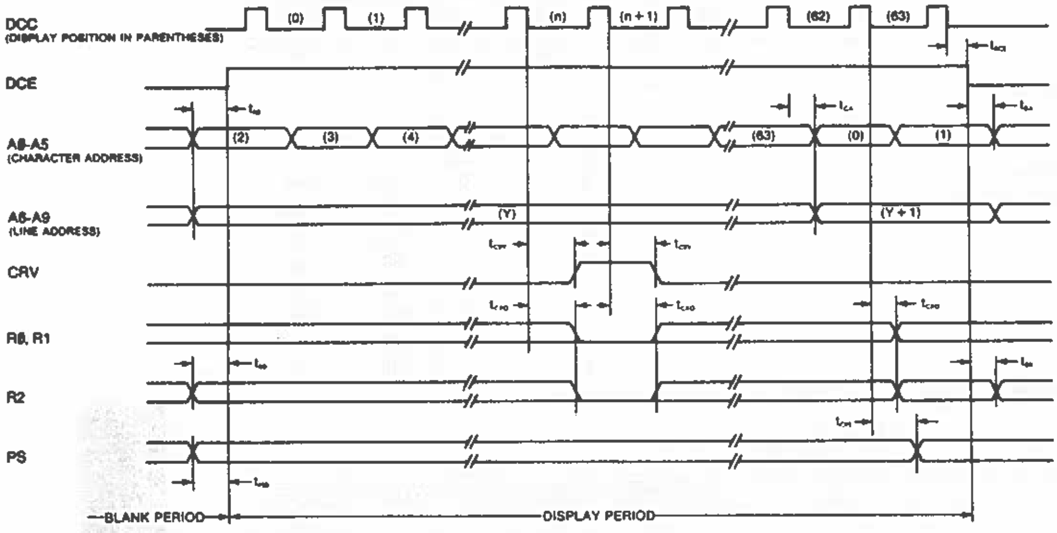
Parameter	Min.	Typ.	Max.	Unit	Comments
D.C. CHARACTERISTICS					
INPUT VOLTAGE LEVELS (except DCC)					
Low-level, V_{IL}			0.65	V	excluding DCC
High-level, V_{IH}	2.2			V	excluding DCC
INPUT VOLTAGE LEVELS—DCC					
Low-level, V_{IL}			0.65	V	
High-level, V_{IH}	3.5			V	
OUTPUT VOLTAGE LEVELS (DCE Only)					
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	

AC CHARACTERISTICS

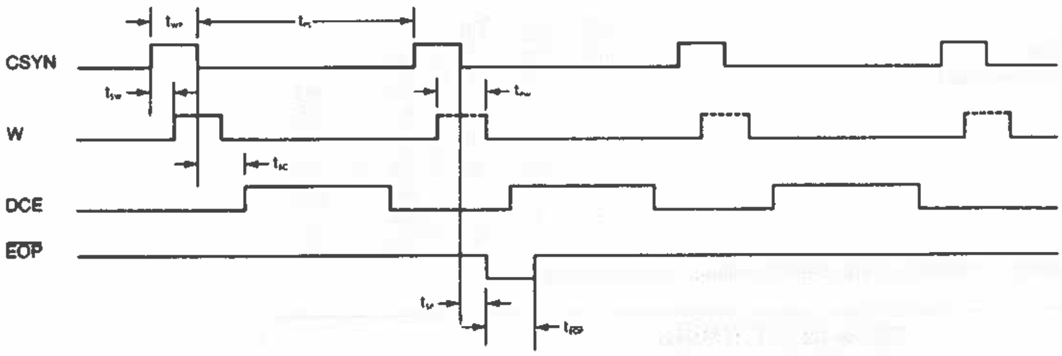
PARAMETERS	SYMBOL	VALUES MIN.	TYP.	MAX.	UNIT
Frequency of control clock DCC	f_{DCC}		1.6		MHz
Crystal Frequency CRT 96364 A CRT 96364 B	f_X f_X		1.008 1.018		MHz MHz
DCC pulse width	t_{DCC}	200			ns
Rise and fall times	t_r t_f		20	40	ns
Refresh memory address access time	t_{CA}		200	250	ns
Character 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
$\overline{\text{EOP}}$ 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
$\overline{\text{EOP}}$ 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

SECTION IV

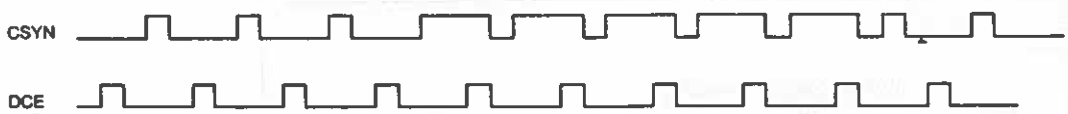
LINE TIMING



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	MAX	UNIT
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.

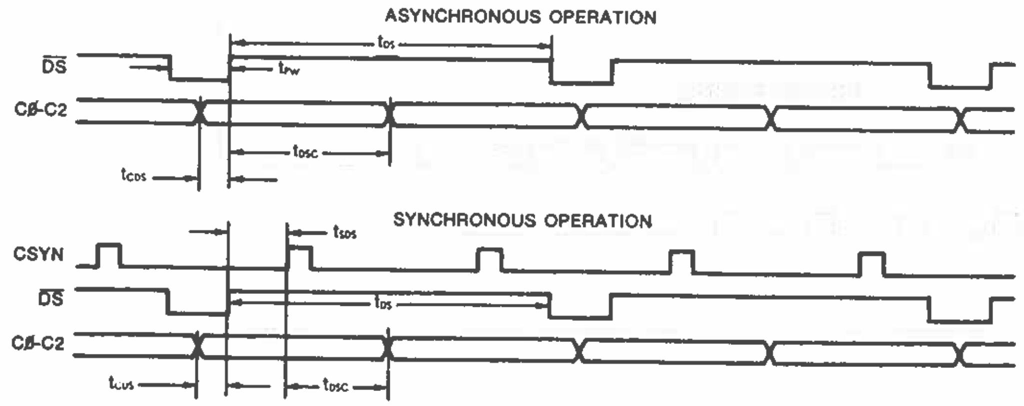
Synchronous Operation

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	MAX	UNIT
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.

DATA INPUT TIMING

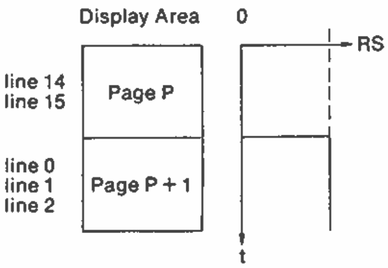


SECTION IV

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.

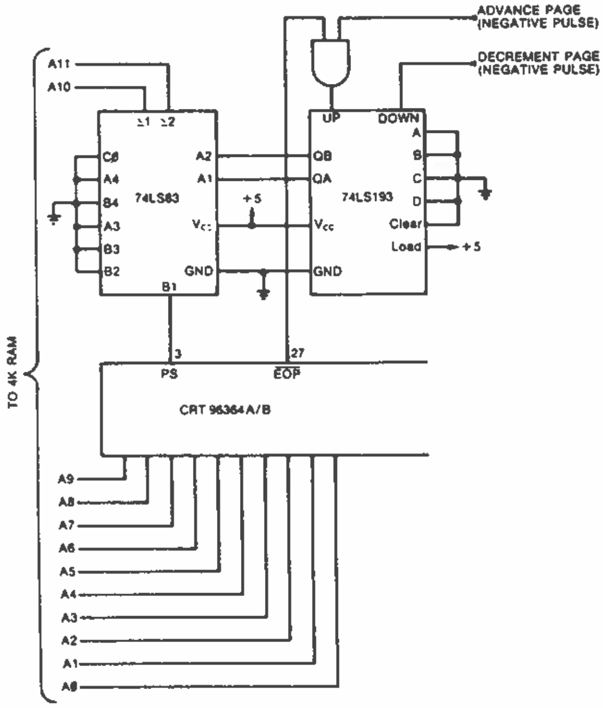
PS (Page Select) indicates the end of page location. If a scroll has occurred, PS will show the transition from the end of line 15 of page P and the beginning of line 0 of page P + 1.



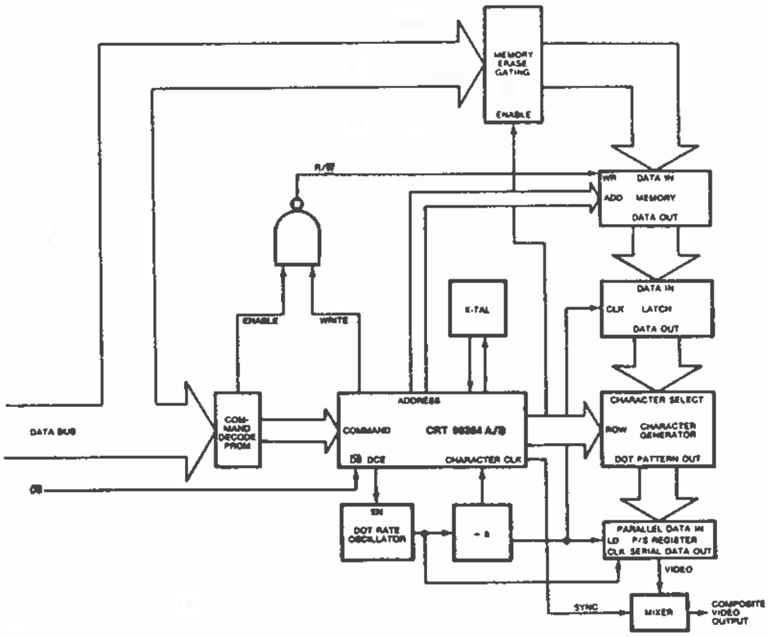
To properly maintain the memory address when displaying more than two pages, $\overline{EOP}$ pulses low at the point in time when page P is scrolled completely off the screen. At this time, RS will remain low for the entire frame since page P + 1 is now the only displayed page.

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.

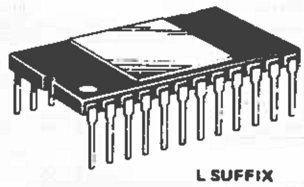
The bus interface of the MC6850 includes select, enable, read/write, interrupt and bus interface logic to allow data transfer over an 8-bit bi-directional data bus. The parallel data of the bus system is serially transmitted and received by the asynchronous data interface, with proper formatting and 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)

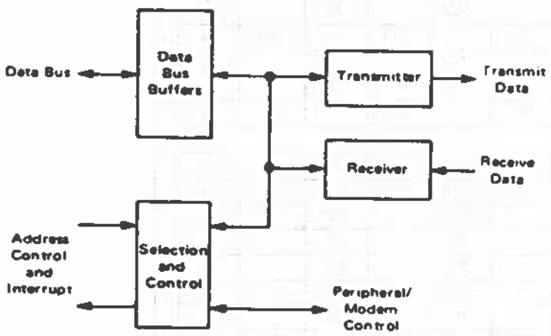
ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTER



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) (I_{Load} = -100 µAdc, Enable Pulse Width <25 µs)	D0-D7 Tx Data, $\overline{RTS}$	V_{OH}	 V_{SS} + 2.4 V_{SS} + 2.4	 – –	 – –	Vdc
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 E, Tx Clk, Rx Clk, R/W, RS, Rx Data, CS0, CS1, $\overline{CS2}$, $\overline{CTS}$, $\overline{DCD}$	C_{in}	– –	10 7.0	12.5 7.5	pF
Output Capacitance (V_{in} = 0, T_A = 25°C, f = 1.0 MHz)	$\overline{RTS}$, Tx Data $\overline{IRQ}$	C_{out}	– –	– –	10 5.0	pF
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 ÷16, ÷64 Modes	f_C	– –	– –	500 800	kHz
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
$\overline{\text{Interrupt Request}}$ Release Time (Figure 6)		t_{IR}	–	–	1.2	µs
$\overline{\text{Request-to-Send}}$ 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

MOTOROLA *Semiconductor Products Inc.*

FIGURE 1 – CLOCK PULSE WIDTH, LOW-STATE

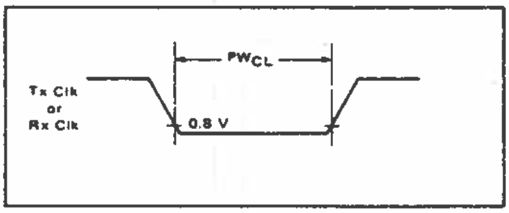


FIGURE 2 – CLOCK PULSE WIDTH, HIGH-STATE

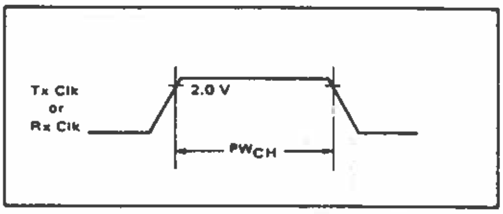


FIGURE 3 – TRANSMIT DATA OUTPUT DELAY

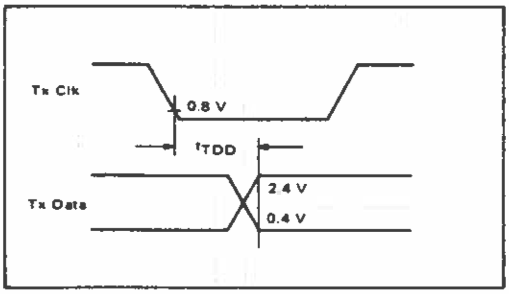


FIGURE 4 – RECEIVE DATA SETUP TIME
(÷1 Mode)

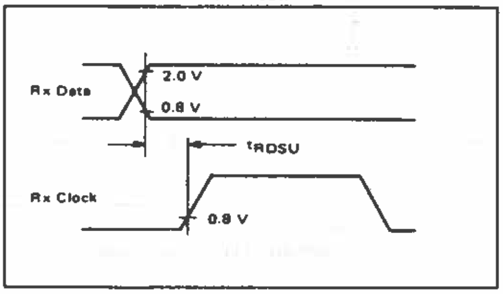


FIGURE 5 – RECEIVE DATA HOLD TIME
(÷1 Mode)

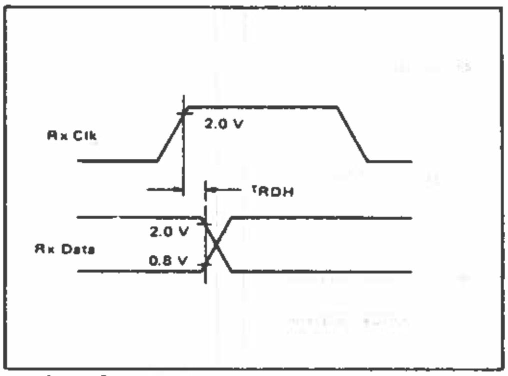


FIGURE 6 – $\overline{\text{REQUEST-TO-SEND}}$ DELAY AND
$\overline{\text{INTERRUPT-REQUEST}}$ RELEASE TIMES

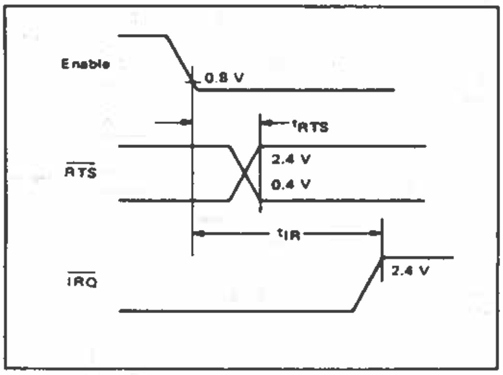


FIGURE 7 – BUS READ TIMING CHARACTERISTICS
(Read information from ACIA)

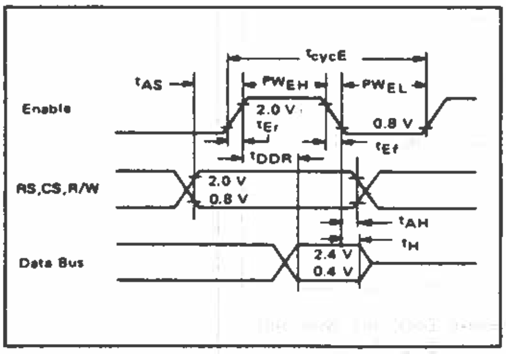


FIGURE 8 – BUS WRITE TIMING CHARACTERISTICS
(Write information into ACIA)

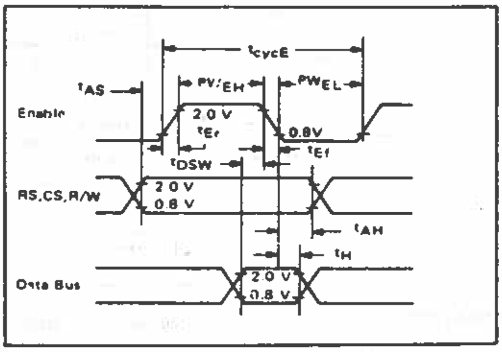
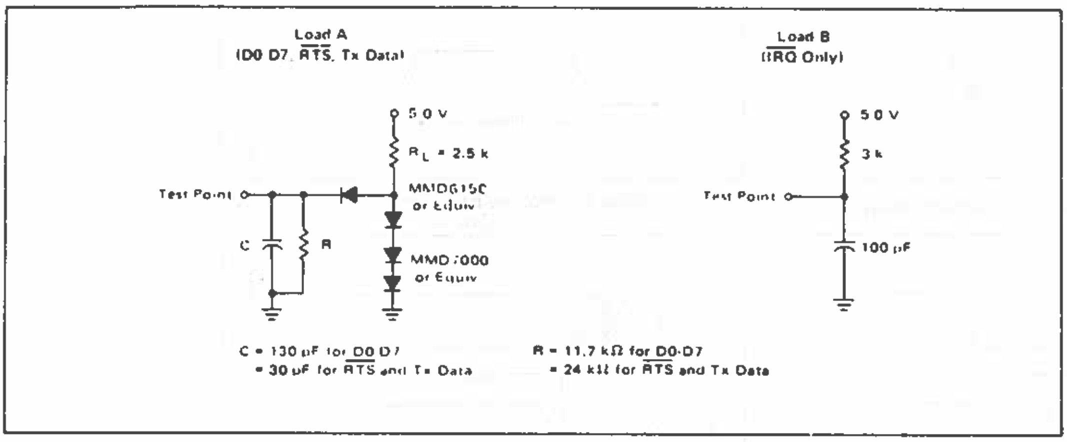
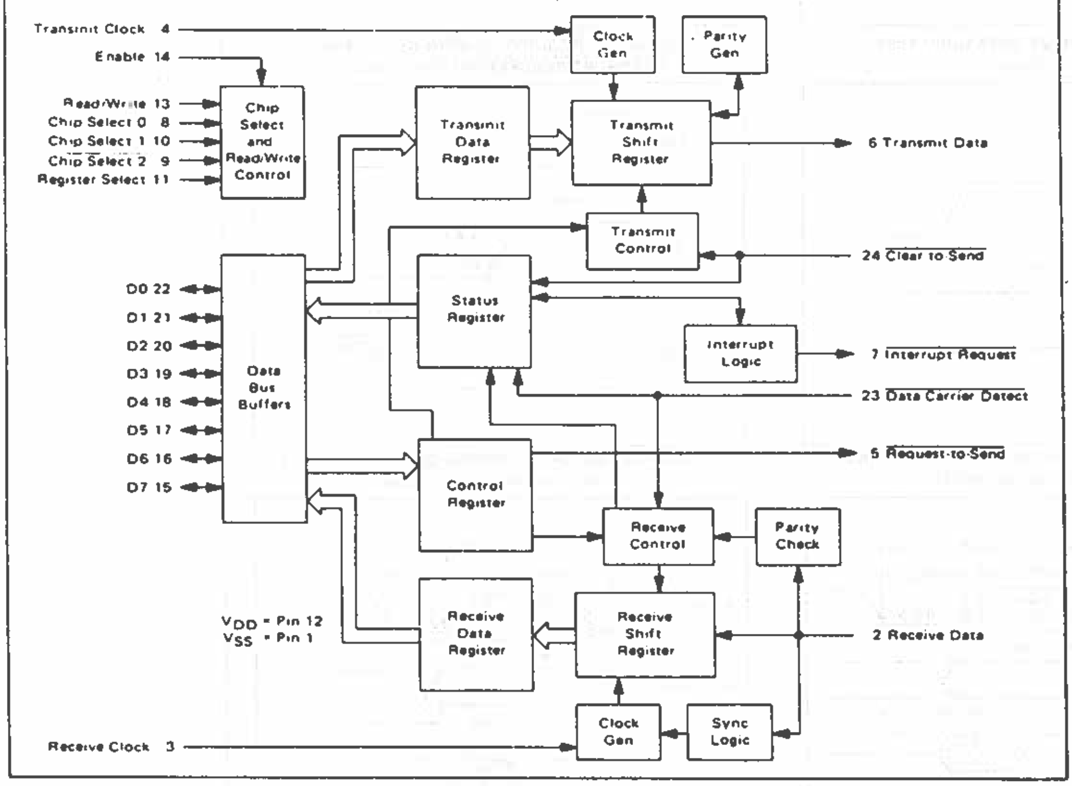


FIGURE 9 – BUS TIMING TEST LOADS



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 $\phi 2$ 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 $\overline{\text{IRQ}}$ 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 $\overline{\text{IRQ}}$ 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 · $\overline{\text{CR6}}$), 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 $\overline{\text{Clear-to-Send}}$ ($\overline{\text{CTS}}$) 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 $\overline{\text{CTS}}$ 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 $\overline{\text{Data Carrier Detect}}$ ($\overline{\text{DCD}}$) 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 $\overline{\text{DCD}}$ 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 $\overline{\text{Clear-to-Send}}$, $\overline{\text{Request-to-Send}}$ and $\overline{\text{Data Carrier Detect}}$.

$\overline{\text{Clear-to-Send}}$ ($\overline{\text{CTS}}$) – This high impedance TTL compatible input provides automatic control of the transmitting end of a communications link via the modem $\overline{\text{Clear-to-Send}}$ active low output by inhibiting the Transmit Data Register Empty (TDRE) status bit.

$\overline{\text{Request-to-Send}}$ ($\overline{\text{RTS}}$) – The $\overline{\text{Request-to-Send}}$ output enables the MPU to control a peripheral or modem via the data bus. The $\overline{\text{RTS}}$ output corresponds to the state of the Control Register bits CR5 and CR6. When CR6 = 0 or both CR5 and CR6 = 1, the $\overline{\text{RTS}}$ output is low (the active state). This output can also be used for $\overline{\text{Data Terminal Ready}}$ ($\overline{\text{DTR}}$).

$\overline{\text{Data Carrier Detect}}$ ($\overline{\text{DCD}}$) – This high impedance TTL compatible input provides automatic control, such as in the receiving end of a communications link by means of a modem $\overline{\text{Data Carrier Detect}}$ output. The $\overline{\text{DCD}}$ input inhibits and initializes the receiver section of the ACIA when high. A low to high transition of the $\overline{\text{Data Carrier Detect}}$ 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 · $\overline{\text{R/W}}$ 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 • $\overline{R/W}$ Transmit Data Register	RS • R/W Receive Data Register	$\overline{RS}$ • $\overline{R/W}$ Control Register	$\overline{RS}$ • R/W Status Register
	(Write Only)	(Read Only)	(Write Only)	(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)	$\overline{\text{Data Carrier Detect}}$ ($\overline{DCD}$)
3	Data Bit 3	Data Bit 3	Word Select 2 (CR3)	$\overline{\text{Clear to Send}}$ ($\overline{CTS}$)
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 $\overline{\text{Request-to-Send}}$ 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 $\overline{CTS}$ and $\overline{DCD}$) 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 $\overline{\text{Request-to-Send}}$ ($\overline{RTS}$) output, and the transmission of a Break level (space). The following encoding format is used:

CR6	CR5	Function
0	0	$\overline{RTS}$ = low, Transmitting Interrupt Disabled.
0	1	$\overline{RTS}$ = low, Transmitting Interrupt Enabled.
1	0	$\overline{RTS}$ = high, Transmitting Interrupt Disabled.
1	1	$\overline{RTS}$ = 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 $\overline{\text{Data Carrier Detect}}$ ($\overline{DCD}$) 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.

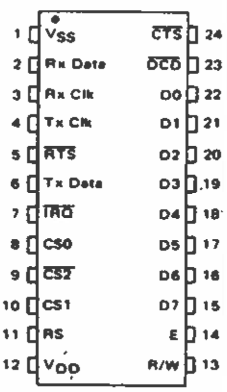
Framing Error (FE), Bit 4 – Framing error indicates that the received character is improperly framed by a start and a stop bit and is detected by the absence of the 1st stop bit. This error indicates a synchronization error, faulty transmission, or a break condition. The framing error flag is set or reset during the receive data transfer time. Therefore, this error indicator is present throughout the time that the associated character is available.

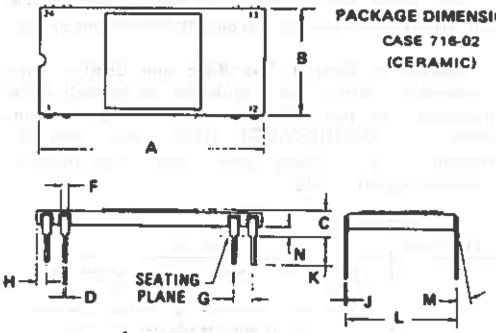
Receiver Overrun (OVRN), Bit 5 – Overrun is an error flag that indicates that one or more characters in the data stream were lost. That is, a character or a number of characters were received but not read from the Receive Data Register (RDR) prior to subsequent characters being received. The overrun 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		INCHES	
	MIN	MAX	MIN	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-A78/5.0 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.

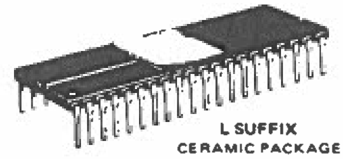
This very cost-effective MPU allows the designer to use the MC6808 in consumer as well as industrial applications without sacrificing industrial specifications.

- 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

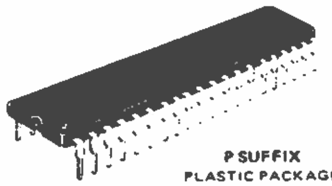
MOS

(N-CHANNEL, SILICON-GATE, DEPLETION LOAD)

MICROPROCESSOR WITH CLOCK

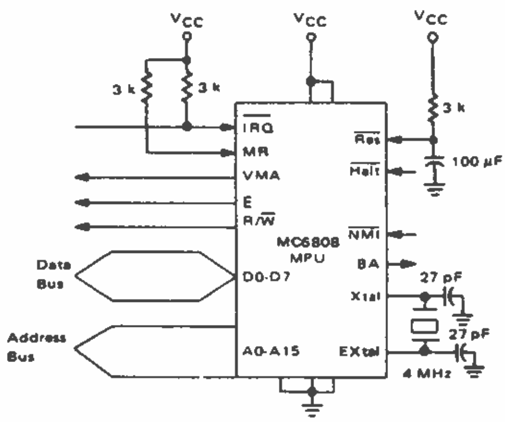


L SUFFIX
CERAMIC PACKAGE
CASE 715

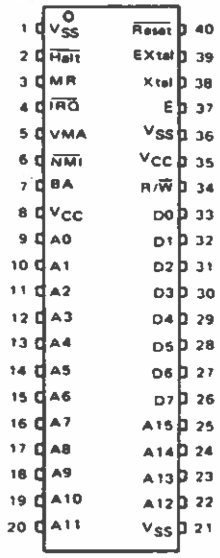


P SUFFIX
PLASTIC PACKAGE
CASE 711

FIGURE 1 – TYPICAL MICROPROCESSOR INTERFACE



PIN ASSIGNMENT



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

 ADI-806

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 unles otherwise noted.)

Characteristic		Symbol	Min	Typ	Max	Unit
Input High Voltage	Logic, EXtal $\overline{\text{Reset}}$	V_{IH}	V_{SS} + 2.0 V_{SS} + 4.0		V_{CC} V_{CC}	Vdc
Input Low Voltage	Logic, EXtal $\overline{\text{Reset}}$	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/$\overline{\text{W}}$, 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	C_{in}		 10 6.5	 12.5 10	pF
	A0-A15, R/$\overline{\text{W}}$, VMA	C_{out}			12	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		t_{cyc}	1.0		10	µs
Clock Pulse Width (measured at 2.4V) (measured at 0.4V)		$PW_{\phi H}$ $PW_{\phi L}$	450 450		4500 4500	ns
Fall Time (Measured between V_{SS} + 0.4 V and V_{SS} + 2.4 V)		t_{ϕ}			25	ns

*Except $\overline{\text{IRQ}}$ and $\overline{\text{NMI}}$, 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 $t_{acc} = t_{ut} - (t_{AD} + t_{DSR})$; $t_{ut} = t_{cyc} - t_{\phi}$	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/$\overline{\text{W}}$, 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

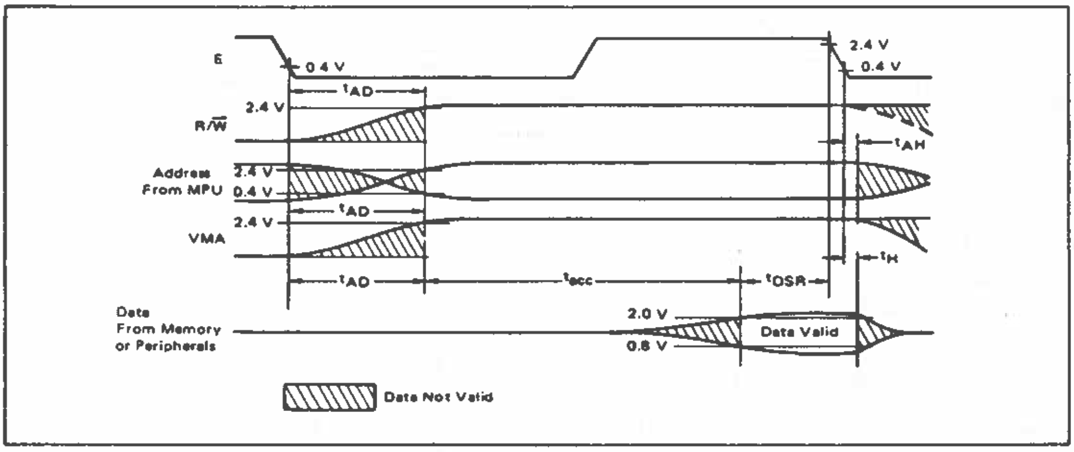


FIGURE 3 – WRITE DATA IN MEMORY OR PERIPHERALS

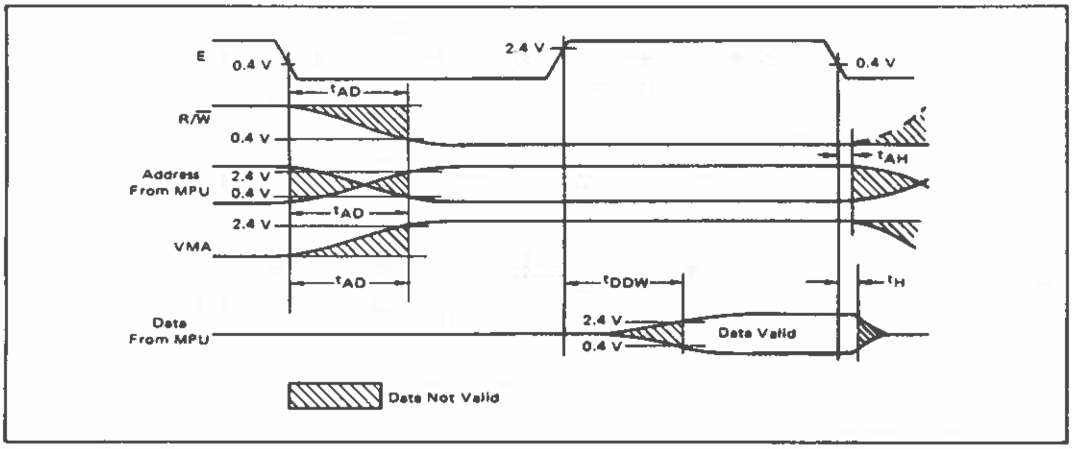
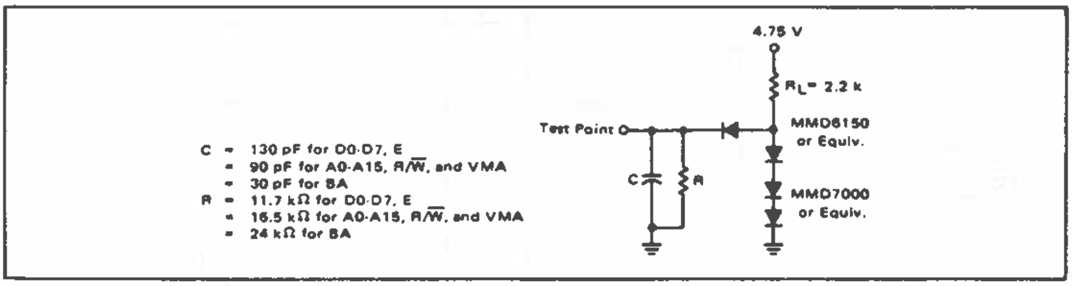


FIGURE 4 – BUS TIMING TEST LOAD



MC6808

FIGURE 5 – TYPICAL DATA BUS OUTPUT DELAY versus CAPACITIVE LOADING

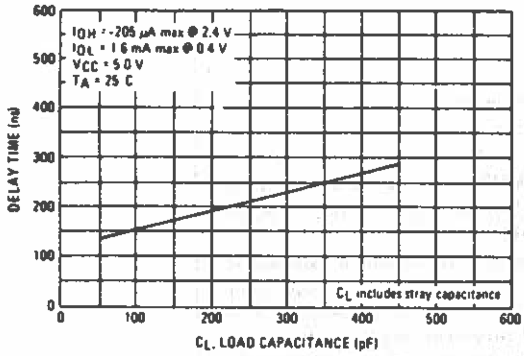


FIGURE 6 – TYPICAL READ/WRITE, VMA, AND ADDRESS OUTPUT DELAY versus CAPACITIVE LOADING

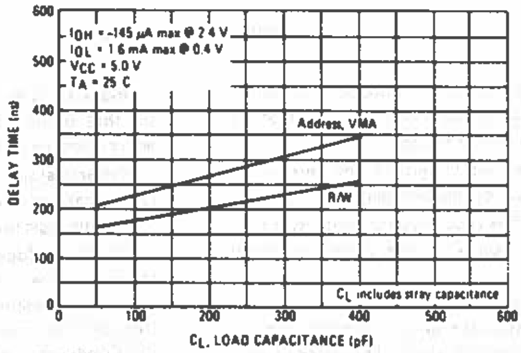
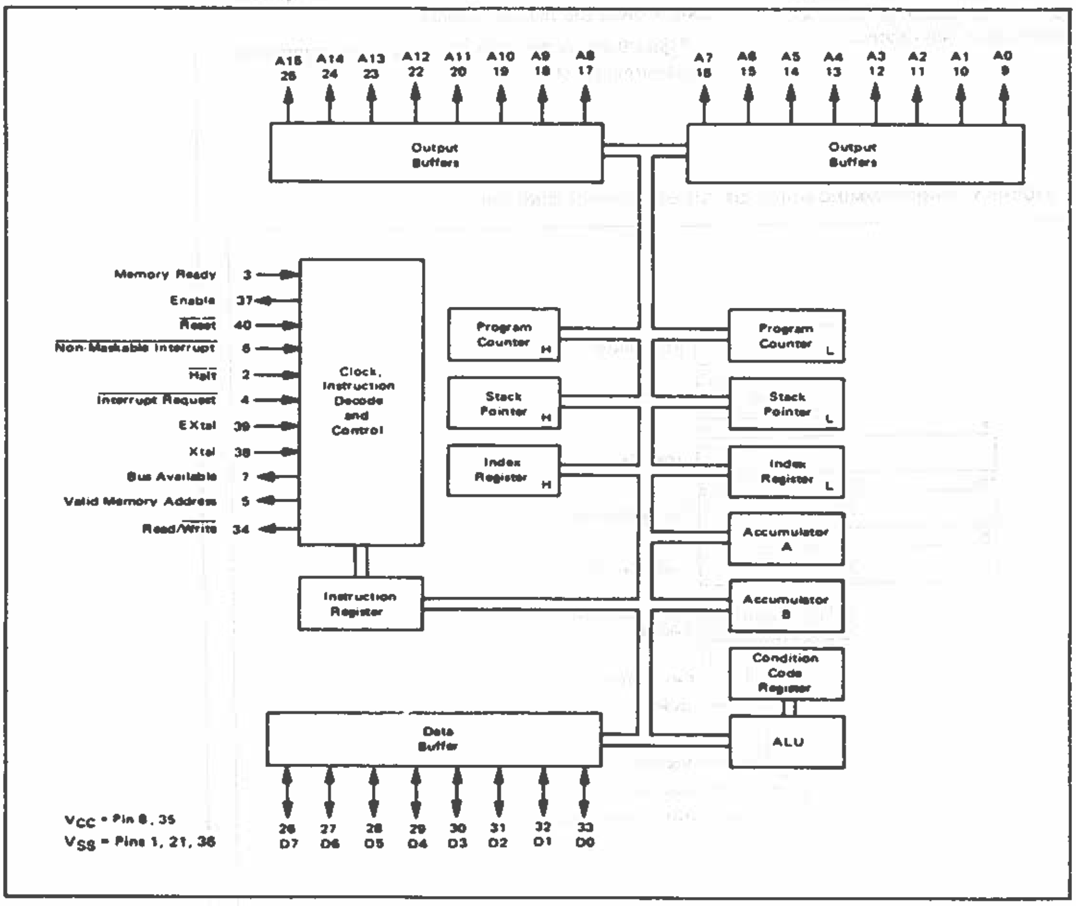


FIGURE 7 – MC6808 EXPANDED BLOCK DIAGRAM



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

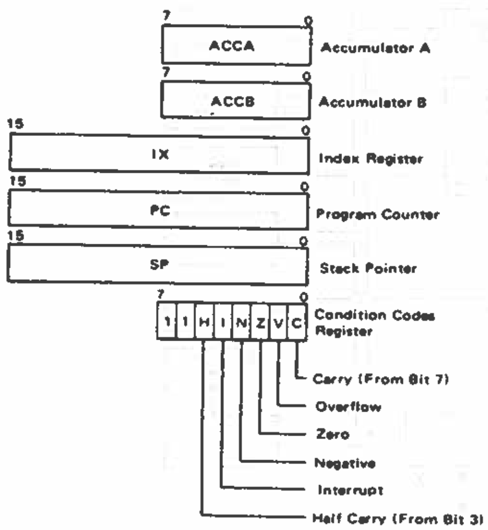
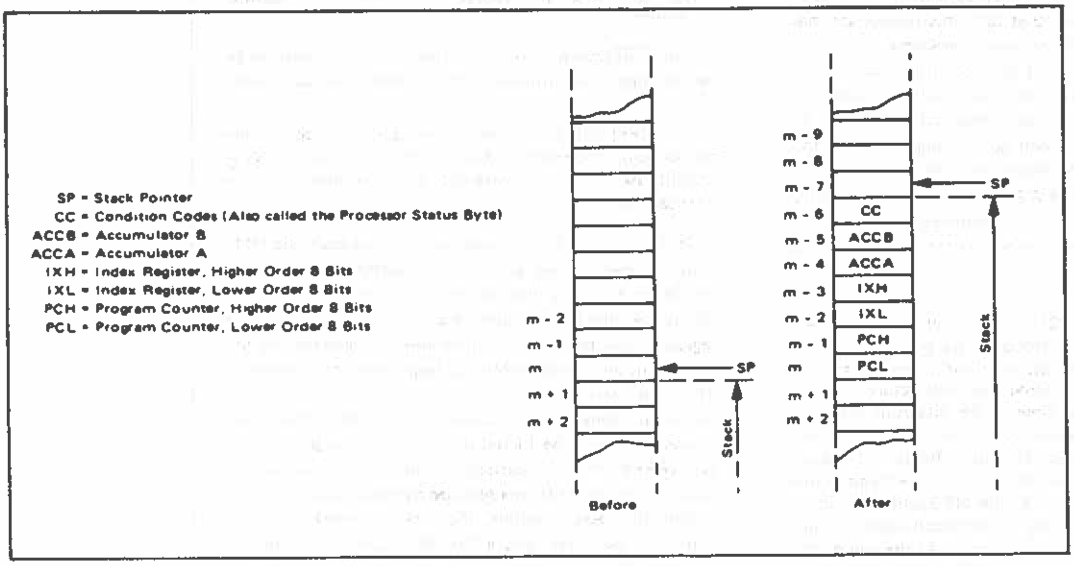


FIGURE 9 – SAVING THE STATUS OF THE MICROPROCESSOR IN THE STACK



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, $\phi1$, $\phi2$ 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 $\phi2$ 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 $\overline{\text{Halt}}$ 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.

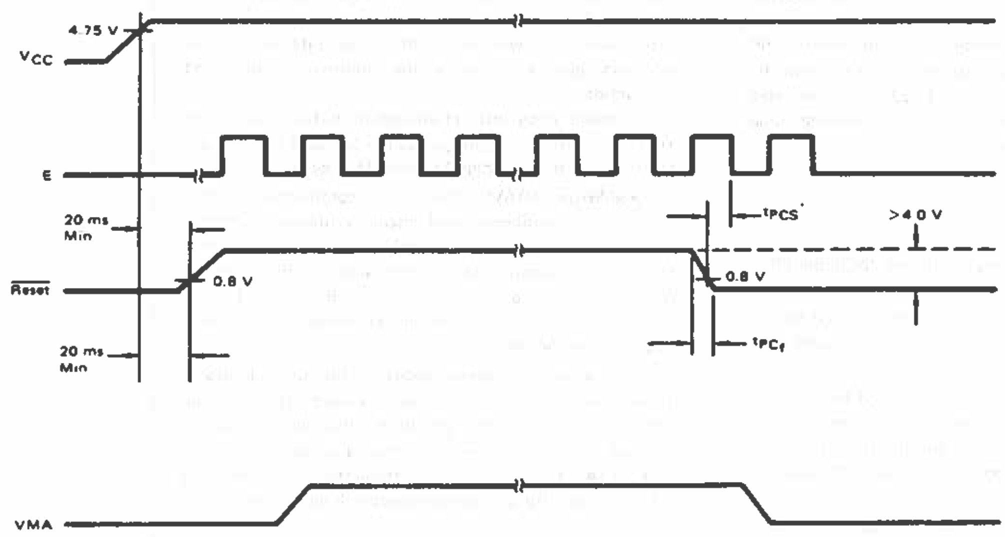
Interrupt Request ($\overline{\text{IRQ}}$) — This level sensitive input requests that an interrupt sequence be generated within the machine. The processor will wait until it completes the current instruction that is being executed before it recognizes the request. At that time, if the interrupt mask 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 $\overline{\text{Halt}}$ line must be in the high state for interrupts to be serviced. Interrupts will be latched internally while $\overline{\text{Halt}}$ is low.

The $\overline{\text{IRQ}}$ 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.

$\overline{\text{Reset}}$ — 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 IRQ. Power-up and reset timing sequences are shown in Figure 10.

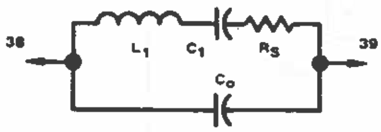
FIGURE 10 – POWER-UP AND RESET TIMING



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

Crystal Parameters:



AT — Cut Parallel Resonance Crystal
C_o = 7 pF Max.
FREQ = 4.0 MHz @ C_L = 24pF
R_S = 50 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.

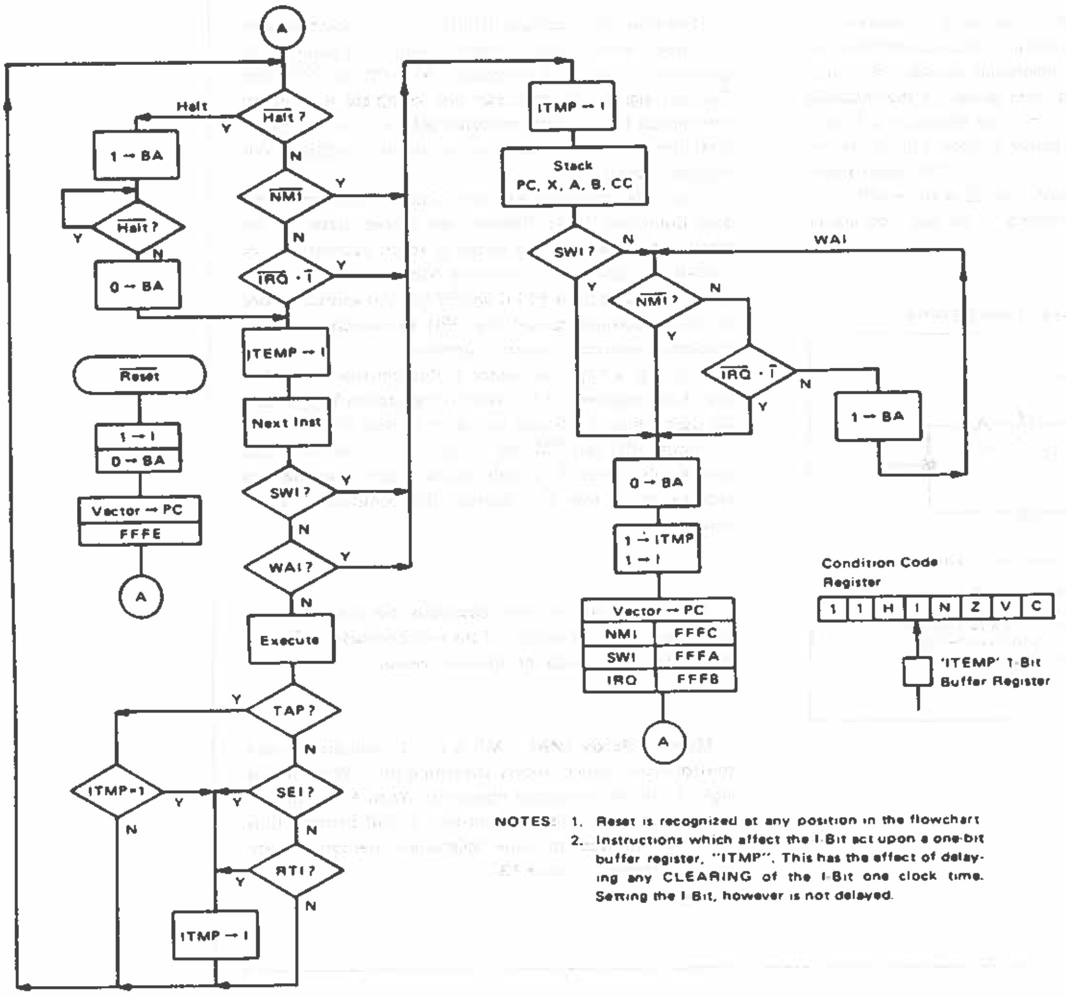
Figure 12 is a flow chart describing the major decision paths and interrupt vectors of the microprocessor. Table 1 gives the memory map for interrupt vectors.

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

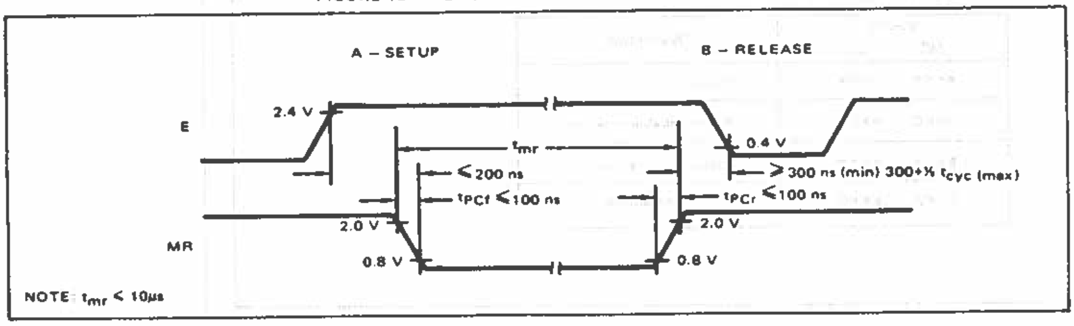
Vector MS	Vector LS	Description
FFFE	FFFF	Restart
FFFC	FFFD	Non-maskable Interrupt
FFFA	FFFB	Software Interrupt
FFF8	FFF9	Interrupt Request

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

		ADDRESSING MODES															BOOLEAN/ARITHMETIC OPERATION	COND. CODE REG.					
		IMMED			DIRECT			INDEX			EXTND			IMPLIED			(All register labels refer to contents)	5	4	3	2	1	0
OPERATIONS	MNEMONIC	OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Add	ADDA	8B	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 Acmltrs	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 Acmltrs	CBA													11	2	1	A − B	•	•	↕	↕	↕	↕
Complement, 1's	COM							63	7	2	73	6	3				$\overline{M}$ → M	•	•	↕	↕	R	S
	COMA													43	2	1	$\overline{A}$ → A	•	•	↕	↕	R	S
	COMB													53	2	1	$\overline{B}$ → B	•	•	↕	↕	R	S
Complement, 2's (Negate)	NEG							60	7	2	70	6	3				00 − M → M	•	•	↕	↕	1	2
	NEGA													40	2	1	00 − A → A	•	•	↕	↕	1	2
	NEGB													50	2	1	00 − B → B	•	•	↕	↕	1	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 Acmltr	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 C ← b7 ← b0	•	•	↕	↕	6	↕
	ROLB													59	2	1	B	•	•	↕	↕	6	↕
Rotate Right	ROR							66	7	2	76	6	3				M	•	•	↕	↕	6	↕
	RORA													46	2	1	A C ← b7 → b0	•	•	↕	↕	6	↕
	RORB													56	2	1	B	•	•	↕	↕	6	↕
Shift Left, Arithmetic	ASL							68	7	2	78	6	3				M	•	•	↕	↕	6	↕
	ASLA													48	2	1	A C ← b7 … b0 ← 0	•	•	↕	↕	6	↕
	ASLB													58	2	1	B	•	•	↕	↕	6	↕
Shift Right, Arithmetic	ASR							67	7	2	77	6	3				M	•	•	↕	↕	6	↕
	ASRA													47	2	1	A b7 … b0 → C	•	•	↕	↕	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 0 → b7 … b0 → C	•	•	R	↕	6	↕
	LSRB													54	2	1	B	•	•	R	↕	6	↕
Store Acmltr	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 Acmltrs	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 Acmltrs	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

LEGEND

OP Operation Code (Hexadecimal)
~ Number of MPU Cycles
# Number of Program Bytes
\+ Arithmetic Plus
− Arithmetic Minus
· Boolean AND
M_{SP} Contents of memory location pointed to be Stack Pointer

\+ Boolean Inclusive OR
⊕ Boolean Exclusive OR
$\overline{M}$ Complement of M
→ Transfer Into
0 Bit = Zero
00 Byte = Zero

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

MC6808

TABLE 4 – INDEX REGISTER AND STACK MANIPULATION INSTRUCTIONS

		IMMED			DIRECT			INDEX			EXTND			IMPLIED				COND. CODE REG. 5	4	3	2	1	0
POINTER OPERATIONS	MNEMONIC	OP	~	=	OP	~	=	OP	~	=	OP	~	=	OP	~	=	BOOLEAN/ARITHMETIC OPERATION	H	I	N	Z	V	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

		RELATIVE			INDEX			EXTND			IMPLIED				COND. CODE REG. 5	4	3	2	1	0
OPERATIONS	MNEMONIC	OP	~	=	OP	~	=	OP	~	=	OP	~	=	BRANCH TEST	H	I	N	Z	V	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										$N \oplus V = 0$	•	•	•	•	•	•
Branch If > Zero	BGT	2E	4	2										$Z + (N \oplus V) = 0$	•	•	•	•	•	•
Branch If Higher	BHI	22	4	2										$C + Z = 0$	•	•	•	•	•	•
Branch If ≤ Zero	BLE	2F	4	2										$Z + (N \oplus V) = 1$	•	•	•	•	•	•
Branch If Lower Or Same	BLS	23	4	2										$C + Z = 1$	•	•	•	•	•	•
Branch If < Zero	BLT	2D	4	2										$N \oplus V = 1$	•	•	•	•	•	•
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										See Special Operations	•	•	•	•	•	•
Jump	JMP				6E	4	2	7E	3	3				See Special Operations	•	•	•	•	•	•
Jump To Subroutine	JSR				AD	8	2	BD	9	3				See Special Operations	•	•	•	•	•	•
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	See Special Operations	•	•	•	•	•	•
Software Interrupt	SWI										3F	12	1	See Special Operations	•	•	•	•	•	•
Wait for Interrupt*	WAI										3E	9	1	See Special Operations	•	11	•	•	•	•

*WAI puts Address Bus, R/W, and Data Bus in the three-state mode while VMA is held low.

MC6808

SPECIAL OPERATIONS

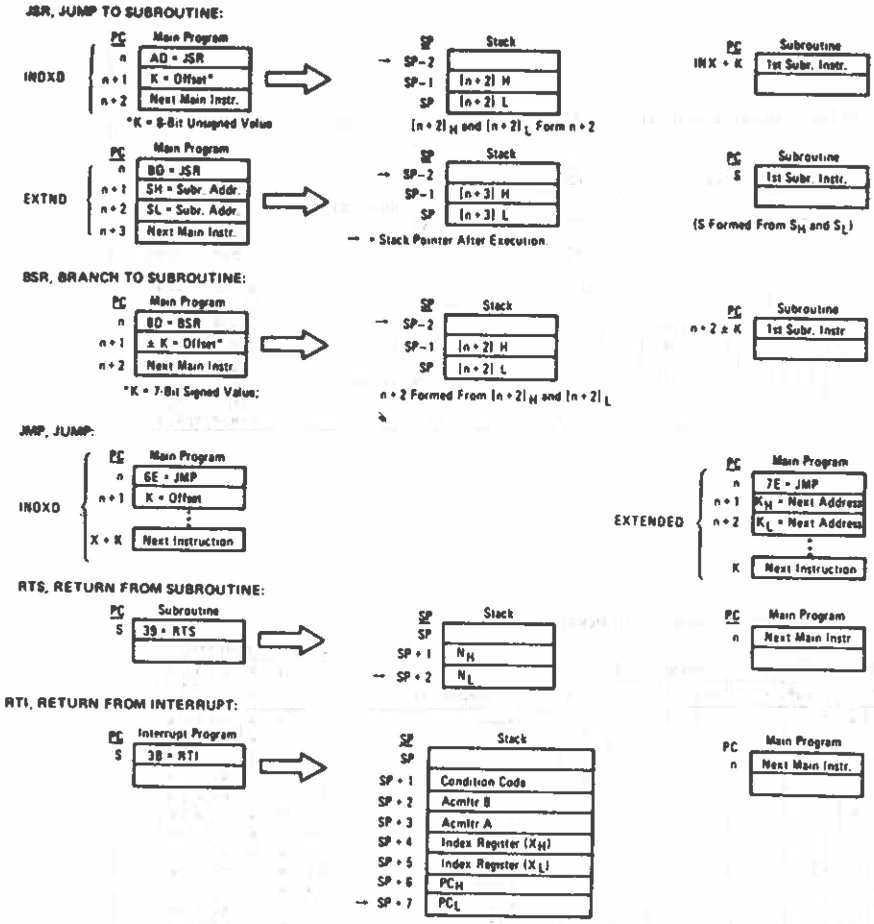


TABLE 6 – CONDITION CODE REGISTER MANIPULATION INSTRUCTIONS

		IMPLIED				COND. CODE REG.					
						5	4	3	2	1	0
OPERATIONS	MNEMONIC	OP	~	=	BOOLEAN OPERATION	H	I	N	Z	V	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.

MC6808

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)

MC6808

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)

TABLE 8 – OPERATION SUMMARY (Continued)

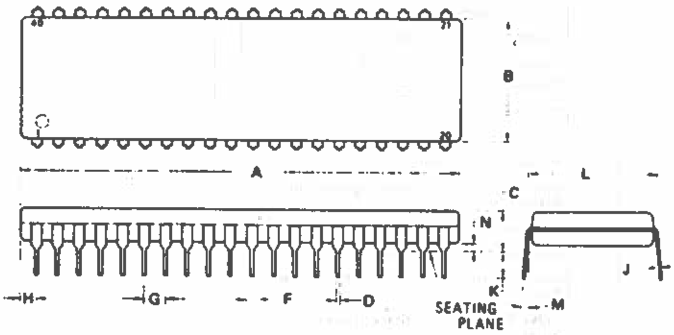
Address Mode and Instructions	Cycles	Cycle #	VMA Line	Address Bus	R/W Line	Data Bus
EXTENDED (Continued)						
STS STX	6	1	1	Op Code Address	1	Op Code
		2	1	Op Code Address + 1	1	Address of Operand (High Order Byte)
		3	1	Op Code Address + 2	1	Address of Operand (Low Order Byte)
		4	0	Address of Operand	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 FFFB (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.

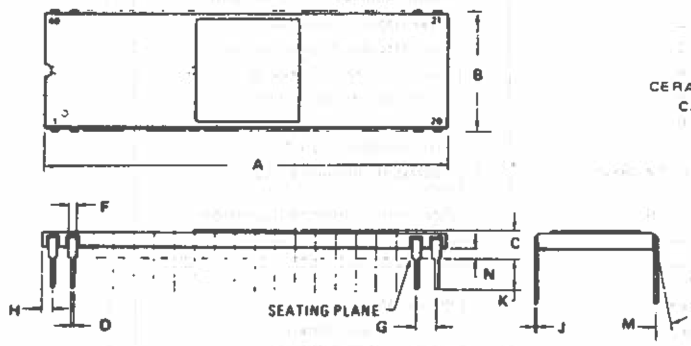
P SUFFIX
PLASTIC PACKAGE
CASE 711-03



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.

DIM	MILLIMETERS MIN	MILLIMETERS MAX	INCHES MIN	INCHES 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



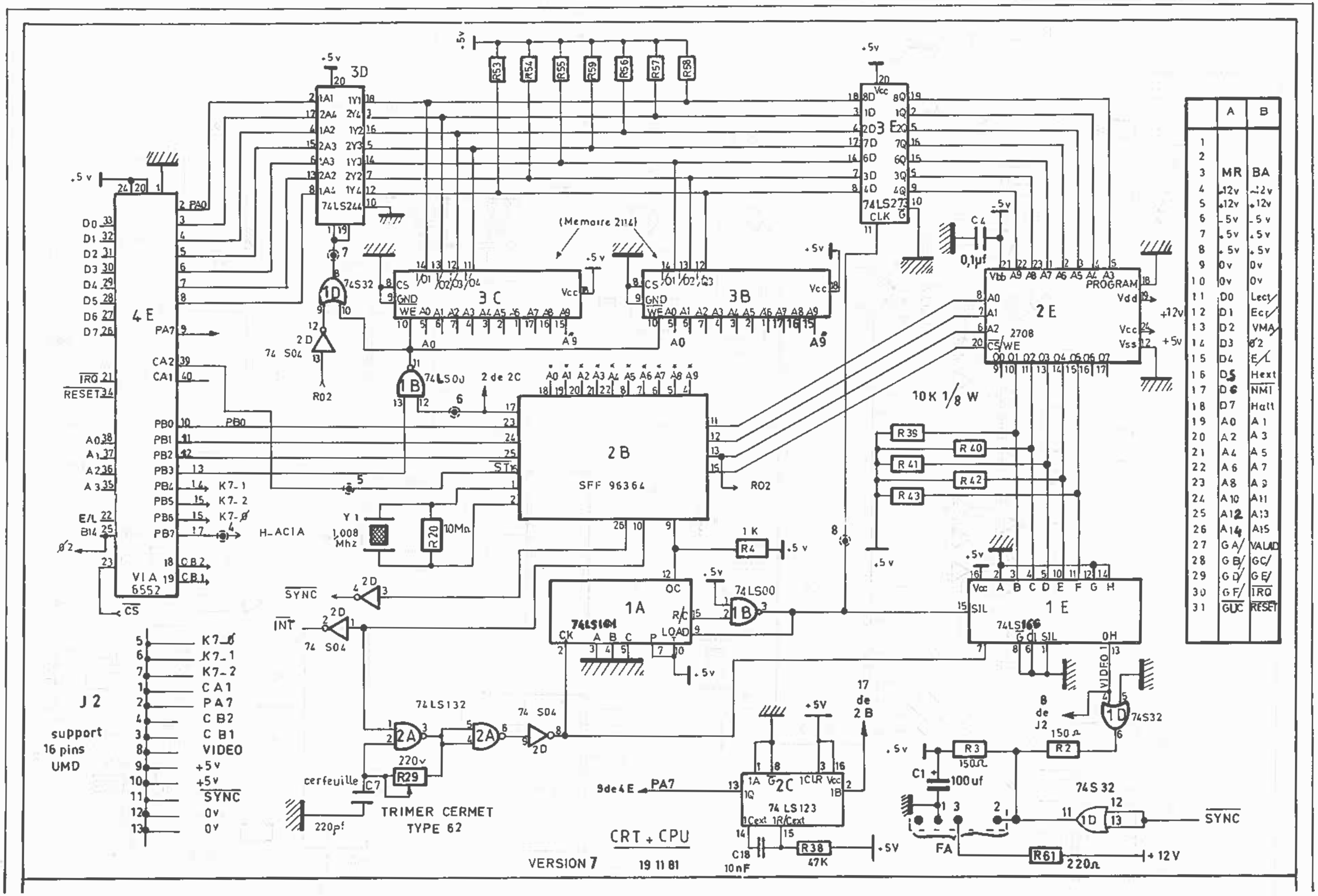
DIM	MILLIMETERS MIN	MILLIMETERS MAX	INCHES MIN	INCHES MAX
A	50.29	51.31	1.980	2.020
B	14.86	15.62	0.585	0.615
C	2.54	4.19	0.100	0.165
D	0.38	0.53	0.015	0.021
F	0.76	1.40	0.030	0.055
G	2.54 BSC		0.100 BSC	
H	0.76	1.78	0.030	0.070
J	0.20	0.33	0.008	0.013
K	2.54	4.19	0.100	0.165
M	0°	10°	0°	10°
N	0.51	1.52	0.020	0.060

NOTE:
1. LEADS, TRUE POSITIONED WITHIN 0.25 mm (0.010) DIA (AT SEATING PLANE), AT MAX. MAT'L CONDITION.

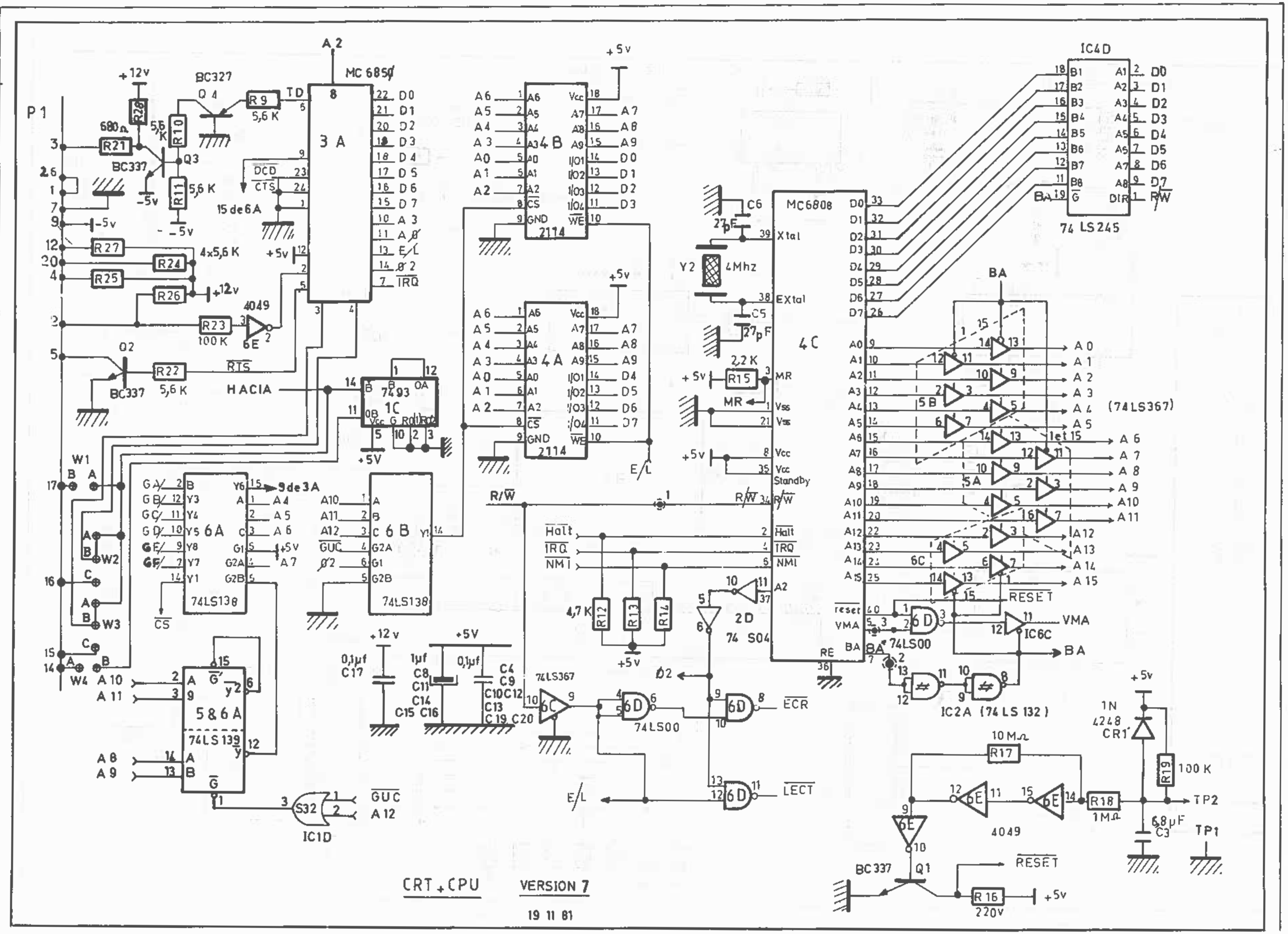
- VIII -

- SCHEMAS LOGIQUES DE FONCTIONNEMENT DES CARTES DE GOUPIL 2 -

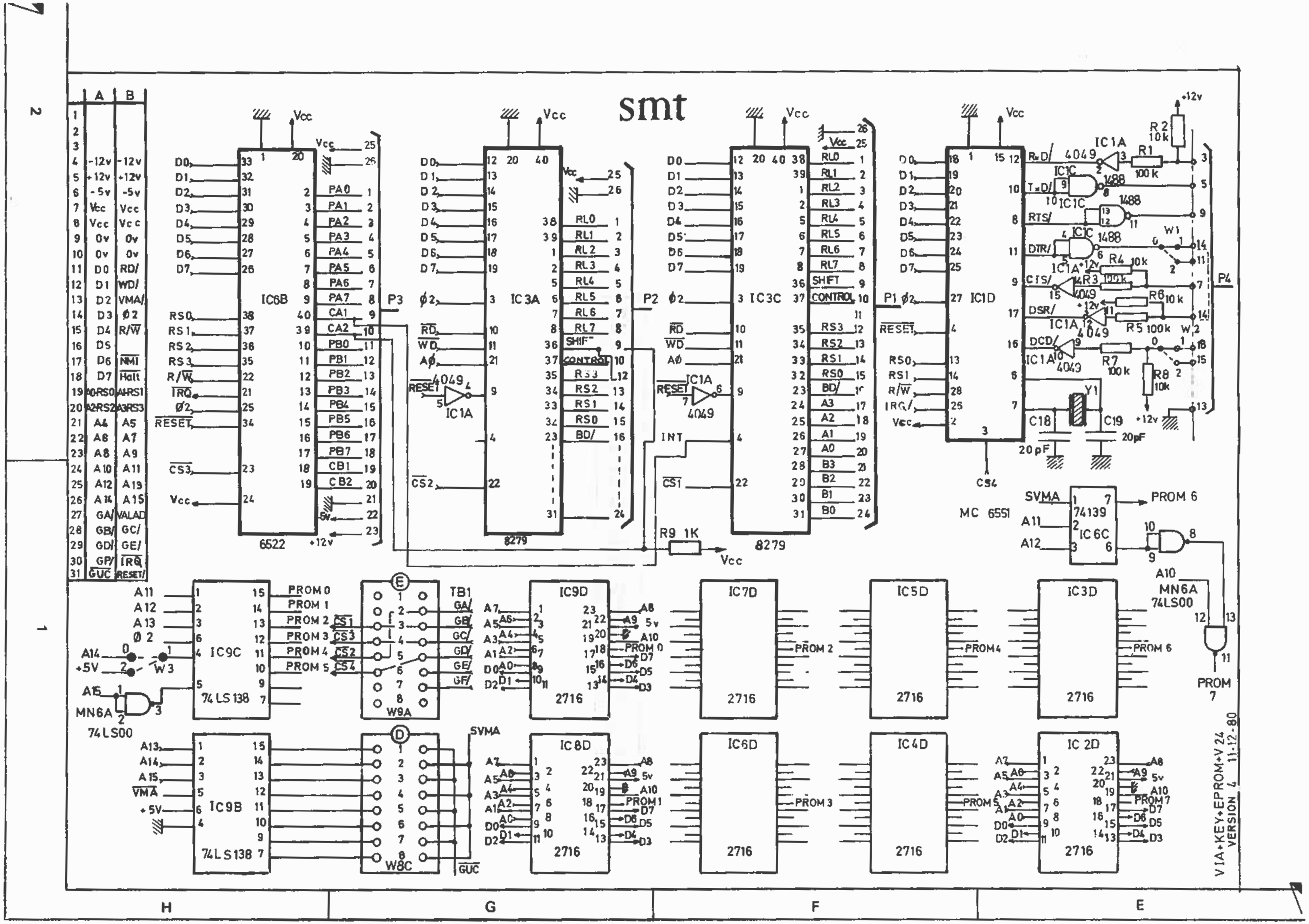
SCHEMAS DE LA CARTE CRT-CPU



	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/	VALID
28	GB/	GC/
29	GD/	GE/
30	GF/	IRQ
31	GUC	RESET


MC 6850
3 A
MC6808
4 C
2114
4 B
4 A
7493
1C
74LS138
6 A
6 B
74LS139
5&6A
IC4D
74 LS245
(74LS367)
74LS00
IC2A (74 LS 132)
74 S04
4049
HACIA
CRT + CPU
VERSION 7
19 11 81

SCHEMAS DE LA CARTE VIA ENTREES-SORTIES


smt
IC6B
6522
IC3A
8279
IC3C
8279
IC1D
MC 6551
IC1A 4049
IC1C 1488
R9 1K
74139
IC6C
MN6A 74LS00
IC9C 74LS138
IC9B 74LS138
W9A
W8C
IC9D 2716
IC7D 2716
IC5D 2716
IC3D 2716
IC8D 2716
IC6D 2716
IC4D 2716
IC 2D 2716
PROM 7
VIA+KEY+EPROM+V24
VERSION 4
11.12.80
H
G
F
E

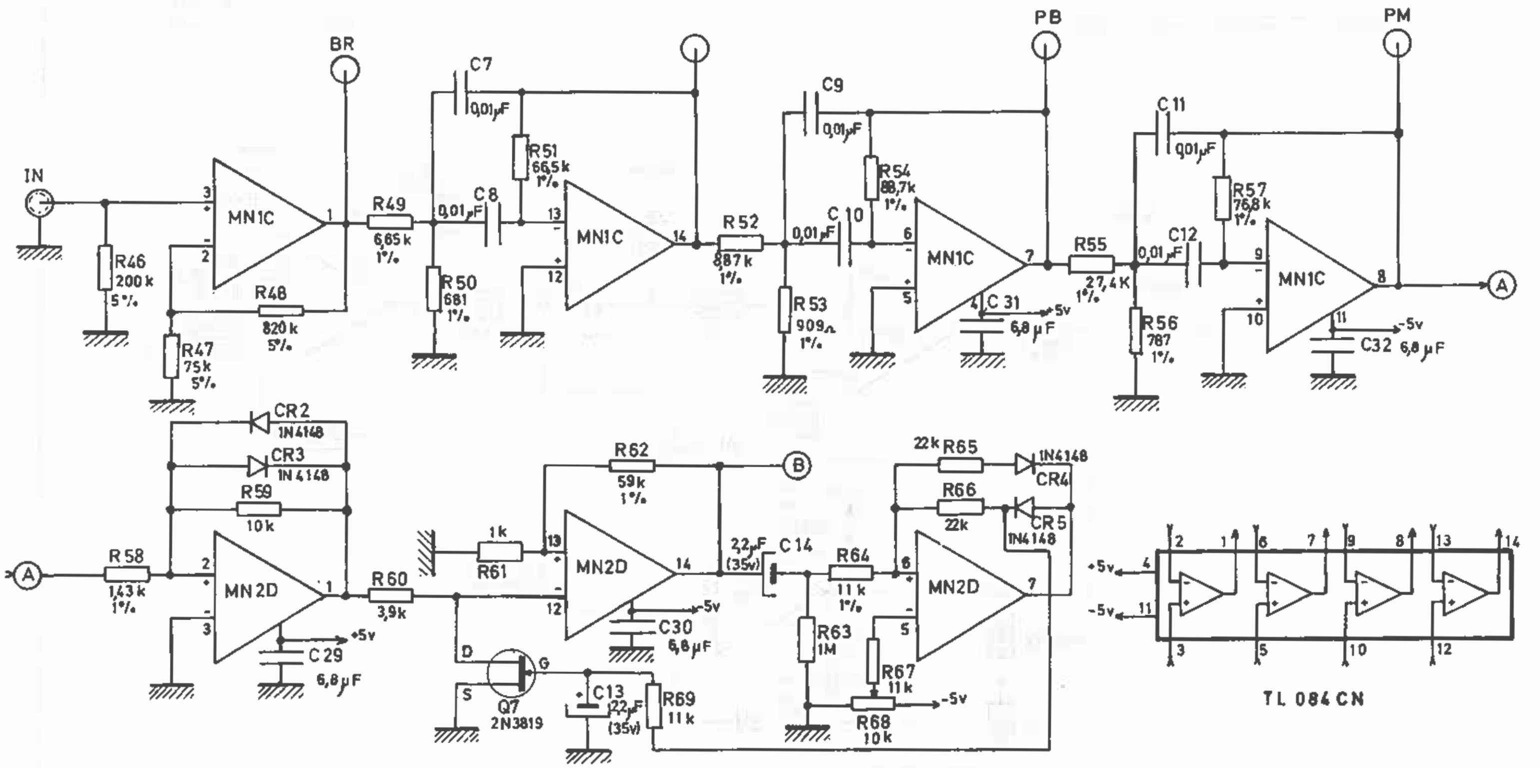
SCHEMAS DE LA CARTE MEMOIRE 16 K STATIQUE

VIII 8

IMPAIRE
PAIRE
2708
PROM 1
PROM 2
74LS138
2114
A
B
-12v
+12v
-5v
+5v
0v
PROTEC
smt
MEMOIRE 16K

SCHEMAS DE LA CARTE MODEM + K7 + MEMOIRE DYNAMIQUE

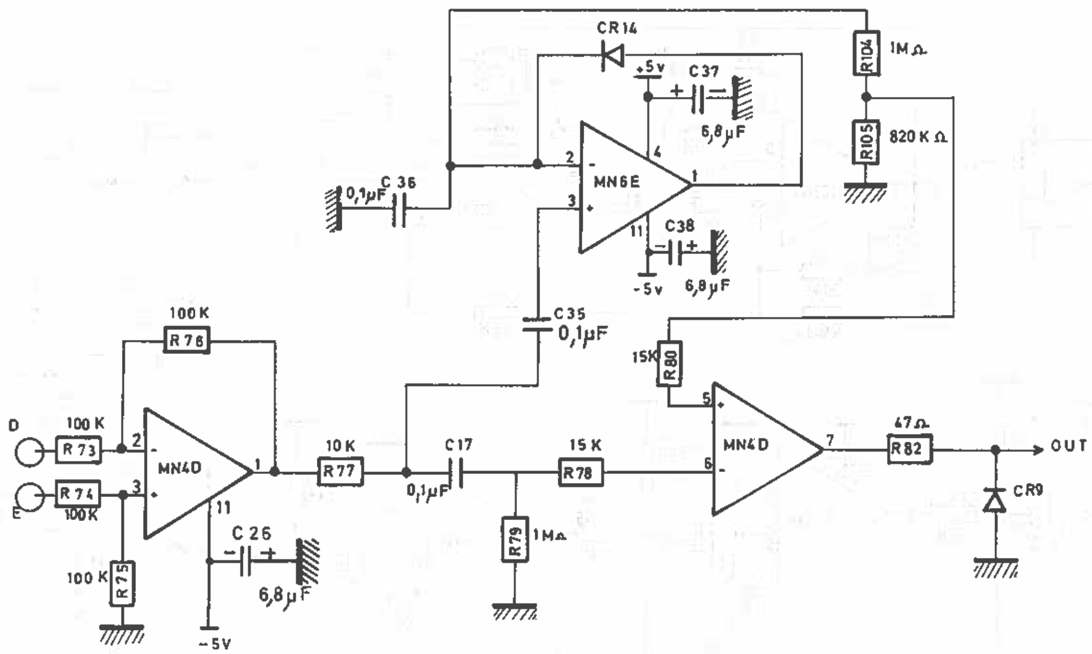
- Version 3 sans paddles
- Version 6 avec paddles



MN 1C : TL084CN
MN 2D : "

smt

MODEM+K7+MUSIQUE+REPROM
VERSION 3 31-10-80

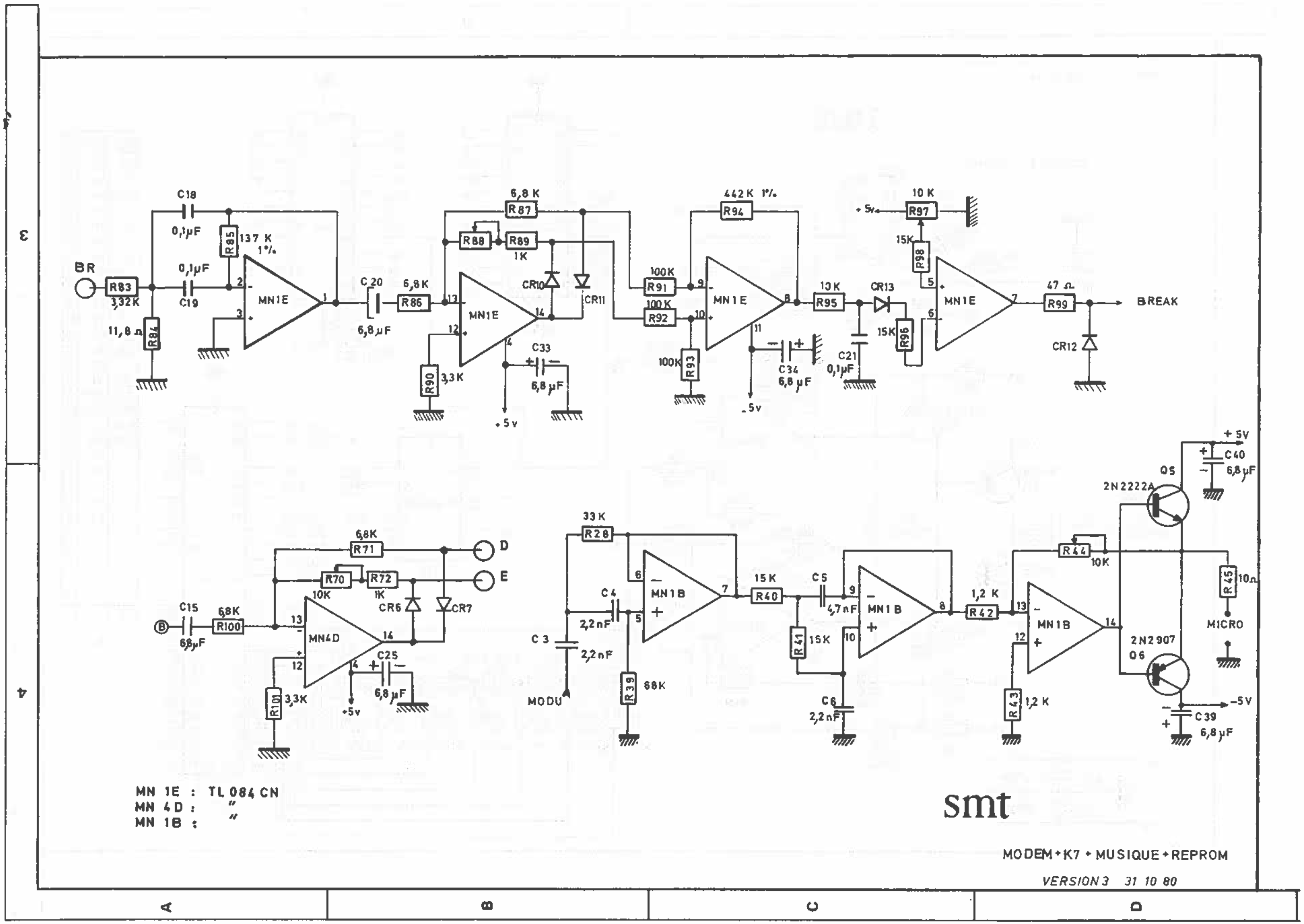


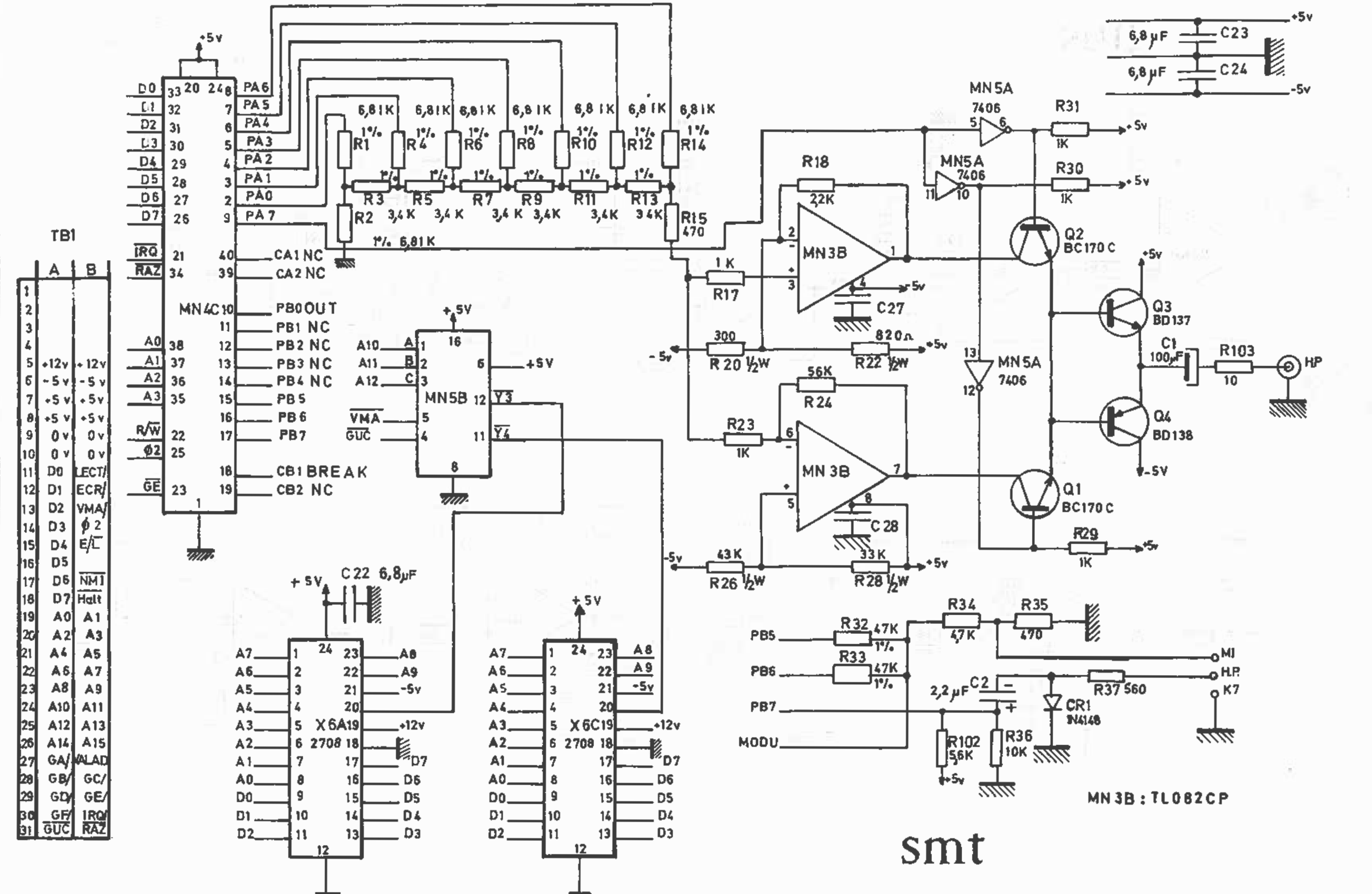
MN4D : TL084CN
MN6E : "

MODEM + K7 + MUSIQUE + REPROM

VERSION 3 31 10 80

smt

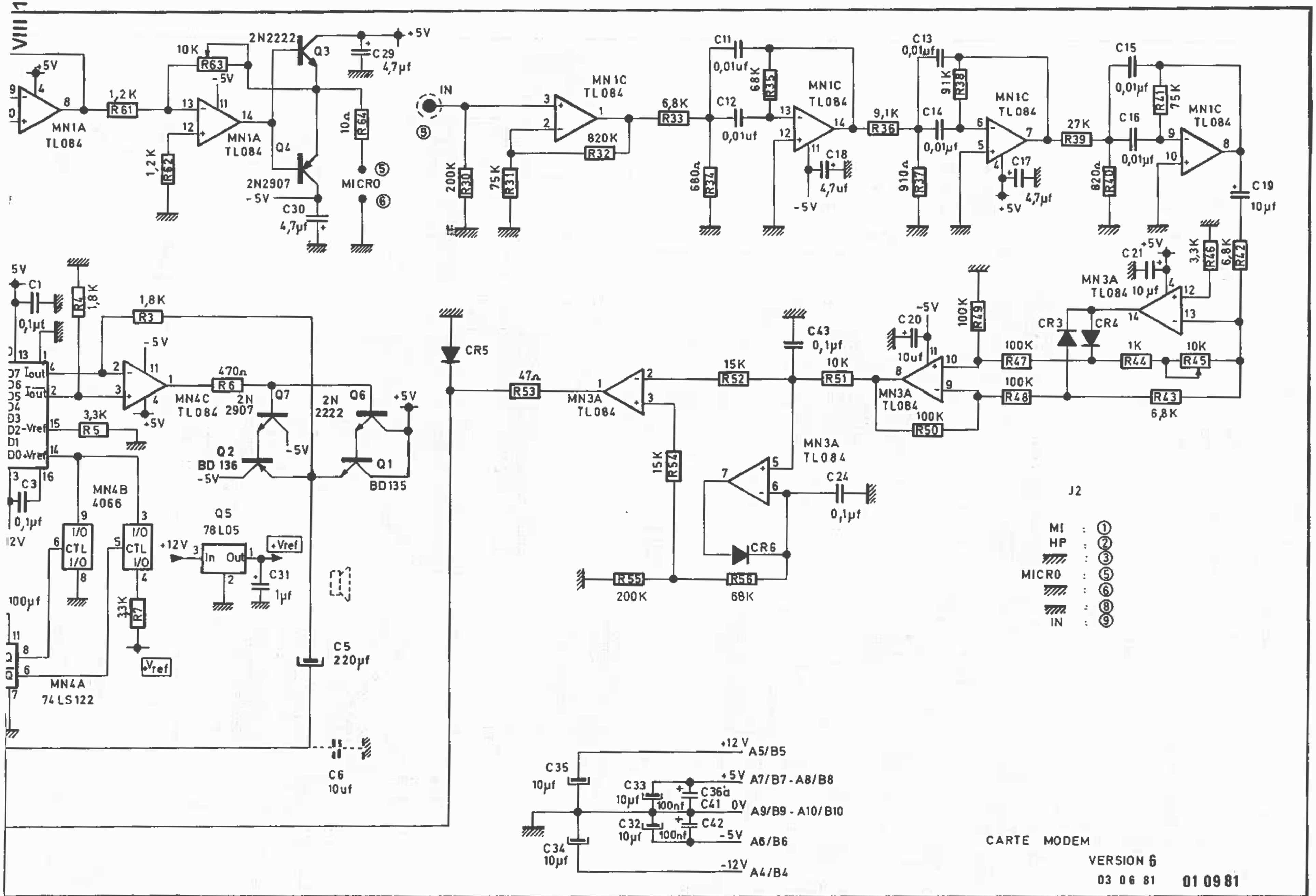

BR
R83
3,32K
R84
11,8 Ω
C18
0,1µF
C19
0,1µF
R85
137 K 1%
MN1E
C 20
6,8µF
R86
6,8K
R87
6,8 K
R88
R89
1K
CR10
CR11
MN1E
R90
3,3K
C33
6,8µF
+5v
R94
442K 1%
R91
100K
R92
100K
R93
100K
MN1E
C34
6,8µF
-5v
R95
10K
CR13
C21
0,1µF
R96
15K
R97
10 K
+5v
R98
15K
MN1E
R99
47 Ω
BREAK
CR12
+5V
C40
6,8µF
Q5
2N2222A
R71
6,8K
D
E
R70
10K
R72
1K
CR6
CR7
C15
6,8µF
R100
68K
MN4D
C25
6,8µF
+5v
R101
3,3K
R28
33K
C4
2,2nF
C3
2,2nF
MODU
R39
68K
MN1B
R40
15K
C5
4,7nF
R41
15K
C6
2,2nF
MN1B
R42
1,2 K
R44
10K
MN1B
R43
1,2 K
R45
10Ω
MICRO
2N2907
Q6
C39
6,8µF
-5V
MN 1E : TL 084 CN
MN 4D : "
MN 1B : "
smt
MODEM + K7 + MUSIQUE + REPROM
VERSION 3 31 10 80
A
B
C
D
3
4



smt

31 10 80 VERSION 3
MODEM + K7 + MUSIQUE + REPROM

H	G	F	E

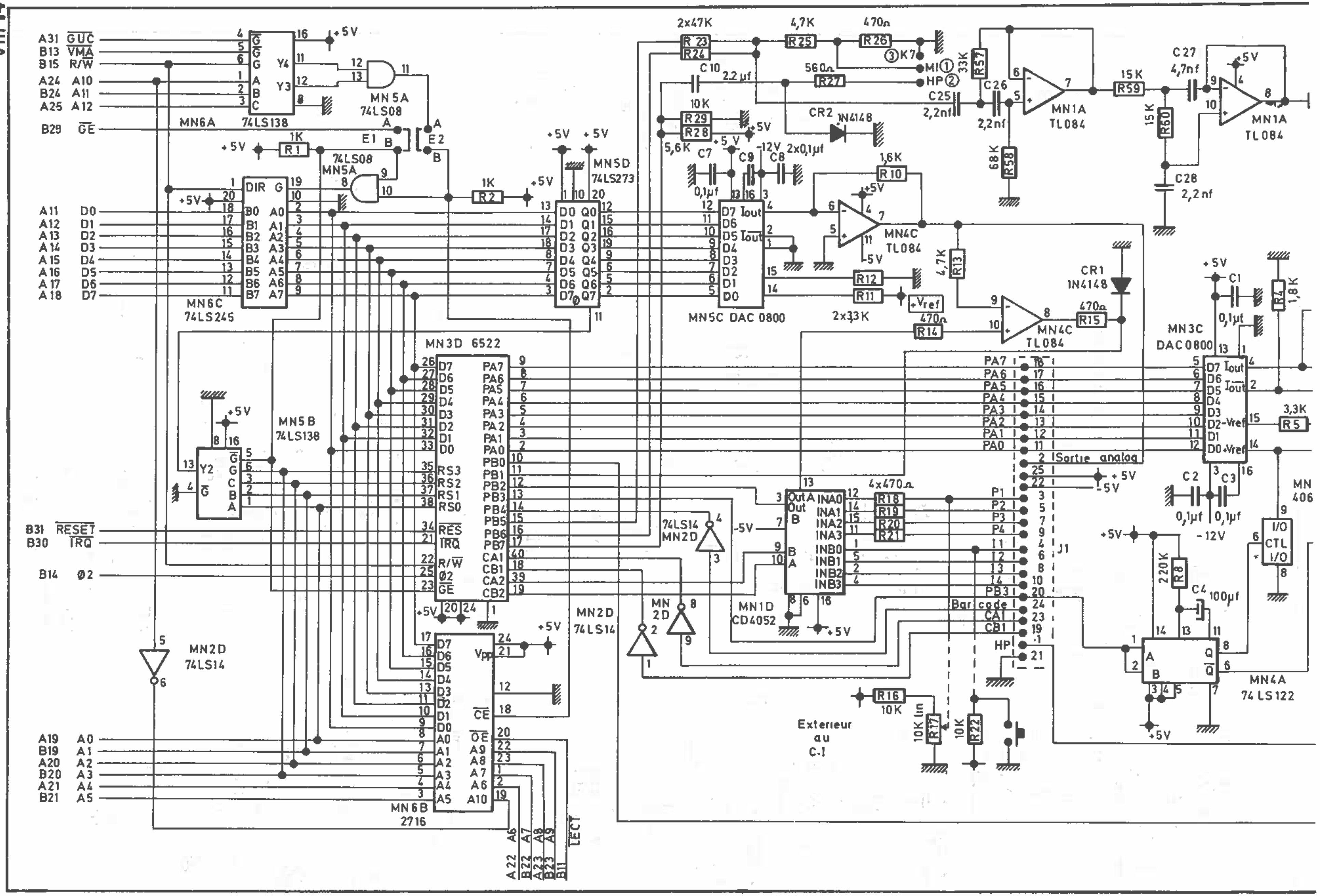


CARTE MODEM

VERSION 6

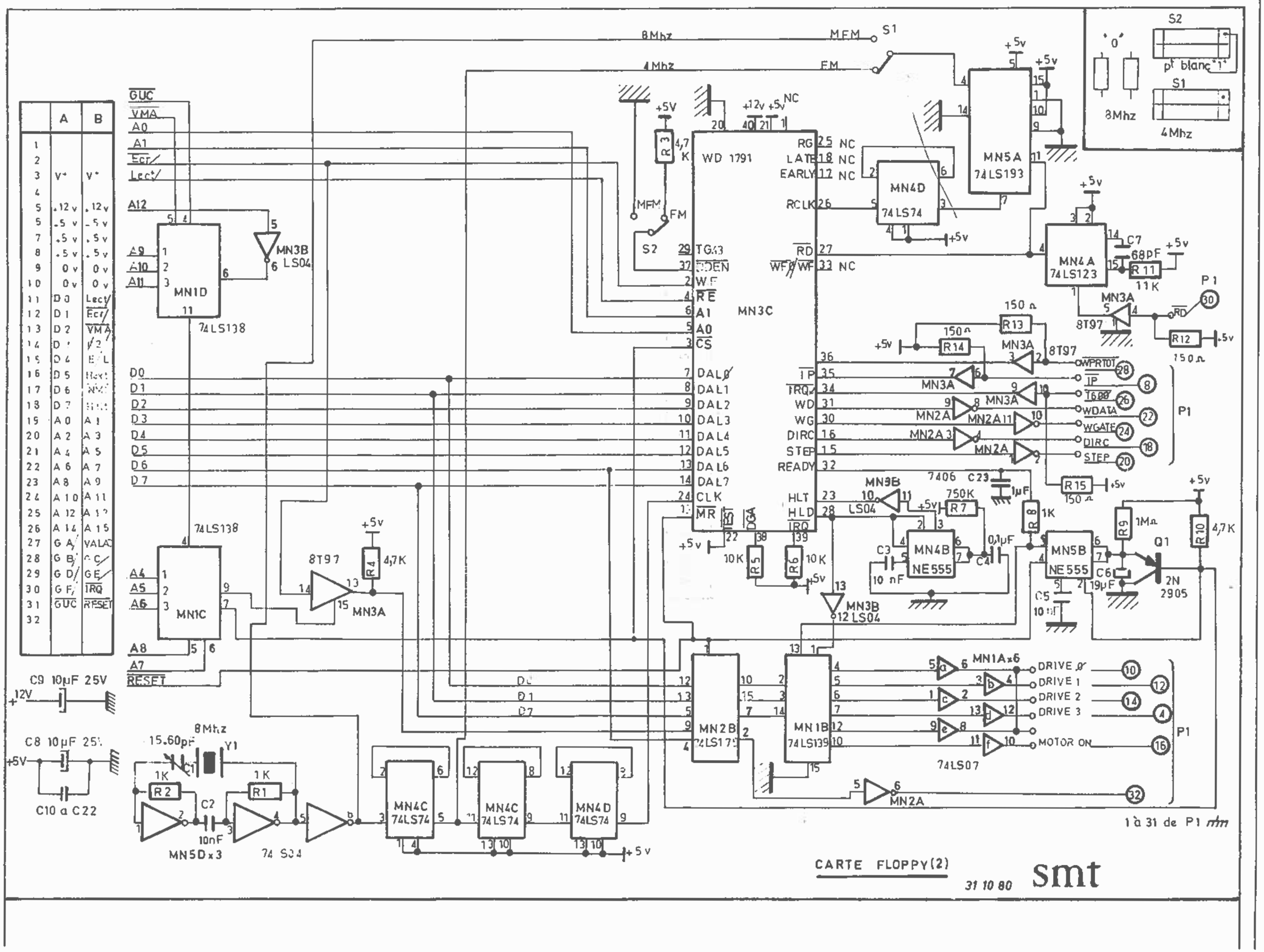
03 06 81

01 09 81



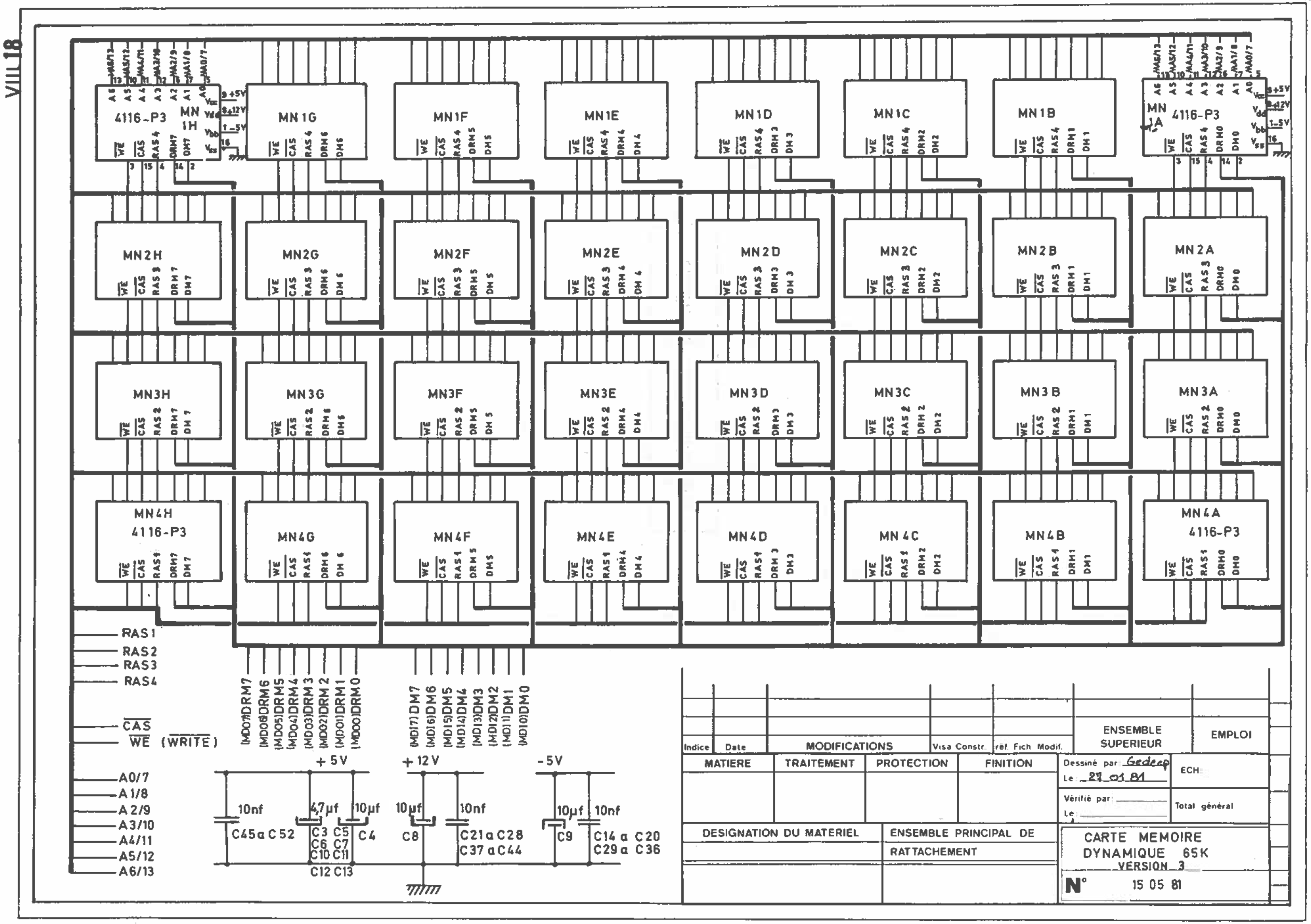
MN6A 74LS138
MN5A 74LS08
MN6C 74LS245
MN5D 74LS273
MN5C DAC 0800
MN4C TL084
MN1A TL084
MN3D 6522
MN5B 74LS138
MN2D 74LS14
MN6B 2716
MN1D CD4052
MN3C DAC0800
MN4A 74 LS122
A31 GUC
B13 VMA
B15 R/W
A24 A10
B24 A11
A25 A12
B29 GE
A11 D0
A12 D1
A13 D2
A14 D3
A15 D4
A16 D5
A17 D6
A18 D7
B31 RESET
B30 IRQ
B14 Ø2
A19 A0
B19 A1
A20 A2
B20 A3
A21 A4
B21 A5
A22 A6
B22 A7
A23 A8
B23 A9
B11 LECT
Sortie analog
Bar code
Exterieur au C-1
J1

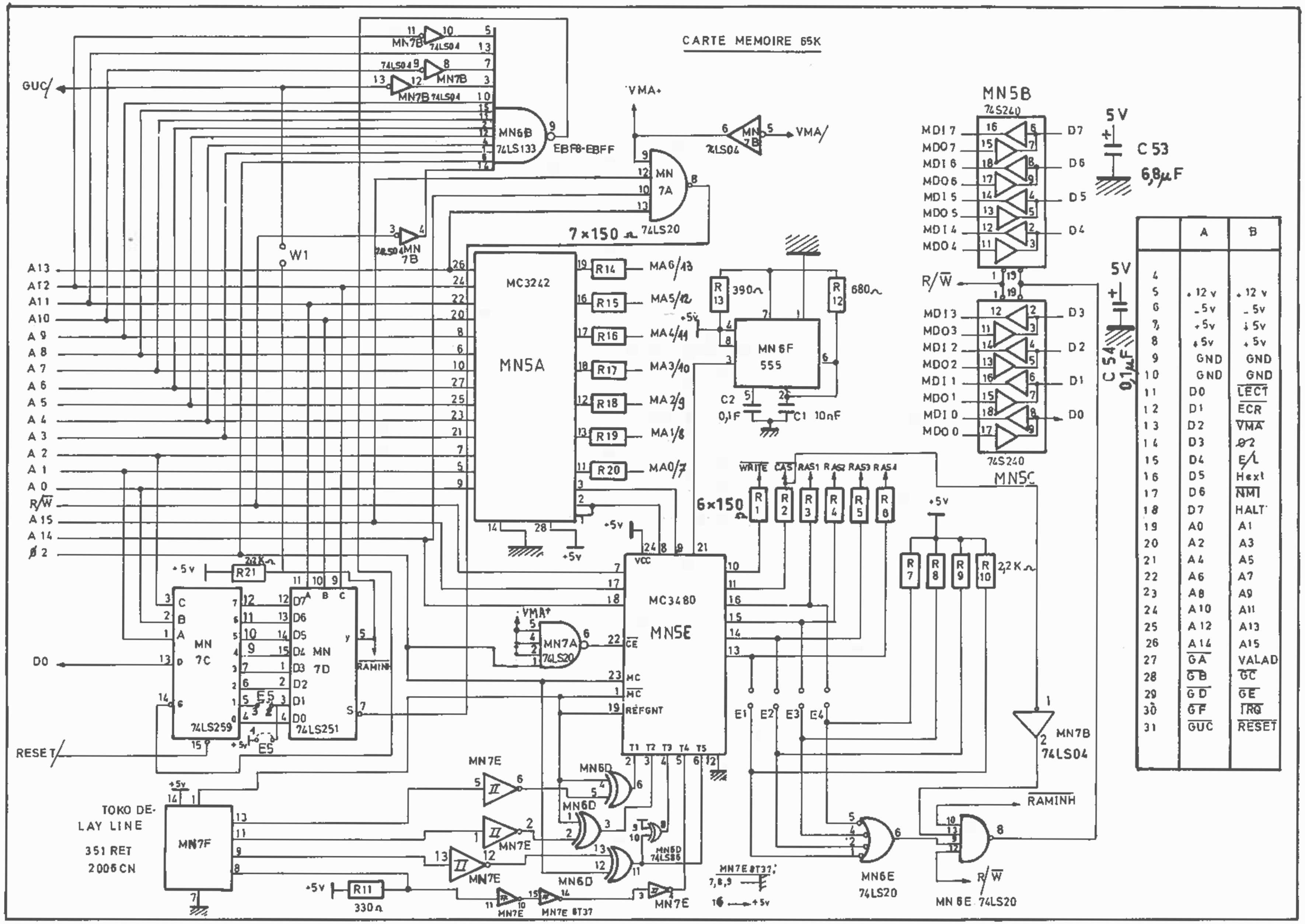
SCHEMAS DE LA CARTE FLOPPY



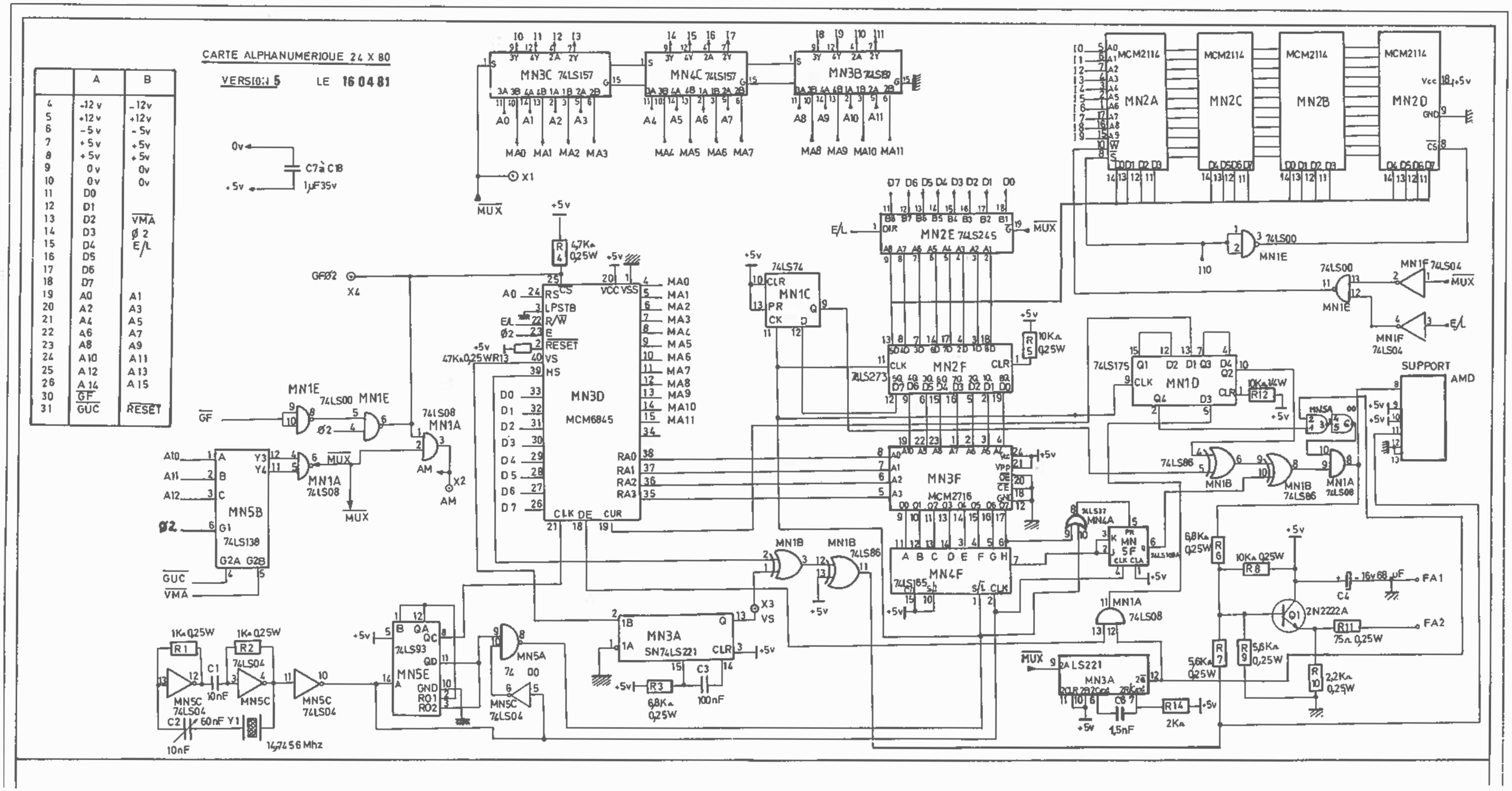
WD 1791
MN3C
MN5A 74LS193
MN4D 74LS74
MN4A 74LS123
MN1D 74LS138
MN1C 74LS138
MN3B LS04
MN3A 8T97
MN2A
MN4B NE555
MN5B NE555
MN2B 74LS175
MN1B 74LS139
MN1A x6 74LS07
MN4C 74LS74
MN4D 74LS74
MN5D x3 74 S04
8Mhz
4Mhz
MFM
FM
S1
S2
pt blanc "1"
DRIVE 0
DRIVE 1
DRIVE 2
DRIVE 3
MOTOR ON
P1
1 à 31 de P1
C9 10µF 25V
C8 10µF 25V
C10 à C22
2N 2905
CARTE FLOPPY (2)
31 10 80
smt

CARTE MEMOIRE DYNAMIQUE


VIII 18
MN 1H
4116-P3
MN1G
MN1F
MN1E
MN1D
MN1C
MN1B
MN 1A
4116-P3
MN2H
MN2G
MN2F
MN2E
MN2D
MN2C
MN2B
MN2A
MN3H
MN3G
MN3F
MN3E
MN3D
MN3C
MN3B
MN3A
MN4H
4116-P3
MN4G
MN4F
MN4E
MN4D
MN4C
MN4B
MN4A
4116-P3
WE
CAS
RAS1
RAS2
RAS3
RAS4
CAS
WE (WRITE)
A0/7
A1/8
A2/9
A3/10
A4/11
A5/12
A6/13
(MD07)DRM7
(MD06)DRM6
(MD05)DRM5
(MD04)DRM4
(MD03)DRM3
(MD02)DRM2
(MD01)DRM1
(MD00)DRM0
(MD17)DM7
(MD16)DM6
(MD15)DM5
(MD14)DM4
(MD13)DM3
(MD12)DM2
(MD11)DM1
(MD10)DM0
+5V
+12V
-5V
10nf
C45 a C52
4,7µf
C3 C5
C6 C7
C10 C11
C12 C13
10µf
C4
10µf
C8
10nf
C21 a C28
C37 a C44
10µf
C9
10nf
C14 a C20
C29 a C36
Indice
Date
MODIFICATIONS
Visa Constr.
réf. Fich. Modif.
ENSEMBLE SUPERIEUR
EMPLOI
MATIERE
TRAITEMENT
PROTECTION
FINITION
Dessiné par Gedeep
Le 27 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


CARTE MEMOIRE 65K
MN5A
MC3242
MN5E
MC3480
MN5B
74S240
MN5C
MN6B
74LS133
EBF8-EBFF
MN 7A
74LS20
MN 6F
555
MN 7C
74LS259
MN 7D
74LS251
MN7F
TOKO DELAY LINE
351 RET
2006 CN
MN6E
MN6D
74LS86
MN7E
MN7B
74LS04
VMA+
VMA/
GUC/
RESET
RAMINH
R/W
W1
7×150
6×150
C 53
6,8µF
C 54
0,1µF
5V
A
B

CARTE 24 x 80


CARTE ALPHANUMERIQUE 24 X 80
VERSION 5 LE 160481
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
MN5E 74LS93
MN3A SN74LS221
SUPPORT AMD
2N2222A
14,7456 Mhz
C7 à C18 1µF35v
MUX
FA1
FA2

Imprimé en France. — Imprimerie JOUVE, 18, rue Saint-Denis, 75001 PARIS