

GEBRUIKERSHANDLEIDING
KLOKHUIS - VIERLINGKAART

G E B R U I K E R S H A N D L E I D I N G
= = = = =

K L O K H U I S - V I E R L I N G K A A R T
= = = = =

Versie 5.0
Datum 850607

COLOFON

Klokhuis Vierlingkaart uitgebracht door:

Stichting Klokhuis,
Postbus 427,
3200 AK Spijkenisse.

Bestuur Stichting Klokhuis:
Gerard van der Woude (Voorzitter)
Ted Mos (Secretaris)
Dirk van der Waal (Penningmeester)
Jan-Willem Oomen (Produkten-coördinator)
Arthur van Riet (Distributie-coördinator)

Medewerkers aan het Vierlingkaart-project:

Johan van Domselaar:
Ontwikkelen van het besturingsprogramma (firmware)
en de applicatieprogramma's.

Eric van der Meer:
Schrijven van de gebruikershandleiding.

Peter van Rooyen:
Ontwerpen van de hardware.

Jan-Willem Oomen:
Algemene coördinatie en projectadministratie.

D & P Electronics, Staten Bolwerk 20, 2011 ML Haarlem:
CAD-ontwerp print en produktie eerste serie.

Alevo, Delft:
Drukwerk.

De Stichting Klokhuis heeft de hardware en de software voor de Vierlingkaart met de grootste zorg ontworpen en gecontroleerd. De Stichting Klokhuis kan echter op generlei wijze aansprakelijk worden gesteld voor schade, ontstaan door het gebruik van de Klokhuis-Vierlingkaart.

Inhoudsopgave

Over deze handleiding

| | | |
|--------|---|------------|
| 1 | Inleiding | 1-1 |
| 2 | Installatie van de Vierlingkaart Onderhoud | 2-1 2-4 |
| 3 | Principe van de werking | 3-1 |
| 4 | Gebruik in BASIC | 4-1 |
| 4.0 | BSB-0 - Programmakiezer | 4-3 |
| 4.1 | BSB-1 - Parallele printer | 4-5 |
| 4.2 | BSB-2 - Seriele printer | 4-8 |
| 4.3 | BSB-3 - Terminal | 4-10 |
| 4.4 | BSB-4 - Klok uitlezen | 4-12 |
| 4.5 | BSB-5 - Klok gelijkzetten | 4-14 |
| 4.6 | EPROMs | 4-15 |
| 5 | Gebruik zonder DOS | 5-1 |
| 6 | Gebruik met ProDOS | 6-1 |
| 7 | Gebruik vanuit Pascal | 7-1 |
| 8 | Gebruik onder CP/M | 8-1 |
| 9 | Gebruik in machinetaal | 9-1 |
| 9.1 | Bankswitching | 9-2 |
| 9.2 | Interrupts | 9-10 |
| 9.3 | De functies | 9-17 |
| 9.3.1 | De VIA | 9-17 |
| 9.3.2 | De ACIA | 9-23 |
| 9.3.3 | De klok | 9-31 |
| 9.3.4 | EPROMs | 9-34 |
| 10 | De Hardware | 10-1 |
| 10.1 | De DIP-schakelaars | 10-1 |
| 10.1.1 | SW1: BSB-instelling en klokinterrupts | 10-1 |
| 10.1.2 | SW4: EPROM-type en snelheid | 10-3 |
| 10.1.3 | SW7: Interrupts | 10-4 |
| 10.2 | Connectoren | 10-6 |
| 10.2.1 | De batterij-connector | 10-6 |
| 10.2.2 | De VIA-connector | 10-7 |
| 10.2.3 | De ACIA-connector | 10-8 |
| 10.3 | Doorverbindingen | 10-9 |

| | |
|---|-------------------|
| A | Probleemformulier |
| B | Geheugenindeling |
| C | Schema |
| D | Sourcelistings |
| E | Beperkingen |
| F | Datasheets |

Paklijst:

- . Vierlingkaart
- . Connector met snoer voor de parallele poort
- . Connector met snoer en steker voor de seriele poort
- . Batterijhouder met snoer en steker
- . 3 oplaadbare batterijen
- . Diskette met software
- . 2 Gebruikershandleidingen

Klokhuis Vierlingkaart

Over deze handleiding

=====

Dit is de gebruiksaanwijzing voor de Klokhuis-Vierlingkaart voor de Apple][-computer. Achtereenvolgens komen de volgende onderwerpen aan de orde:

- * Een overzicht van de functies die in de kaart zijn opgenomen, staat in hoofdstuk 1.
- * Het installeren van de kaart in de computer staat beschreven in hoofdstuk 2.
- * Hoe de kaart vier functies kan herbergen, en hoe daaruit wordt gekozen, wordt uitgelegd in hoofdstuk 3.
- * Het gebruik van de ingebouwde functies in een BASIC-programma wordt besproken in hoofdstuk 4.
- * De Vierlingkaart kan worden gebruikt in samenhang met de diverse Operating Systems die op de Apple bruikbaar zijn, nl. DOS 3.3, ProDOS, Pascal en CP/M. De hoofdstukken 5 t/m 8 gaan hier nader op in.
- * Instructies voor diegenen die zelf m.b.v. 6502-machinetaal nieuwe functies in de kaart willen programmeren, of hardware-uitbreidingen willen aansluiten, worden gegeven in hoofdstuk 9.
- * Details over de hardware, nl. de DIP-schakelaars en de diverse connectoren, staan in hoofdstuk 10.
- * De Aanshangsels geven de details van de gebruikte IC's, en een afdruk van de meegeleverde software, en een formulier waarmee problemen en vragen kunnen worden aangemeld.

Klokhuis Vierlingkaart

1 Inleiding

=====

De Vierlingkaart vervangt vier verschillende kaarten door een enkele kaart. Deze kaart bezet (ook vanuit een programma gezien) maar een enkele slot van de computer.

De Vierlingkaart bevat:

- 1) een klok;
- 2) twee parallelle in- of uitgangspoorten;
- 3) een seriele in- en uitgangspoort;
- 4) geheugen, opgedeeld in:
 - ruimte voor gebruikersprogramma's in EPROM of RAM (maximaal 40 kilobytes);
 - 1 kilobyte aan werkgeheugen (RAM).

De klok houdt de tijd (uren, minuten en seconden), de datum (jaar, maand en dag), en de dag van de week bij. De klok kan interrupts genereren naar de 6502-microprocessor, en wel per uur, minuut, seconde of milli-seconde.

Met de kaart wordt een houder met oplaadbare batterijen meegeleverd; hiermee blijft de klok doorlopen als de computer is uitgeschakeld.

De twee parallelle in- en uitgangen zijn aanwezig in de vorm van een zgn. VIA, die verder ook nog twee tellers en vier stuurlijnen herbergt. Ook dit IC kan interrupts genereren.

De seriele in- en uitgang is aanwezig in de vorm van een zgn. ACIA, die ook de stuursignalen kan leveren t.b.v. de communicatie met een modem. Dit IC kan eveneens interrupts veroorzaken.

Het besturingsprogramma voor de Vierlingkaart ondersteunt het gebruik ervan in programma's geschreven in BASIC of machinetaal, en in principe ook programma's geschreven in Pascal of gebruikmakend van het CP/M-systeem. Het besturingsprogramma kan 8K bytes groot worden, en staat in een EPROM.

Klokhuis Vierlingkaart

Er zijn vijf IC-voetjes aanwezig waarin machine-taalprogramma's kunnen worden geplaatst. Elk voetje kan een EPROM van 2, 4, of 8 K bytes bergen. In plaats van EPROMs is het ook mogelijk, RAMs van 2K bytes te plaatsen.

Op de kaart is 1K bytes aan RAM aanwezig. Dit fungeert als werkgeheugen voor de besturingssoftware, maar staat ook ter beschikking van de gebruiker. Hierdoor wordt er minder snel een beroep op het hoofdgeheugen van de Apple gedaan.

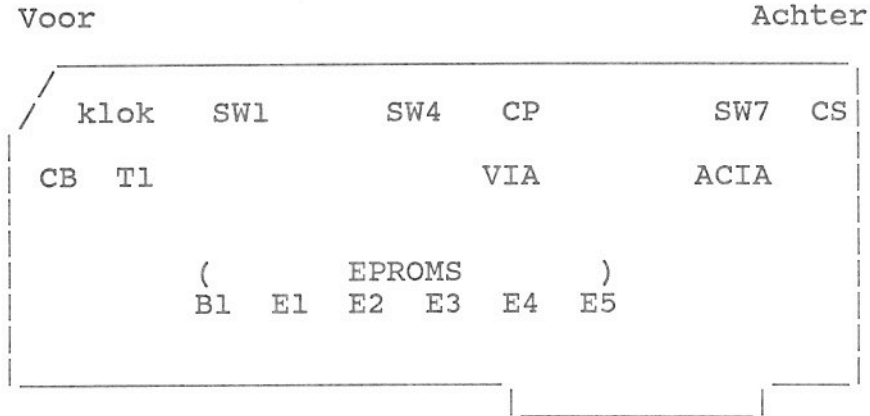
De kaart is voorzien van connectoren voor de parallele en seriele poorten, en voor de batterij.

De Vierlingkaart is te gebruiken in een computer van de Apple][-familie. Hiertoe behoren de Apple][,][Plus en //e, alsmede de ITT 2020 en de CHE. De computer mag zowel met Applesoft als met Integer BASIC zijn uitgerust.

Klokhuis Vierlingkaart

2 Installatie van de Vierlingkaart

De Vierlingkaart bestaat uit een printplaat met daarop een dertigtal IC's. Houden we de kaart met de afgeschuinde kant linksboven, dan zien we rechtsonder de connector voor de Apple.



Figuur 2.1

Langs de bovenrand zitten connectoren voor randapparaten (CP en CS) en het batterijblok van de klok (CB), en drie blokken met zgn. DIP-schakelaars (SW1, SW4 en SW7). Met trimmer T1 kan de snelheid van de klok worden bijgesteld. In voetje B1 zit een EPROM met de Besturingssoftware, en in voetje E1 t/m E5 kunnen EPROMS worden gestoken met gebruikerssoftware. De DIP-schakelaars beïnvloeden de verbinding van de onderdelen van de kaart. De precieze functie ervan wordt besproken in hoofdstuk 10. Op dit moment is het voldoende, ze in te stellen volgens het patroon:

SW1 SW4 SW7
 ----1111 ----- -----

Een '1' geeft aan dat een schakelaar in de ON-stand staat.

aflevering: --1-1111 -----111 -1-----1

Klokhuis Vierlingkaart

De Vierlingkaart moet in één van de slots van de Apple worden gestoken. Alvorens dat te doen, moeten we eerst de COMPUTER UITSCHAKELEN. Doen we dat niet, dan is beschadiging van de computer en de kaart vrijwel zeker.

Verwijder nu van de Apple de kap door deze aan de achterkant omhoog te trekken. Til de kap van de computer af en leg hem opzij. Achter op het moederbord van de Apple zit een rij connectoren, de slots. De Apple][(Plus) heeft er acht, de //e zeven. De meest rechtse slot heeft nummer 7. De Vierlingkaart kan in elk van de slots worden gestoken, behalve slot 0. Ook past de kaart niet in slot 3 van een Apple //e indien er een 80-kolomskaart is gemonteerd. Tevens werkt de kaart niet goed in slot 2 als in slot 3 een Extended 80-kolomskaart zit.

In de loop der jaren is het gewoonte geworden, de keuze van het slotnummer te laten hangen van de functie van de kaart. De conventie is:

| <u>Slot</u> | <u>Kaart</u> |
|-------------|---|
| 0 | RAM- of ROM-kaart |
| 1 | Printer-interface (parallel of serieel) |
| 2 | Communicatie-interface (bv. Modem) |
| 3 | 80-kolomskaart |
| 4 | Klokkaart |
| 5 | Vrij |
| 6 | Disk-interface |
| 7 | PAL-kleurenkaart |

De Vierlingkaart is tegelijk printerinterface, communicatieinterface en klokkaart, en kan dus zowel in slot 1 of 2 als in slot 4. Aan te raden is, een slotnummer te kiezen dat past bij het meest voorkomende gebruik van de kaart.

De BASIC-voorbeelden in hoofdstuk 4 gaan ervan uit dat de kaart voornamelijk als klok gebruikt gaat worden, en dus in slot 4 hoort. Veel programma's staan er overigens op dat een printer-interface in slot 1 zit.

Klokhuis Vierlingkaart

De batterijvoeding van de klok moet nu nog worden aangesloten. De drie meegeleverde oplaadbare batterijen worden in de batterijhouder geplaatst. Het is van groot belang dat dit op de juiste manier gebeurt. Op de batterijen staat vermeld, wat de plus (+) en wat de min (-) is. De min-kant komt op de veertjes in de houder te zitten, de plus-kant tegen de kontaktlipjes. Als de batterijen correct in de houder zijn gemonteerd, zitten ze om en om: plus boven, min boven, plus boven.

De stekker aan het batterijsnoertje wordt gestoken op connector CB (zie figuur 2.1). Dit moet op de juiste manier gebeuren, namelijk zodanig dat elk pennetje in een gaatje zitten. Klopt dit niet, dan kan de klok worden beschadigd.

Neem nu de batterijhouder en plaats deze aan de rechterzijde naast het moederbord van de Apple. Pak de kaart en steek deze voorzichtig in de gekozen slot. (Staat uw Apple ECHT uit?) Denk eraan dat de kaart goed recht wordt gehouden daar deze anders niet goed in de slot glijdt. De kaart moet nu netjes horizontaal in de Apple zitten. Plaats de kap weer op de computer. Deze mag nu weer worden ingeschakeld.

Uw Apple is nu gereed om de Klokhuis Vierlingkaart te gebruiken.

Klokhuis Vierlingkaart

Onderhoud

Nieuwe oplaadbare batterijen leveren geen spanning af. Om ze zoveel lading te geven dat de klok ook met uitgeschakelde netspanning blijft lopen, moet de Apple minstens een uur aan blijven staan.

Bij aflevering is de klok op de Vierlingkaart zo goed mogelijk afgeregeld. Mocht echter blijken dat de klok voor of achter loopt, dan is kan dit met trimmer T1 (zie figuur 2.1) worden gecorrigeerd. Bij linksom draaien gaat de klok langzamer lopen, bij rechtsom draaien sneller. Het schroefje mag maximaal een kwart slag naar beide kanten worden verdraaid.

In de praktijk zal gelijkzetten vrijwel nooit nodig zijn.

Klokhuis Vierlingkaart

3 Principe van de werking

=====

In dit hoofdstuk wordt uitgelegd hoe de Vierlingkaart is opgebouwd, en hoe de functies ervan worden bereikt. Begrip hiervan is nodig om de kaart ten volle te kunnen benutten. In de rest van deze handleiding wordt steeds naar dit hoofdstuk verwezen.

De Vierlingkaart is een uitbreidingskaart voor de Apple waarop vier functies verenigd zijn. Hoe is dat mogelijk? Andere kaarten bevatten immers een printerinterface OF een klok, maar niet beide. Wel, om te beginnen biedt de Vierlingkaart niet alle functies op het zelfde moment. Als de klokfunctie is geselecteerd, is bv. de printer niet aanspreekbaar, en andersom.

Vergelijk de Apple maar eens met een postkantoor met zeven loketten (de slots). Er zijn loketten met een enkele functie, bv. afgifte van giro-pasjes. Bij zo'n loket is onmiddellijk duidelijk, wat de klant wenst. Andere bieden meer functies, bv. verkoop van postzegels en loten. Bij deze moet de klant eerst tegen de beambte zeggen of hij postzegels danwel loten wil kopen. Dit komt overeen met een diskinterface waaraan twee drives zijn aangesloten, en waarbij telkens het nummer van de gewenste drive moet worden opgegeven.

De Vierlingkaart biedt keuze uit vier functies:

1. Parallele poorten. Hiermee kunnen twee groepen van tien signalen de computer worden in- en uitgestuurd.
2. Seriele poorten. Hiermee kunnen gegevens de computer worden in- en uitgestuurd, en wel telkens met 1 bit tegelijk.
3. Klok. Deze houdt de datum en de dag van de week bij, en de tijd tot op seconden nauwkeurig.
4. EPROMs. Hierin kunnen programma's in 6502-machinetaal worden opgenomen. Op de plaats van EPROMs kunnen ook RAMs worden gemonteerd.

Verder bevat de kaart nog een stukje werkgeheugen.

Klokhuis Vierlingkaart

Nu is het allemaal goed en wel dat er een handvol IC's op de kaart zitten die met de buitenwereld kunnen praten, of uit zichzelf nuttige dingen doen, maar erg eenvoudig in het gebruik zijn ze niet. Zeker vanuit een BASIC-programma is het geen sinecure om bv. een seriële poort netjes tegen een modem te laten praten. Daarom is op de Vierlingkaart, zoals de gewoonte is op de meeste uitbreidingskaarten voor de Apple][, plaats ingeruimd voor Besturingssoftware. Hiermee worden diverse populaire toepassingen van bovengenoemde functies zeer eenvoudig, namelijk:

- Een interface voor een parallelle printer. Deze gebruikt een parallelle poort.
- Een interface voor een seriële printer. Deze gebruikt de seriële poort.
- Een programma waarmee de Apple wordt omgebouwd tot terminal. Dit gebruikt ook de seriële poort.
- Een interface om de klok mee uit te lezen.
- Een programma om de klok mee gelijk te zetten.
- Een hulpmiddel om een programma in een EPROM te starten.

Elk van deze programma's zit in een apart deel van de besturingssoftware. Zo'n deel heet een 'bank'.

Nu rijst de vraag, hoe we de kaart vertellen, welke functie (of bank dus) we wensen. Het is heel eenvoudig: dat zeggen we tegen de beambte. Maar natuurlijk zit er geen kaboutertje op de kaart te luisteren. We leven in het tijdperk van de electronica. De beambte zit in een van de IC's, en daar kunnen we alleen met een programma iets mee doen. De beambte heet trouwens officieel 'B-register' (met de B van bank, inderdaad).

Het B-register beslaat een plaats in het geheugen van de Apple. Wordt in die plaats het nummer van een bank gezet, dan wordt onmiddellijk (nou ja, het kost wel een hele microseconde) de functie van die bank actief. Dus: als in het B-register het nummer van de klok-interface wordt gezet, dan IS de Vierlingkaart vanaf dat moment een klok!

Klokhuis Vierlingkaart

En hoe, zult u vragen, vullen we het B-register? BASIC-programmeurs hebben daarvoor met het POKE-commando. Voor details kunnen zij terecht in hoofdstuk 4. Assembler-experts zullen meteen grijpen naar een store-instructie. Zij kunnen hun hart ophalen in hoofdstuk 9.

Rest alleen nog de vraag, welke functie er is geselecteerd als de Apple wordt aangezet. 'Geen functie' kan niet: het B-register bevat altijd een nummer. Welnu, bij het aanzetten van de computer (en ook na RESET) staat in het B-register het banknummer dat met SW1 is ingesteld. Wat is SW1 nou weer? Even opletten. Aan de bovenrand van de Vierlingkaart zitten, tussen wat andere onderdelen, drie blokjes met zgn. DIP-schakelaars. SW1 is hiervan de voorste d.w.z. die welke het dichtst bij het toetsenbord zit als de kaart in de computer zit. Er zitten acht schakelaartjes op, genummerd van 1 t/m 8. Schakelaar 1 t/m 4 hiervan worden gebruikt om het banknummer in te stellen. Hoe dat precies in z'n werk gaat, staat beschreven in hoofdstuk 10.

Om alles nog eenvoudiger te maken bevat één van de banken een programma om een bank te kiezen. Op die manier kunt u vanaf het toetsenbord de Vierlingkaart tot printerinterface ombouwen, of de klok gelijkzetten. Hoofdstuk 4 vertelt er alles over.

Samenvatting

De Vierlingkaart bezit vier functies. Deze functies worden gebruikt in standaard-toepassingen. Deze toepassingen zijn verwezenlijkt in de besturingssoftware. Elke toepassing bezet een 'bank' van de besturingssoftware, en wordt gekozen met behulp van het B-register. Dit wordt in BASIC gevuld met POKE, en in assembler met een store-instructie. Bij het aanzetten van de Apple staat in het B-register wat er in schakelaarblok SW1 is ingesteld.

Klokhuis Vierlingkaart

4 Gebruik in BASIC

=====

De Vierlingkaart is uitgerust met Besturingssoftware die het mogelijk maakt, de diverse functies vanuit BASIC te gebruiken.

De Besturingssoftware is opgebouwd uit 32 banken, genaamd BSB's (Besturings-Software-Bank). De eerste 16 banken worden aangeduid met BSB-0 t/m BSB-9 en BSB-A t/m BSB-F. Elke BSB is 256 bytes groot. Er zijn twee soorten BSB: primair en secundair. De primaire banken bevatten elk een toepassing van een van de Vierlingfuncties; ze zijn inschakelbaar vanuit gebruikersprogramma's. De secundaire banken zijn alleen inschakelbaar vanuit een andere BSB. Het tweede zestiental banken is altijd secundair.

Het instellen van de actieve BSB kan op diverse manieren gebeuren.

- 1) Met DIP-schakelaar SW1. Hiermee wordt ingesteld, welke van de eerste 16 banken actief is bij het aanzetten van de computer of na een RESET. De actieve bank bepaalt het 'gezicht' van de Vierlingkaart, bv. printerinterface of klokkaart.
- 2) Vanaf het toetsenbord. Hierbij wordt gebruik gemaakt van het programma in BSB-0; zie de hierna volgende beschrijving.
- 3) Vanuit een programma. De actieve bank wordt veranderd door het nummer ervan in het B-register van de Vierlingkaart te schrijven. In een BASIC-programma gaat dit met de POKE-instructie.

| |
|---|
| Dit nummer moet worden verhoogd met 128. Dit geeft aan dat er vanuit een programma wordt geschakeld. Anders blijft de waarde gelden die met schakelaar SW1 was ingesteld. |
|---|

Klokhuis Vierlingkaart

Het B-register staat op byte 254 (hex \$FE) van de Besturingssoftware. Voorbeeld: als de Vierlingkaart in slot 4 zit, dan is het adres van het B-register gelijk aan \$C4FE ofwel 50430. In het algemeen is het adres van het B-register: $49406 + 256 * \text{SLOTNUMMER}$. Dus met:

```
POKE 50430,128+4
wordt BSB-4 ingesteld. Dat dit inderdaad is
gebeurd, is te controleren met:
```

```
PRINT PEEK(50430)
Dit levert de waarde 4 op (dus niet verhoogd
met 128).
```

Is er eenmaal een BSB gekozen, dan moet de kaart nog worden ingeschakeld. Dit gaat met de BASIC-commando's PR#n of IN#n, waarin 'n' het nummer is van de slot waar de kaart in zit.

Uitschakelen gebeurt met PR#0 of IN#0.

N.B. Deze commando's moeten op een speciale manier worden opgegeven in programma's die draaien onder (Pro)DOS.

De momenteel gedefinieerde primaire BSB's zijn:-

| BSB | Omschrijving |
|-----|-------------------|
| 0 | Programmakieler |
| 1 | Parallele printer |
| 2 | Seriele printer |
| 3 | Terminal |
| 4 | Klok uitlezen |
| 5 | Klok gelijkzetten |

De overige BSB's zijn secundair, en dus alleen voor intern gebruik door de primaire banken. Ze kunnen ook niet m.b.v. SW1 worden ingeschakeld.

Klokhuis Vierlingkaart

4.0 BSB-0 - Programmakiezer

=====

De programmakiezer wordt gebruikt voor het vanaf het toetsenbord activeren van een van de BSB's of een programma in een van de vijf gebruikers-EPROMs.

Als BSB-0 actief is, kan de programmakiezer worden gestart met IN#n. Als de Vierlingkaart in slot 4 zit, is dit dus IN#4.

Het commando PR#n is niet bruikbaar voor het activeren van de programmakiezer.

Er verschijnen hierna enkele vragen, zodat de volgende dialoog ontstaat.

1) Keuze uit EPROM of BSB

Vraag : GEBRUIKERSEEPROM STARTEN J/N?

| | |
|-----------|-----------------|
| Invoer: J | Ga naar stap 3. |
| N | Ga naar stap 2. |

2) Keuze van een BSB-bank

Vraag : GEEF NUMMER IN TE SCHAKELEN BANK 1-F?

Invoer: cijfer

Dit is het nummer van de in te schakelen bank van de Besturingssoftware. Alleen primaire BSB's (de eerste 16) kunnen op deze manier worden ingeschakeld. Voorbeeld: invoer van 1 schakelt de parallele printerinterface in.

De gekozen functie wordt hierna daadwerkelijk in gebruik genomen m.b.v. PR#n of IN#n.

Klokhuis Vierlingkaart

3) Starten van een Gebruikers-EPROM

Vraag : GEEF EPROM NUMMER(1-5)?

Invoer: cijfer

Dit is het nummer van de te starten EPROM. De voorste is nummer 1, de achterste 5. Zie Figuur 2.1.

Vraag : GEEF ENTRYPOINT NUMMER(1-9)?

Invoer: cijfer

Dit is het beginpunt binnen de gekozen EPROM.

Een enkele EPROM kan tot 8 K bytes aan 6502-instructies bevatten. Dat is zo veel dat er meer programma's in een EPROM passen. Deze programma's worden aangeduid met een zogenaamd 'Entry-pointnummer'. Een entrypoint is in dit verband een ingang in een EPROM, dus het beginpunt van één van de programma's daarin. Welke entrypoints voor een bepaalde EPROM zijn gedefinieerd, staat in de bijbehorende documentatie.

Het aantal entrypoints per EPROM is onbeperkt, maar met BSB-0 zijn alleen de eerste negen bereikbaar.

Klokhuis Vierlingkaart

4.1 BSB-1 - Parallele printer

=====

Na activeren van BSB-1 fungeert de Vierlingkaart als een interface voor een parallele printer. Zo'n printer kan met een parallele-printerkabel worden aangesloten op connector CP (zie fig. 2.1). De werking van de interface wordt beïnvloed door op zekere plaatsen in het geheugen een waarde te 'poken'. Het adres van die plaatsen hangt af van het slotnummer van de Vierlingkaart; dit wordt hieronder aangeduid met 'n'.

Regelbreedte | Adres 1784+n/\$6F8+n | Standaard 255

De interface houdt bij, hoeveel tekens er op een regel zijn geprint. Worden dit er meer dan de gegeven regelbreedte, dan wordt automatisch overgegaan op een nieuwe regel. De regelbreedte kan variëren van 1 t/m 254. Een waarde van 255 schakelt deze faciliteit uit; de linker marge wordt dan ook niet meer bijgehouden (zie beneden).

Bladhoogte | Adres 1272+n/\$4F8+n | Standaard 0

Bij het bedrukken van kettingformulieren bestaat de kans dat er tekst op de scheurrand terecht komt. Dit is te voorkomen door elk blad niet helemaal vol te zetten maar telkens een paar regels over te slaan. De printerinterface kan dit verzorgen als in de bladhoogte een waarde wordt gezet. De waarde moet 8 lager zijn dan het aantal regels.

Het maximaal aantal regels op een blad hangt af van de bladlengte en van de hoogte van elke regel. De meeste printers produceren 6 regels per inch. Er passen dus 66 regels op een blad van 11 inch, en 72 op een blad van 12 inch. De correcte waarde voor 'bladhoogte' is dan respectievelijk 58 en 64.

Klokhuis Vierlingkaart

De interface veroorzaakt de overgang naar het volgende blad door 7 blanco regels te versturen. Er wordt dus geen FF-code (Form Feed, decimaal 12) naar de printer gestuurd. Deze functie vervalt als er een bladhoogte van 0 is ingesteld.

Linkermarge | Adres 1144+n/\$478+n | Standaard 0

Elke regel die een printer afdrukt, begint in principe zo ver mogelijk naar links. Dat kan echter lastig zijn als de prints bijv. in een ringband moeten worden opgeborgen. Daarom kan de printerinterface automatisch een linkermarge bijhouden. Het aantal posities dat deze marge inneemt, wordt betrokken uit de Marge-byte. De maximale waarde voor de marge is 255 (hoewel dat waarschijnlijk nooit zal voorkomen).

Vlaggen | Adres 2040+n/\$7F8+n | Standaard 0

Er zijn nog drie stuurgegevens voor de printer mogelijk. Deze hebben de vorm van een vlag, d.w.z. ze kunnen 'aan' of 'uit' staan. De vlaggen zijn:

1) Video Standaard uit

In sommige omstandigheden kan het nuttig zijn, op het beeldscherm van de Apple te zien, welke tekens er naar de printer worden gestuurd. Dit wordt bereikt door de Videovlag aan te zetten. Hiermee wordt wel de regelbreedte tot 40 posities beperkt.

Klokhuis Vierlingkaart

2) Automatische Linefeed Standaard uit

De meeste printers gaan na het ontvangen van een CR-code (Carriage Return, decimaal 13) naar de linkerkant van het papier en tegelijk naar de volgende regel. Er zijn echter ook printers die tevens een LF-code (Line Feed, decimaal 10) nodig hebben om naar de volgende regel te gaan. Ontbreekt deze, dan worden alle regels over elkaar geprint. Voor die printers kan de interface deze Line Feed verzorgen.

3) Formfeed-simulatie Standaard uit

Een FF-code (Form Feed, decimaal 12) heeft als functie, een overgang naar een nieuw blad te bewerkstelligen. Sommige printers reageren hier echter niet op. Om de uitvoer dan toch in bladzijden te verdelen, kan de formfeed-vlag worden aangezet. De interface produceert dan na ontvangst van een FF-code zoveel Linefeeds dat het einde van het blad (ingesteld met de Bladhoogte) bereikt is.

De vlaggen staan alle op het zelfde adres en worden ingesteld met de som van de codes voor de gewenste vlaggen.

| dec | hex | Effect |
|-----|------|----------------------|
| 32 | \$20 | Formfeed |
| 64 | \$40 | Linefeed |
| 96 | \$60 | Linefeed en formfeed |
| 128 | \$80 | Video |
| 160 | \$A0 | Video en formfeed |
| 192 | \$C0 | Video en linefeed |
| 224 | \$E0 | Alles |

Klokhuis Vierlingkaart

4.2 BSB-2 - Seriele printer

=====

Na activeren van BSB-2 fungeert de Vierling-kaart als een interface voor een seriële printer. Deze kan met een seriële printerkabel worden aangesloten op connector CS (fig. 2.1).

De manier waarop de tekens naar de printer gaan, is de volgende:

- 8 bits.
- 1 startbit.
- 2 stopbits.
- Geen pariteitsbit.
- Snelheid instelbaar met schakelaar SW4.

N.B. De aangesloten printer moet de tekens wel op deze manier kunnen accepteren. Blijkt dat niet zo te zijn, dan moet er iets aan de instelling van de printer worden veranderd. Hoe dit gaat, staat in de gebruiksaanwijzing.

De seriële poort heeft twee kanten, nl. voor in- en voor uitvoer. Via de uitvoerkant gaan de gegevens van de computer naar de printer; via de invoerkant kan de printer aangeven dat er even niets gezonden mag worden. Dit gaat met CONTROL-S (DC3, decimaal 19). Met CONTROL-Q (DC1, decimaal 17) wordt de lijn weer vrij gegeven. Dit zogenaamde XON/XOFF-protocol wordt door de interface ondersteund.

Voor de seriële printer zijn de zelfde stuurmogelijkheden beschikbaar als voor de parallele, dus:

| | adres | | stand. |
|--------------|--------|---------|--------|
| Regelbreedte | 1784+n | \$6F8+n | 255 |
| Bladhoogte | 1272+n | \$4F8+n | 0 |
| Linkermarge | 1144+n | \$478+n | 0 |
| Vlaggen | 2040+n | \$7F8+n | 0 |

Klokhuis Vierlingkaart

Daarenboven is er de volgende instelling:

Lijnsnelheid | Geen adres | Standaard: SW4

Elke seriële lijn werkt met een bepaalde snelheid. Deze kan variëren van 150 tot 19200 bits per seconde. Bij inschakelen van de Vierlingkaart als seriële printerinterface geldt de snelheid die met schakelaar SW4 is ingesteld. Zie hoofdstuk 10 voor een beschrijving daarvan.

Klokhuis Vierlingkaart

4.3 BSB-3 - Terminal

=====

Deze bank maakt m.b.v. de seriële poorten een terminal van de Apple. In het algemeen zal de Vierlingkaart daarvoor verbonden worden met een 'modem' die op zijn beurt is verbonden met het telefoonnet. Aan de andere kant van de lijn bevindt zich dan een andere computer.

Het is een 'dumme' terminal: er zijn bijv. geen mogelijkheden, tekst op schijf op te slaan. Alles wat op het toetsenbord wordt ingetikt, gaat regelrecht de lijn op, en alles wat van buiten komt, verschijnt op het scherm. Dit is 40 kolommen breed; 80 is niet mogelijk.

Initialiseren

Direct na het inschakelen moet de terminal worden ingesteld. De dialoog verloopt aldus:

```
VIERLINGKAART TERMINAL EMULATIE Vx.y
```

```
GEEF LIJNSNELHEID: 1=150 2=300 3=600  
4=1200 5=2400 6=4800 7=9600 8=19200 ?
```

Dit geeft de transportsnelheid aan in bits per seconde. De zendsnelheid is gelijk aan de ontvangsnelheid. De Apple is met dit programma dus niet bruikbaar als Viditel-terminal.

```
GEEF DATAFORMAAT: 1=7-EVEN 2=7-ODD  
3=8-NONE 4=8-EVEN 5=8-ODD
```

Dit geeft het aantal bits per teken aan (7 of 8) en de pariteit (even, oneven of geen).

```
GEEF DUPLEXMODE: H=HALF F=FULL ?
```

Dit geeft aan of de andere kant elk ontvangen teken terugstuurt. Is dat niet het geval (Half Duplex), dan zet het programma de ingetikte tekst zelf op het scherm. Komt de tekst dubbel op het scherm, dan werken we Full Duplex.

Klokhuis Vierlingkaart

Speciale tekens

Boven is gezegd dat alle tekens ongewijzigd worden uitgestuurd. Dit is niet helemaal waar: de volgende tekens (of combinaties daarvan) hebben een speciale betekenis.

DEL (decimaal 127) werkt hetzelfde als de pijl naar links (BS, decimaal 8), d.w.z. dat het teken links van de cursor verdwijnt.

CONTROL-S (decimaal 19) werkt net zoals bij de seriële interface, d.w.z. dat uitvoer NAAR de terminal wordt tegengehouden tot er een CONTROL-Q (decimaal 17) wordt ingetoetst. Komt CONTROL-S naar de terminal toe, dan kan er niets meer worden ingetoetst totdat een CONTROL-Q wordt ontvangen.

CONTROL-Q Zie CONTROL-S.

De volgende combinaties werken alleen vanaf het toetsenbord.

<ESC> Dit stelt de Escape-toets voor. Deze heeft een speciale functie:

<ESC> D Schakelt tussen Half en Full Duplex (zie Initialiseren).

<ESC> L Verstuurt na elke Carriage Return al dan niet een Line Feed.

<ESC> E Beeindigt het terminalprogramma en gaat terug naar BASIC.

<ESC> <ESC> Verzendt een <ESC>-teken.

<ESC> rest Elk ander teken na een <ESC> wordt (zonder <ESC>) verzonden.

Klokhuis Vierlingkaart

4.4 BSB-4 - Klok uitlezen

De klok op de Vierlingkaart biedt de mogelijkheid, op zeer eenvoudige wijze in een programma te beschikken over de tijd, datum, en dag van de week.

Het gebruik van de klok in een BASIC-programma gaat als volgt.

1. Onthoud de oude BSB.
2. Kies de klokfunctie (BSB-4).
3. Schakel de invoer om naar de Vierlingkaart.
4. Lees de tijd uit de Vierlingkaart (met INPUT).
5. Schakel de invoer terug.
6. Schakel de oude BSB weer in.

De tijd wordt geleverd in een string met als inhoud:

```
Positie 11111111
12345678901234567
-----
|uu;mm;ss jjmdd w|
-----
```

Hierin betekenen de letters achtereenvolgens:

| | | |
|----|-----------------|------------------|
| uu | Uur | (00-23) |
| mm | Minuten | (00-59) |
| ss | Seconden | (00-59) |
| jj | Jaar | (00-99) |
| mm | Maand | (01-12) |
| dd | Dag | (01-31) |
| w | Dag van de week | (0-6, 0=maandag) |

Na het lezen van de tijd moet het programma er weer voor zorgen dat de invoer goed worden omgeleid. Dit gaat gewoonlijk met IN#0.

Klokhuis Vierlingkaart

Voorbeeld

Het volgende programma leest voortdurend de tijd en laat deze op de bovenste regel van het scherm zien.

```
10 HOME: D$=CHR$(4): REM Code voor DOS
20 SV=4: REM Slotnummer van Vierlingkaart
30 BREG=49406 + 256*SV: REM Adres van Bregister
40 OB=PEEK(BREG): REM Bewaar oud banknummer
50 POKE BREG, 128+4: REM Klokrees-BSB aan
100 PRINT D$"IN#"SV: REM Invoer van Vierling
110 INPUT " ";T$: REM Lees de tijd
120 PRINT D$"IN#0": REM Invoer van toetsenbord
140 REM Toon de tijd midden bovenaan het scherm
150 VTAB 1: HTAB 9: PRINT T$
165 REM Stop als er een toets is aangeraakt
160 IF PEEK(-16384) < 128 THEN 100
170 POKE -16368,0: REM Vergeet de toets
175 REM Herstel de originele BSB
180 POKE BREG,128+OB
999 END
```

Dit programma wordt afgebroken met een willekeurige toets.

N.B. Op de bijgeleverde schijf staat VIND4LING.SUB, een BASIC-subroutine die het slotnummer van de Vierlingkaart bepaalt. Door daarvan gebruik te maken, hoeft dat nummer niet in het programma te worden ingebouwd.

N.B. In de "Gebruikershandleiding Vierlingkaartprogramma's" staat het KLOK-commando beschreven, een commando dat aan DOS 3.3 kan worden toegevoegd, en dat de klokgegevens op het scherm toont.

Klokhuis Vierlingkaart

4.5 BSB-5 - Klok gelijkzetten

Bank 5 dient voor het gelijkzetten van de klok. Dit is zelden nodig daar de Vierlingkaart is voorzien van batterijen. Hiermee blijft de klok doorlopen als de computer is uitgeschakeld.

Allereerst wordt BSB-5 ingeschakeld en daarna IN#n getikt ('n' is het slotnummer van de Vierlingkaart). De computer toont de klokgegevens nu in de vorm:

```
VIERLINGKAART KLOK ZETTEN Vx.y
```

```
UU:MM:SS JJMMDD W  
uu;mm;ss jjmdd w
```

Dit zijn achtereenvolgens uren, minuten, seconden, het jaar, de maand, dag, en weekdag. De weekdays zijn genummerd van 0 t/m 6; maandag is 0.

De gegevens kunnen nu worden aangepast met de volgende toetsen:

```
0-9   Hiermee wordt een getal aangepast.  
      Veranderen van de seconden heeft echter  
      geen zin daar deze altijd nul worden.  
<-   Deze pijl zet de cursor een plaats naar  
      links.  
RETURN Dit sluit de invoer af. Dit gebeurt  
      automatisch als het einde van de regel  
      wordt bereikt.  
rest  Deze laten staan wat er stond.
```

Als bijvoorbeeld het scherm toont:

```
17;13;47 850127 7
```

en de invoer bestaat uit 4 spaties, een 5, en RETURN, dan is het resultaat:

```
17;15;00 850127 7
```

Klokhuis Vierlingkaart

4.6 EPROMs

=====

Op de Vierlingkaart is plaats voor vijf vrij te programmeren EPROMs, genummerd van 1 t/m 5. In een EPROM staat een programma in 6502-machine-taal. Hoe dat er precies moet uitzien, staat beschreven Hoofdstuk 9.

Bij het activeren van een programma in een EPROM moet, behalve het nummer van de EPROM, ook het zgn. entrypoint worden opgegeven. Dit geeft het deel aan van het programma dat gewenst is. Als een EPROM-programma is afgelopen, dan keert het terug naar het punt waar het was geactiveerd.

Het activeren van een EPROM-programma vanaf het toetsenbord is beschreven bij BSB-0. Om een EPROM-programma vanuit een BASIC-programma te activeren, moet er een aantal dingen gebeuren:

- 1) Onthouden van het oude BSB-nummer.
- 2) Selecteren van de Programmakiezer (BSB-0).
- 3) Doorgeven van het nummer van de EPROM en het gewenste entrypoint.
- 4) Aanroepen van de EPROM-activeringsroutine.
- 5) Inschakelen van de oude BSB.

Om bijv. entrypoint 1 van EPROM 3 te activeren is het volgende nodig:

```
1010 CN00 = 49152 + SV*256: REM Vierling-adres
1020 BREG = CN00 + 254: REM B-register
1030 OB = PEEK(BREG): REM Onthoud oude BSB
1035 REM Kies de Programmakiezer (BSB-0):
1040 POKE BREG,128+0
1050 POKE 1400,3: REM Het EPROM-nummer
1060 POKE 1528,1: REM Het entrypointnummer
1070 CALL CN00+241: REM Start EPROM-programma
1080 POKE BREG,128+OB: REM Herstel oude BSB
```

Klokhuis Vierlingkaart

5 Gebruik zonder DOS

=====

De voorbeelden in het BASIC-hoofdstuk zijn er telkens vanuit gegaan dat er gebruik gemaakt werd van DOS (Disk Operating System). De redenatie hierachter was dat dit in de praktijk meestal het geval zal zijn. Mocht zich de noodzaak voordoen, de Vierlingkaart te gebruiken in een BASIC-programma zonder DOS, dan zijn er enkele eenvoudige aanpassingen nodig. Het verschil zit 'm in het omleiden van de invoer en uitvoer. Wat met DOS wordt geschreven als:

```
120 PRINT CHR$(4);"PR#1"
```

moet zonder DOS worden geschreven als:

```
120 PR#1
```

En wat met DOS:

```
345 PRINT CHR$(4);"IN#4"
```

heet, is zonder DOS:

```
345 IN#4
```

Let wel: deze verschillen hebben alleen betrekking op programma's. Vanaf het toetsenbord blijven we gewoon IN#4 en PR#1 tikken.

Klokhuis Vierlingkaart

6 Gebruik onder ProDOS

=====

ProDOS, het nieuwste Disk Operating System voor de Apple][, verschilt bij gebruik in BASIC niet erg van zijn voorganger, DOS 3.3. Er is echter een verschil dat bezitters van de Vierlingkaart zal interesseren, nl. het gebruik van een klokkaart.

Bij het maken en wijzigen van een file op een diskette noteert ProDOS automatisch de datum en het tijdstip waarop dat gebeurt. ProDOS haalt deze gegevens uit vaste plaatsen in het geheugen. Deze kunnen vanuit een programma worden gevuld, bijv. door het STARTUP-programma dat op de ProDOS-schijf staat. Ze blijven dan verder gelijk, d.w.z. de tijd wordt niet automatisch bijgehouden. Maar als ProDOS bij het opstarten ziet dat er een klokkaart aanwezig is, dan wordt elke keer de datum en tijd gebruikt die deze kaart levert.

Hoe ontdekt ProDOS dat een bepaalde kaart een klokkaart is? Eenvoudig: de klokkaart van het enige type dat ProDOS kent, heeft een besturingsprogramma dat op een paar plaatsen bekende waarden bevat. De Vierlingkaart kan op deze plaatsen heel andere waarden hebben, daar bij het opstarten best een andere functie geselecteerd kan zijn dan de klok; lees daar 'Principe van de werking' (hoofdstuk 3) nog maar eens op na.

Toch willen we wel graag de Vierlingklok door ProDOS laten gebruiken. Daar moeten we wel wat voor doen: als ProDOS niet zelf kan zien dat we een klokkaart hebben, vertellen we het hem gewoon.

De manier om ProDOS bekend te maken met het bestaan van een klokkaart is de volgende.

- Stap 1. Zoek de Vierlingkaart op.
- Stap 2. Selecteer de klokfunctie (BSB-4).
- Stap 3. Zet het adres van de klokuitleesroutine in de 'System Page' van ProDOS.

Klokhuis Vierlingkaart

Het opzoeken van de kaart kan het beste gebeuren in STARTUP, het programma dat automatisch wordt gedraaid als een ProDOS-schijf met BASIC wordt opgestart. Een aangepast STARTUP-programma wordt meegeleverd.

Voor het opzoeken van de Vierlingkaart in BASIC kan de zelfde methode worden gebruikt als die welke ProDOS toepast. Elke BSB bevat op enkele vaste plaatsen bekende waarden. We hoeven dus alleen maar achtereenvolgens alle slots onder de loep nemen, en na te gaan, welke daarvan op de juiste plaats de juiste codes bevat.

Een programma dat dit doet, kan er aldus uitzien.

```
4 D$=CHR$(4): REM Nodig voor ProDOS
10 GOSUB 1000
20 IF SV = 0 THEN 100: REM Geen Vierling
30 POKE CN+254,128+4: REM Klok aan
40 PRINT D$;"IN#";SV
50 INPUT " ";T$
60 PRINT D$;"IN#0"
70 PRINT "De tijd is ";T$
100 REM Rest van het programma...

1000 REM Subroutine die de klok zoekt.
1010 CN = 49152: REM Hex $C000
1015 REM Bekijk alle toepasselijke slots
1020 FOR SV=1 TO 7
1030 CN = CN + 256: REM Beginadres van slot SV:
           $C100, $C200, ... $C700
1060 REM Kijk of de codes kloppen
1061 IF PEEK( CN+247 ) <> 9 THEN 1090
1062 IF PEEK( CN+249 ) <> 145 THEN 1090
1063 IF PEEK( CN+251 ) <> 108 THEN 1090
1080 RETURN: REM Gevonden; BSB-4 staat al aan
1084 REM           SV bevat het slotnummer,
1085 REM           CN het slotadres
1090 NEXT SV
1099 SV=0: RETURN: REM Niet gevonden
```

Klokhuis Vierlingkaart

N.B. Een BASIC-subroutine die bovenstaande bewerkingen uitvoert, staat onder de naam VIND4LING.SUB op de bijgeleverde schijf. Daarop staat tevens een STARTUP-programma dat de Vierling-klok aan ProDOS koppelt. Ten gevolge van een leemte in de documentatie van ProDOS is het niet mogelijk gebleken, dit op de meest elegante wijze te verwezenlijken. Er is nu gekozen voor een hulproutinetje in pagina 3. Het begin ligt omstreeks adres \$3B0; er is dus nog wat ruimte over.

LET OP! Er zijn BASIC-programma's die in pagina 3 machinetaalroutines zetten. Het is dus mogelijk dat deze programma's niet werken als de klok van Vierlingkaart aan ProDOS is gekoppeld.

Klokhuis Vierlingkaart

7 Gebruik vanuit Pascal

=====

Het gebruik van de Vierlingkaart vanuit Pascal is in principe mogelijk maar wordt op dit moment nog niet ondersteund.

Klokhuis Vierlingkaart

8 Gebruik onder CP/M

=====

Het gebruik van de Vierlingkaart onder het CP/M Operating System is in principe mogelijk maar wordt op dit moment nog niet ondersteund.

Klokhuis Vierlingkaart

9 Gebruik in Machinetaal

=====

Dit hoofdstuk is bestemd voor diegenen die de Vierlingkaart willen gebruiken vanuit machinetaalprogramma's. Zulke programma's kunnen in het RAM-geheugen van de Apple staan, maar ook in een EPROM die in een voetje op de Vierlingkaart wordt gestoken.

De onderwerpen die aan de orde komen, zijn:

- Bankswitching - het gebruiken van de beperkte adresruimte van de 6502 om toch een groot aantal geheugenplaatsen te bereiken.
- Interrupts - signalen die door de diverse functies van de kaart op onvoorspelbare momenten kunnen worden opgewekt om de aandacht te vragen voor iets dat zij hebben waargenomen.
- Registers - speciale geheugenposities op de kaart die de werking ervan besturen. Hiertoe behoren:

- B-register - BSB-selectie
- C-register - Functie-keuze
- K-register - Klokbesturing
- D-register - Klokgegevens
- S-register - ACIA-snelheid

Om ten volle profijt van dit hoofdstuk te kunnen trekken, kan enige kennis van programmeren in 6502-assembleertaal nuttig zijn.

Klokhuis Vierlingkaart

9.1 Bankswitching

Centraal in de opzet van de Vierlingkaart staat een techniek genaamd 'bank switching'. Dit is een methode om een groter geheugen te kunnen gebruiken dan er met directe adressering bereikbaar is.

Geheugenindeling

Een probleem met 8-bits microprocessors is dat ze 'maar' 65536 bytes kunnen adresseren. Dit is namelijk het grootste aantal verschillende adressen dat met 16 bits is te maken: 2 tot de macht 16. Dit wordt afgekort tot 64 K (64 x 1024 = 65536). Het geheugen van een microprocessor ziet er dus zo uit:

Adres

\$0000 eerste byte
\$...
\$FFFF laatste byte

| |
|--|
| De \$ geeft aan dat het adres als hexadecimaal getal is genoteerd. |
|--|

De geheugenindeling van de Apple wordt gedeeltematig bepaald door de 6502, en ziet er aldus uit:

Adres Gebruik

| | |
|--------|---|
| \$0000 | Page 0 |
| \$0100 | Stack |
| \$.... | Text- & Hires-pagina's, Basic-programma en DOS. |
| \$C000 | I/O-adressen (4096 bytes) |
| \$D000 | ROM (3 x 4096 bytes) |
| \$FFFF | laatste byte |

In een 48K-systeem bestaat het geheugen van adres \$0000 t/m \$BFFF uit RAM.

Via de I/O-adressen communiceert de Apple met de buitenwereld.

De 12K bytes vanaf \$D000 bevatten ROM met Basic en de Monitor.

Dit alles bij elkaar is dus 64 K bytes.

Klokhuis Vierlingkaart

De RAM-kaart

Er bestaat echter ook nog zo iets als de 16 K RAM-kaart. Deze is o.a. nodig om de Apple genoeg geheugen te geven voor het Pascalsysteem. Rijst de vraag, hoe de 6502 dit geheugen adresseert. Welnu, dat gaat dus met bank switching. Het geheugen op de RAM-kaart heeft de adressen \$D000 t/m \$FFFF toegewezen gekregen. De geheugen-indeling is dus:

| Adres | Gebruik | Alternatief | Alternatief 2 |
|--------|---------|-------------|------------------|
| \$0000 | RAM | | |
| \$C000 | I/O | | |
| \$D000 | ROM | RAM-kaart | RAM-kaart, extra |
| \$E000 | ROM | RAM-kaart | - |
| \$F000 | ROM | RAM-kaart | - |

De ingebouwde ROM en de RAM-kaart bezetten dezelfde adressen maar zitten elkaar niet in de weg omdat ze naar keuze kunnen worden in- en uitgeschakeld. Dit gebeurt met de 'soft switches', speciale adressen in het I/O-gedeelte. Ze heten 'soft' om ze te onderscheiden van 'harde' draai- en wipschakelaars. Door bv. gebruik te maken van de soft switch op adres \$C080 wordt de ROM uit-, en de RAM-kaart ingeschakeld.

De RAM-kaart bevat 16 K bytes; 12 K hiervan vinden plaats van \$D000 t/m \$FFFF. De resterende 4 K komen in het gebied van \$D000 t/m \$DFFF, dat hiermee dus driedubbel wordt gebruikt. (Plaatsing vanaf \$C000 is niet mogelijk omdat daar de soft switches zitten, zodat dan onder andere het omschakelen van RAM niet meer mogelijk zou zijn.)

Overigens: in de Apple //e is de extra 16K RAM reeds ingebouwd. Het heet daar Extension-RAM.

Klokhuis Vierlingkaart

Besturings-ROM

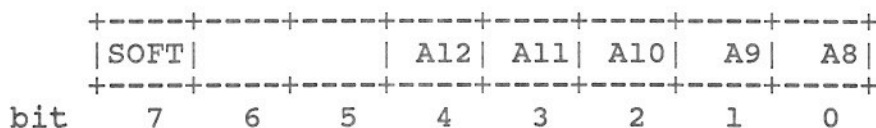
Een uitgebreide versie van de schakelmethode van de RAM-kaart wordt gebruikt in de Vierlingkaart. Elke uitbreidingskaart voor de Apple heeft een vast deel van het geheugen tot zijn beschikking. Het adres daarvan is kan worden afgeleid uit het nummer van de slot waar de kaart in zit, nl. vanaf \$Cn00. Een kaart in slot 5 kan bv. beschikken over de adressen \$C500 t/m \$C5FF.

In dit adresgebied zit op de meeste kaarten een ROM-geheugen. Op een diskinterface zit hier bv. het programma dat het allereerste begin van DOS van de diskette leest. Op de Vierlingkaart zit hier een EPROM met de Besturingssoftware. In hoofdstuk 3 is verteld dat de Besturingssoftware bestaat uit een aantal banken, genaamd BSB-0 t/m BSB-1F. Al deze banken zijn op hetzelfde adres te bereiken, dus vanaf \$Cn00. De keuze van de bank wordt bepaald door de inhoud van het B-register.

Het B-register

Het B-register beslaat een positie in het geheugen dat hoort bij de slot met de Vierlingkaart. Het is in feite byte 254 van de Besturings-software. Het zit dus zo: van \$Cn00 t/m \$CnFF zit een ROM met de Besturingssoftware voor de Vierlingkaart. Gewoonlijk heeft het totaal geen zin, naar een ROM te schrijven. Maar op adres \$CnFE zit een stukje RAM, en als daarin een banknummer wordt gezet, verandert de inhoud van de rest van het adresgebied.

De indeling van het B-register is:



Figuur 9.1

Klokhuis Vierlingkaart

- * SOFT geeft aan, op welke manier er een BSB moet worden gekozen.
 - Is SOFT 1, dan gebeurt dit met het B-register.
 - Is SOFT 0, dan gebeurt dit met schakelaar SW1.
- * A12 t/m A8 vormen samen het BSB-nummer. Schakelaars 1 t/m 4 van SW1 zijn verbonden met A8 t/m A11. Hiermee kunnen BSB-0 t/m BSB-F worden ingesteld.

Let wel: het vanuit een programma inschakelen van een BSB gebeurt door naar het B-register te SCHRIJVEN met SOFT=1. Het LEZEN van het B-register levert het nummer van de actieve BSB zonder het SOFT-bit.

In feite zit het B-register vast aan alle adressen van \$Cn00 t/m \$CnFE, maar de afspraak is dat alleen \$CnFE wordt gebruikt.

Extension-ROM

De oplettende lezertjes zal iets zijn opgevallen. Het hoogste slotnummer in een Apple is 7. Het hoogste adres van een ROM op een uitbreidingskaart is dus \$C7FF. Maar het I/O-deel van het geheugen loopt tot \$CFFF. Er zijn dus meer dan 2000 adressen over. Wees gerust, zo verspillend is de ontwerper niet geweest. Alle uitbreidingskaarten zijn gerechtigd, dit gebied te bezetten met een stuk ROM. Omdat dit een uitbreiding is van de 'normale' kaart-ROM, heet dit 'Extension ROM'. Het gebruik ervan is uiterst eenvoudig: een programma in de ROM in \$Cnxx springt gewoon naar een adres in het bereik \$C800 t/m \$CFFF. Er zit wel een voorwaarde aan: voordat dit gebeurt moet elke kaart ervoor zorgen dat de Extension ROM van alle andere kaarten is uitgeschakeld. Als er namelijk meer ROMs tegelijk actief zijn, reageren deze allemaal op een gegeven adres, en is het resultaat geheel onvoorspelbaar. Het uitschakelen van Extension ROM gebeurt door het aanspreken van het adres \$CFFF, bv. met een LDA-instructie. Omgekeerd houdt dit in dat elke kaart op dat CFFF-sigitaal moet reageren, en wel met het uitschakelen van zijn eigen ROM. De Vierlingkaart dus ook.

Klokhuis Vierlingkaart

Gebruikers-EPROMs

De Vierlingkaart kan Extension ROM bezetten met een programma in één van de gebruikers-EPROMs. Nu zitten er, zoals bekend, vijf EPROM-voetjes op de Vierlingkaart. Elk ervan kan de hele adresruimte van \$C800 t/m \$CFFF vullen. Er moet dus een keus worden gemaakt. Deze keus geschiedt met het zgn. C-register.

Verder kan elke EPROM 2K, 4K of 8K (resp. 2048, 4096 of 8192) bytes bevatten. De eerste maat past precies, de overige zijn veel te groot en vormen dus een probleem. Om dit op te lossen zijn grote EPROMs verdeeld in stukken van 2K. De keuze van het stuk dat in Extension ROM wordt geplaatst, geschiedt eveneens met het C-register.

Het C-register

Het C-register bezet, evenals het B-register, een plaats in het adresgebied dat hoort bij de slot met de Vierlingkaart. Het heeft adres \$CnFF. De indeling is:



Figuur 9.2

* Het CARD-bit moet op 1 staan om de Vierlingkaart aan te zetten.

* Bits 0 t/m 2 (selectie) bevatten het nummer van de EPROM. Hoe de EPROMs zijn genummerd, staat aangegeven in de volgende tabel.

Klokhuis Vierlingkaart

| 2 1 0 | Functie |
|-------|---------|
| 0 0 0 | Geen |
| 0 0 1 | EPROM 1 |
| 0 1 0 | EPROM 2 |
| 0 1 1 | EPROM 3 |
| 1 0 0 | EPROM 4 |
| 1 0 1 | EPROM 5 |
| 1 1 0 | Klok |
| 1 1 1 | ACIA |

N.B. De klok en de ACIA worden later besproken.

* Met HOOG en LAAG wordt ingesteld, welke 2K-deel van een grote EPROM gewenst is, nl:

| HOOG | LAAG | deel | adres | |
|------|------|------|-------|--------|
| 0 | 0 | 0 | 0000 | \$0000 |
| 0 | 1 | 1 | 2048 | \$0800 |
| 1 | 0 | 2 | 4096 | \$1000 |
| 1 | 1 | 3 | 7144 | \$1800 |

Bovendien moet het programma in de EPROM op de hoogte zijn van de omvang van die EPROM.

* RAM moet op 0 staan om een EPROM in te schakelen. Wat er gebeurt als RAM op 1 staat, wordt beschreven in de volgende paragraaf.

Wat er in het C-register is geschreven, kan ook weer worden teruggelezen.

Klokhuis Vierlingkaart

Vierling-RAM

Op de Vierlingkaart is plaats ingeruimd voor een hoeveelheid werkgeheugen, te weten 1024 bytes RAM. Deze 1K bytes zijn verdeeld in vier banken van 256 bytes; ze bezetten adressen \$CF00 t/m \$CFFE. Adres \$CFFF mag niet worden gebruikt omdat daarmee de Extension ROM (en dus ook de Vierling-RAM) wordt uitgeschakeld.

Het inschakelen van de RAM, en het instellen van de gewenste bank, geschiedt met het C-register. De gebruikte bits zijn de volgende (zie figuur 9.2):

- * RAM (bit 7) moet 1 zijn om een RAM-pagina te plaatsen vanaf \$CF00.
- * Bits 0 t/m 2 (selectie) stellen een EPROM-nummer voor. Van de geselecteerde bank van die EPROM zijn de bovenste 256 bytes onbruikbaar; daar zit RAM.
- * RAMH (bit 6) en HOOG (bit 5) geven de gewenste RAM-bank van 256 bytes aan:

| RAMH | HOOG | bank | adres | |
|------|------|------|-------|-------|
| 0 | 0 | 0 | 000 | \$000 |
| 0 | 1 | 1 | 256 | \$100 |
| 1 | 0 | 2 | 512 | \$200 |
| 1 | 1 | 3 | 768 | \$300 |

Conflicten

De ruimte op de Vierlingkaart is, hoe omvangrijk de kaart ook is, beperkt. Daardoor is het niet mogelijk gebleken, alle functies gelijktijdig toegankelijk te maken. Het komt erop neer dat er een keuze gemaakt zal moeten worden uit:

- Eén van de EPROMS, OF de klok, OF de ACIA. Hieruit volgt dat de klok en de ACIA niet direct bruikbaar zijn vanuit een programma in een gebruikers-EPROM. (De VIA overigens wel.)
- 2K aan gebruikers-EPROM, maar geen toegang tot RAM, OF 1,75K aan EPROM met 256 bytes RAM,

Klokhuis Vierlingkaart

waarbij nog moet worden aangetekend dat niet alle EPROM-banken toegang hebben tot alle RAM-banken.

De Vierling-RAM wordt ook gebruikt door de Besturingssoftware. Deze legt beslag op het gebied van \$CF00 t/m \$CF3F van RAM-bank 0.

9.2 Interrupts

Een interrupt is een tijdelijke onderbreking van de normale afloop van een programma, met als doel, te reageren op een gebeurtenis buiten de computer. Dit kan bv. zijn dat er een teken via een seriele poort is binnengekomen, dat een klok een heel uur heeft bereikt, of dat er iemand op een knop heeft gedrukt.

Hoewel dit hoofdstuk handelt over programmeren in machinetaal, is het wel aardig er op te wijzen dat BASIC ook een interrupt-mogelijkheid bezit. Een BASIC-programma namelijk dat is afgestopt met de CONTROL-C-toets, kan worden doorgestart met het CONT-commando. In de tussentijd kan de inhoud van diverse variabelen zijn bekeken (met PRINT) of zelfs veranderd.

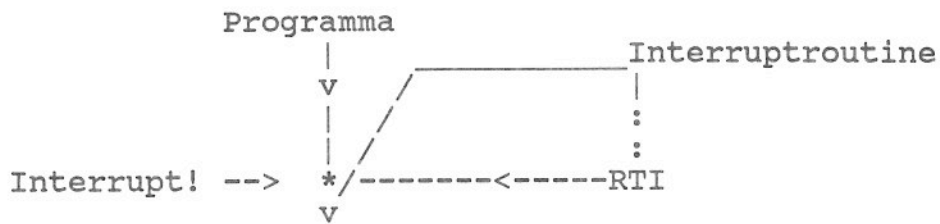
Machine-interrupts

Een interrupt op machineniveau treedt op als een van de interrupt-aansluitingen van de 6502-micro-processor een bepaald signaal krijgt. De 6502 houdt dan op met het uitvoeren van het programma waar hij mee bezig was, en begint een ander programma uit te voeren. Dit nieuwe programma reageert op de situatie die de interrupt heeft veroorzaakt, en staat bekend als een 'interruptroutine'.

Heeft de interruptroutine zijn taak volbracht, dan moet het onderbroken programma worden herstart. Hiervoor is een speciale instructie: RTI. Dit staat voor 'ReTurn from Interrupt'.

Het hele proces wordt schematisch voorgesteld in figuur 9.3.

Klokhuis Vierlingkaart



Figuur 9.3

Interruptssoorten

De 6502 kent twee soorten interrupt.

- NMI: Non-Maskable Interrupt. Deze treedt op als de spanning aan de NMI-pen van de 6502 van hoog naar laag gaat. Deze interrupt is, zoals de naam aangeeft, niet tegen te houden.
- IRQ: Interrupt ReQuest. Deze treedt op zolang de spanning aan de IRQ-pen van de 6502 nul is. Deze interrupt is wel tegen te houden, namelijk met de instructie SEI (SEt Interrupt). In een interrupt-routine zijn IRQ-interrupts automatisch geblokkeerd. Toelaten van IRQ-interrupts gaat met de instructie CLI (CLear Interrupt).

Interruptvectoren

Het starten van een interruptroutine wordt door de 6502 verzorgd. Het beginadres van zo'n routine is niet vast: het staat op een plaats in het geheugen die aan de 6502 bekend is. Zo'n geheugenplaats met een adres heet een 'vector', in dit geval 'interruptvector'.

De interruptvector voor de IRQ staat op adres \$FFFE/FFFF, die voor de NMI op adres \$FFFA/FFFB. Dit zijn adressen die bij het Monitor-programma horen. Omdat deze in een ROM staat en dus niet is te veranderen, verwijzen de vectoren naar plaatsen binnen de Monitor. Daarvandaan wordt dan gesprongen naar de werkelijke interrupt-routines.

Klokhuis Vierlingkaart

De adressen hiervan staan in RAM en kunnen dus naar believen worden veranderd.

De 'zachte' vector voor de NMI-interrupt staat op adres \$3FC/3FD, die voor de IRQ-interrupt op adres \$3FE/3FF.

Vierling-interrupts

Met SW7 (zie hoofdstuk 10) kunnen de klok, de VIA en de ACIA zo worden geschakeld dat ze al dan niet een IRQ en/of een NMI kunnen veroorzaken. Of ze werkelijk een interrupt veroorzaken, hangt af van commando's die ze hebben ontvangen.

Denk nu eens aan de situatie dat SW7 zo is geschakeld dat zowel de ACIA als de VIA een IRQ-interrupt kan veroorzaken, en er treedt een IRQ-interrupt op.

Vraag: Welke van de twee is het?

Antwoord: Dat zoeken we op. Elke functie die een interrupt kan veroorzaken, bezit een status-register waarin de interruptroutine kan zien of deze functie wil interrumpen. Hoe dat register er precies uitziet, staat in de beschrijving van die functie.

Waarschuwing

Er zijn programma's die niet onderbroken mogen worden omdat de goede werking afhangt van het binnen een bepaalde tijd uitvoeren van zekere instructies. Zo'n programma is bijv. de routine van DOS die diskettes leest en schrijft (RWTS voor intimi). Zulke programma's houden de IRQ-interrupt tegen maar de NMI-interrupt natuurlijk niet. Bij gebruik van die programma's moet het dus zeker zijn dat de NMI-interrupt nooit kan optreden.

Interrupts onder DOS 3.3

Het gebruik van interrupts onder DOS 3.3 moet met de nodige voorzichtigheid worden ondernomen. Weliswaar houdt DOS tijdens het schrijven en lezen van een sector van disk alle IRQ-interrupts tegen, NMI-interrupts blijven mogelijk.

Er is echter een ernstig probleem. Na het (op een onvoorspelbaar moment!) optreden van een IRQ-interrupt bergt de Monitor de Accumulator op in byte \$45. Helaas maakt DOS ook intensief gebruik van deze byte. Het gevolg: chaos!

Er is geen eenvoudige en afdoende oplossing voor dit probleem. Misschien kan ervoor gezorgd worden dat DOS nooit bezig is op het moment dat er een interrupt binnenkomt. Overigens bestaan er aangepaste (veelal versnelde) DOS-versies waarin dit probleem is opgelost. Een voorbeeld hiervan is Diversi-DOS.

Interrupts onder ProDOS

In tegenstelling tot DOS 3.3 ondersteunt ProDOS wel het gebruik van interrupts, zij het alleen de IRQ. Het afhandelen daarvan wordt zelfs nog vereenvoudigd doordat ProDOS de adressen van vier interrupt-routines kan bijhouden. Deze routines worden bij het optreden van een interrupt in een vaste volgorde aangeroepen, totdat één ervan terugmeldt dat de interrupt is afgehandeld. Voor het afhandelen van de interrupts door de klok, ACIA en VIA kunnen op deze manier aparte routines worden gebruikt, hetgeen ten goede komt aan de doorzichtigheid van die routines.

Het zou hier te ver voeren, precies aan te geven hoe dit alles in z'n werk gaat. Voor details wordt dus verwezen naar de "ProDOS Technical Reference Manual". Deze is so-wie-so noodzakelijk om gebruik te kunnen maken van de faciliteiten die ProDOS de assemblerprogrammeur te bieden heeft.

Klokhuis Vierlingkaart

Voorbeeld van een Interruptroutine

Als voorbeeld van een interruptroutine volgen nu aanwijzingen voor de bouw van een systeem dat elke seconde een klikje laat horen. Als we even vergeten dat een wekker van een tientje dit even goed kan als een computersysteem van meer dan drieduizend gulden, is het toch wel leuk.

Achtereenvolgens moeten we de klok met de IRQ-aansluiting verbinden, de klokinterrupt-frequentie instellen op eens per seconde, een interrupt-routine aanmelden bij de Monitor, de klok inschakelen, tegen de klok zeggen dat er interrupts gewenst zijn, en uiteindelijk de blokkering van de IRQ-interrupt opheffen.

In de interruptroutine moet dan de ingebouwde luidspreker zo worden bespeeld dat er een klik hoorbaar wordt, waarna het onderbroken programma wordt hervat.

Het voorbeeld is geschreven voor gebruik met BASIC, dus niet speciaal voor DOS of ProDOS. Verder wordt er gebruik gemaakt van de Klok. Wat de diverse handelingen precies inhouden, wordt uitgelegd in 9.3.3.

Opgelet! Tijdens het gebruik van dit programma is de ACIA niet bruikbaar.

Schakelaarinstelling

- SW1 Zet schakelaars 5, 7 en 8 op OFF, en schakelaar 6 op ON. Hierdoor interrumpeert de klok elke seconde.
- SW7 Zet schakelaars 1 en 7 op ON, de rest op OFF. Hierdoor worden alleen de klok-interrupts doorgegeven aan de IRQ-pen van de 6502.

Klokhuis Vierlingkaart

Initialiseren

```
*
* De Vierlingkaart zit in slot 4.
* -----
*
SPEAKER EQU $C030
CREG     EQU $C4FF
KREG     EQU $CFFE
*
* Meld de IRQ-Interruptroutine aan
*
      LDA #INTROUT      ;Onderste 8 bits
      STA $3FE
      LDA #INTROUT/256;Bovenste 8 bits
      STA $3FF
*
* Schakel de klok in m.b.v. het C-register.
* Dit moet zoveel mogelijk onveranderd blijven.
*
      LDA CREG
      AND #$F0          ;Interessante gegevens
                        ;bewaren.
      ORA #$8E          ;Kaart en klok aan
      STA CREG
*
* Maak klokinterrupts mogelijk
*
      LDA #$CF
      STA KREG
*
* Sta interrupts toe, d.w.z. activeer de IRQ-
* pen van de 6502.
*
      CLI
*
* Stond het bovenstaande in een subroutine,
* dan volgt nu:
*
      RTS
*
* Hiermee wordt het hoofdprogramma hervat.
* Dit kan het BASIC-systeem zijn; in dat geval
* is het mogelijk, een BASIC-programma uit te
* voeren, mits dit geen gebruik maakt van de
* Vierlingkaart.
```

Klokhuis Vierlingkaart

Interruptroutine

```
*
* Deze routine wordt aangeroepen zodra de klok
* een interrupt geeft, dus elke seconde.
*
INTROUT EQU *
        SBC #1
        BNE WACHT
*
* ... en tik nog eens. Dat is een klik.
*
        BIT SPEAKER
*
        LDA #$CF
        STA KREG
*
* Herstel de accumulator.
* Vanuit de Monitor is deze gered
* op adres $45.
*
        LDA $45
*
* Keer terug naar het onderbroken programma
*
        RTI
```

Klokhuis Vierlingkaart

9.3 De functies

De diverse Vierling-functies zijn verwezenlijkt met de 'functies' VIA, ACIA, Klok, EPROM en RAM. De eigenschappen daarvan worden hieronder schematisch aangegeven. De beschrijvingen zijn gebaseerd op de specificaties die door de fabrikanten zijn gepubliceerd.

9.3.1 De VIA

De VIA (Versatile Interface Adapter) is van het type 6522.

De VIA kan zonder speciale voorzieningen vanuit een EPROM worden gebruikt, en onderscheidt zich hiermee van de ACIA en de Klok.

Interne registers

Alle gegevens die m.b.v. de VIA de computer in en uit gaan, doen dat via een aantal interne registers. Ook allerlei instellingen, bijv. welke aansluitingen als invoer fungeren en welke als uitvoer, staan in deze registers.

Klokhuis Vierlingkaart

De interne registers zijn:

| | | |
|------|-----------------------------|-----------|
| PCR | Peripheral Control Register | |
| ACR | Auxiliary Control Register | |
| IER | Interrupt Enable Register | |
| IFR | Interrupt Flag Register | |
| DDRA | Data Direction Register A | } Poort A |
| IRA | Input Register A | |
| ORA | Output Register A | |
| DDRB | Data Direction Register B | } Poort B |
| IRB | Input Register B | |
| ORB | Output Register B | |
| SR | Shift Register | |
| T1L | Timer 1 Latch (L en H) | |
| T1C | Timer 1 Counter (L en H) | |
| T2L | Timer 2 Latch (alleen L) | |
| T2C | Timer 2 Counter (L en H) | |

Poort B van de VIA wordt op de Vierlingkaart gebruikt door de parallele printerinterface (BSB-1).

Toegang tot de interne registers

De interne registers zijn bereikbaar op adres $\$C0xa$, waarin 'x' het slotnummer van de Vierlingkaart is, verhoogd met 8. De 'a' is af te lezen uit onderstaande tabel. Zit de kaart bijv. in slot 4, dan is het SR-register bereikbaar op adres $\$C0CA$, want $4+8=\$C$.

Klokhuis Vierlingkaart

| Adr C08a | Schrijven | Lezen | |
|-------------|---------------|-----------|--------|
| C080 | ORB | IRB | #3, #4 |
| C081 | ORA | IRA | #0, #1 |
| C082 | DDRB | DDRB | |
| C083 | DDRA | DDRA | |
| C084 | T1-L | T1C-L; #6 | |
| C085 | T1L-H; #6 [1] | T1C-H | |
| C086 | T1L-L | T1L-L | |
| C087 | T1L-H; #6 | T1L-L | |
| C088 | T2L-H | T2C-L; #5 | |
| C089 | T2C-H [2] | T2C-H | |
| C08A | SR | SR | #2 |
| C08B | ACR | ACR | |
| C08C | PCR | PCR | |
| C08D | IFR | IFR | |
| C08E | IER | IER | |
| C08F | ORA | IRA | #0, #1 |

Opmerkingen.

[1] T1L-H wordt in T1C-H gezet, en T1L-L in T1C-L; teller T1 wordt gestart.

[2] T2L-L wordt in T2C-L gezet; teller T2 wordt gestart.

De constructie '#n' geeft aan dat bit 'n' van het IFR (Interrupt Flag Register) bij het aanspreken van het register op 0 wordt gezet.

Een 1-bit in een DDR geeft aan dat het overeenkomstige bit van het Outputregister een uitvoerbit is. Een 0-bit geeft aan dat het een invoerbit is.

Een 1-bit in een IFR geeft een interrupt aan. Met het IER wordt ingesteld, onder welke omstandigheden er een interrupt kan optreden.

Klokhuis Vierlingkaart

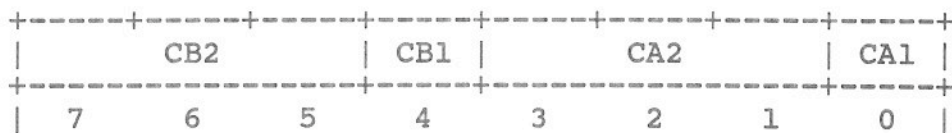
PCR - Peripheral Control Register

===

In het PCR staat hoe de 'hand shaking' is geregeld. Dit heeft betrekking op de uitwisseling van signalen tussen een parallelle zender en ontvanger. Met behulp van deze signalen kan een zender aangeven dat er gegevens beschikbaar zijn, en een ontvanger of deze bereid is, de gegevens over te nemen. Is de ontvanger bijv. een printer, dan kan deze tijdens het teruglopen van de printkop aangeven dat het aannemen van tekst niet mogelijk is.

De signalen worden uitgewisseld via een aantal pennen aan het VIA-IC. Deze heten CA1, CA2, CB1 en CB2.

De indeling van het PCR is:



76543210

- CB2 Bestuurt het CB2-sigitaal.
100..... Handshake: CB2 wordt 0 bij schrijven naar ORB; het CB2-sigitaal wordt 1 bij omslaan van het CB1-sigitaal.
101..... Pulse output: het CB2-sigitaal wordt 0 bij aanspreken van ORB.
110..... Het CB2-sigitaal wordt 0.
111..... Het CB2-sigitaal wordt 1.
- CB1 Bestuurt de CB1-interrupt (IFR4):
...0..... Interrupt bij 1/0-overgang op CB1.
...1..... Interrupt bij 0/1-overgang op CB1.
- CA2 xxx. Als CB2C, maar op ORA, CA1 en CA2.
- CA1 x Als CB1C, maar op IFR0 en CA1.

Klokhuis Vierlingkaart

ACR - Auxiliary Control Register

===

Het ACR bestuurt de werking van de twee timers en het schuifregister. Tevens fungeert het als een uitbreiding van het PCR.

| | T1C | T2C | SRC | PBLE | PALE | | | |
|-----------|--|-----|-----|-----------|------|---------|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | 76543210 | | | | | | | |
| T1C | mode | | | interrupt | | PB7-out | | |
| 00..... | one-shot | | | enkel | | geen | | |
| 01..... | free-running | | | continu | | geen | | |
| 10..... | one-shot | | | enkel | | load T1 | | |
| 11..... | free-running | | | continu | | toggle | | |
| T2C | ..0..... T2 in one-shot mode. | | | | | | | |
| | ..1..... T2 telt voorafbepaald aantal pulsen op PB6. | | | | | | | |
| SRC | Bestuurt het Shift Register. | | | | | | | |
| ...000.. | SR uitgeschakeld. | | | | | | | |
| ...001.. | Inschuiven m.b.v. Timer 2. | | | | | | | |
| ...010.. | Inschuiven m.b.v. phi-2. | | | | | | | |
| ...011.. | Inschuiven m.b.v. externe klok. | | | | | | | |
| ...100.. | Vrijlopende uitvoer m.b.v. Timer 2. | | | | | | | |
| ...101.. | Uitschuiven m.b.v. Timer 2. | | | | | | | |
| ...110.. | Uitschuiven m.b.v. phi-2. | | | | | | | |
| ...111.. | Uitschuiven m.b.v. externe klok. | | | | | | | |
| PBLE..... | 1. De Port-B Latch neemt gegevens over bij een CB1-interrupt (IFR4). | | | | | | | |
| PALE..... | 1 De Port-A Latch neemt gegevens over bij een CA1-interrupt (IFR1). | | | | | | | |

Klokhuis Vierlingkaart

IFR - Interrupt Flag Register

===

Het IFR geeft aan of er een interrupt is opgetreden, en zo ja, welke dat is geweest.

| IRQ | T1 | T2 | CB1 | CB2 | SR | CA1 | CA2 |
|-----|----|----|-----|-----|----|-----|-----|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

IRQ 1: Minstens een van de andere bits is 1.
T1 Timer 1 afgelopen.
T2 " 2 " .
CB1 Niveauwisseling op CB1.
CB2 " " CB2.
SR Schuiving voltooid.
CA1 Niveauwisseling op CA1.
CA2 " " CA2.

IER - Interrupt Enable Register

===

Het IER geeft aan, onder welke omstandigheden er een interrupt wordt doorgegeven.

| IER | T1 | T2 | CB1 | CB2 | SR | CA1 | CA2 |
|-----|----|----|-----|-----|----|-----|-----|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

IER Dit geeft aan of er interrupts al dan niet mogelijk moeten zijn.
0 Houdt interrupts tegen voor de omstandigheden waarvoor het bit op 1 staat.
1 Maak interrupts mogelijk voor die omstandigheden.

Voorbeeld: Een waarde van 10000010 maakt interrupts mogelijk door CA1.

Klokhuis Vierlingkaart

9.3.2 De ACIA

De ACIA (Asynchronous Communications Interface Adapter) is van het type 6551.

De ACIA is alleen bereikbaar als het C-register een inhoud heeft van de vorm:

```
76543210  
xxxx1111
```

De x-en geven bits aan die er hier niet toe doen. De ACIA wordt gebruikt door de seriële printer-interface (BSB-2) en de terminalprogramma (BSB-3).

De ACIA deelt het adresgebied \$C8xx met de EPROMS; als gevolg hiervan is het niet mogelijk, in een programma in EPROM de ACIA direct te gebruiken.

De transportsnelheid van de ACIA wordt bepaald door de software. De printerinterface maakt daarbij gebruik van DIP-schakelaar SW4, maar het terminalprogramma vraagt er bij het aanroepen om.

Seriële communicatie

Over een parallele verbinding worden alle bits van een byte gelijktijdig verzonden. Het kanaal is dus even breed als een byte. De transportsnelheid hangt af van de snelheid waarmee de zender bytes kan produceren, en de ontvanger deze kan consumeren.

Over een seriële verbinding daarentegen worden de bits van elke byte na elkaar verzonden. Het kanaal is dus 1 bit breed.

Een bit kan (het hoeft nauwelijks gezegd te worden) twee waarden hebben. Deze worden op een seriële lijn voorgesteld met twee verschillende spanningen, nl.:

Mark Dit is een hoog niveau.

Space Dit is een laag niveau.

In rusttoestand staat op een seriële lijn een hoog niveau, dus Mark.

Klokhuis Vierlingkaart

Elke verzonden of ontvangen byte ziet er zo uit:

```
+-----+-----+-----+-----+-----+-----+-----+-----+
|start|bit0|bit1|...|bitn|pariteit|stop|stop|
+-----+-----+-----+-----+-----+-----+-----+-----+
```

De betekenis van de bits is:

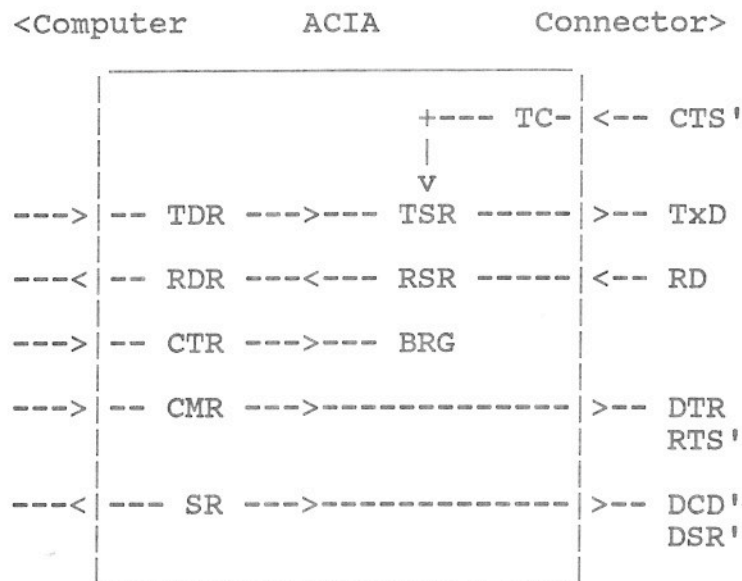
| | |
|----------|---|
| start | Startbit. Dit geeft het begin van een byte aan. Omdat dit moet afwijken van de rusttoestand, heeft een startbit het Space-niveau. |
| bit0 | Eerste verzonden databit. |
| bitn | Laatste verzonden databit. |
| pariteit | Controlebit. Dit heeft een vaste waarde (Mark of Space), of een waarde die de pariteit (het aantal 1-bits, inclusief het pariteitsbit) Even of Oneven maakt. Het hoeft echter niet aanwezig te zijn. |
| stop | Stopbit. Dit geeft het einde van een byte aan. Het aantal stopbits varieert. Het kan 1 of 2 zijn, maar ook 1 1/2. Direct na het laatste stopbit kan het startbit van de volgende byte volgen. Een stopbit heeft het Space-niveau. |

De snelheid waarmee gegevens over een seriële lijn worden verzonden, wordt uitgedrukt in bits per seconde. Om het aantal bytes te bekomen dat per seconde kan worden verzonden, moet dit getal worden gedeeld door het aantal bits in een byte, verhoogd met het aantal extra bits, d.w.z. startbit, stopbit(s) en evt. pariteitsbit. Over een 300 bps-lijn gaan 7-bits-karakters (meestal ASCII), met pariteit verzonden met 1 stopbit, met een snelheid van $300/(7+1+2) = 30$ bytes per seconde.

Klokhuis Vierlingkaart

ACIA-registers

De globale opbouw van de ACIA is aangegeven in onderstaande figuur. Er staan alleen de onderdelen in die op de Vierlingkaart bereikbaar zijn.



De ACIA bevat een aantal registers waarvan er vijf vanuit de computer bereikbaar zijn. Het zijn achtereenvolgens:

- TDR Transmit Data Register. Dit wordt vanuit het programma gevuld met de te verzenden byte, door te schrijven naar een data-register.
- RDR Receive Data Register. Het programma krijgt de inhoud hiervan door te lezen van een dataregister. Het pariteitsbit komt hier niet in terecht.
- CTR Control Register. Dit bestuurt de werking van de ACIA, met name de transportsnelheid, en wordt vanuit het programma gevuld.
- CMR Command Register. Dit bestuurt de werking van de ACIA (zoals interrupts) en wordt vanuit het programma gevuld.

Klokhuis Vierlingkaart

SR Status Register. Dit geeft aan hoe het gegevenstransport is verlopen.

Het RDR en het TDR kunnen gelijktijdig actief zijn. Dit heet 'full duplex'-bedrijf. Het programma moet daar wel op gebouwd zijn.

Overige interne ACIA-registers

BRG Baud Rate Generator. Deze bepaalt de snelheid waarmee bytes worden verzonden en ontvangen.

TSR Transmit Shift Register. Hiermee wordt de te verzenden byte bit voor bit in de juiste snelheid via TxD uitgestuurd.

RSR Receive Shift Register. Hierin wordt een te ontvangen byte opgebouwd uit bits die van RxD worden gelezen.

Signalen

TxD Transmit Data. Aansluiting voor seriële uitvoer. Van elke byte wordt het minst-significante ('onderste') bit het eerst verzonden. De snelheid wordt bepaald door de BRG, die op zijn beurt wordt ingesteld m.b.v. het CTR.

RxD Receive Data. Aansluiting voor seriële invoer. Zie verder TxD.

RTS' Request To Send. Dit dient om een Modem te vertellen dat er een byte mag komen..

CTS' Clear To Send. Hiermee geeft een Modem te kennen dat er een byte gezonden kan worden.

DTR' Data Terminal Ready. Dit meldt aan een Modem dat de ACIA (dus de Apple) is ingeschakeld.

DSR' Data Set Ready. Hiermee geeft een Modem aan dat er gegevens getransporteerd kunnen worden.

DCD' Data Carrier Detect. Hiermee geeft een Modem aan of er een actieve verbinding met een andere Modem is.

N.B. Een ' achter een signaalnaam geeft aan dat dit signaal bij 0 actief is.

Klokhuis Vierlingkaart

Registeradressen

De adressen van de registers zijn:

| adres | Lezen | Schrijven |
|--------|-------|-----------|
| \$C800 | RDR | TDR |
| \$C801 | SR | <Reset> |
| \$C802 | | CMR |
| \$C803 | | CTR |

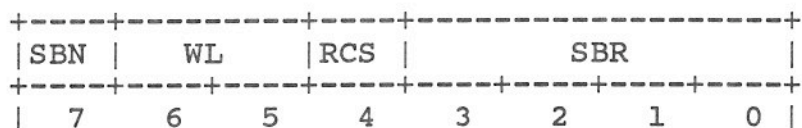
<Reset> is geen register doch een opdracht aan de ACIA om van het CMR bits 0 t/m 4 en van het SR bit 2 op 0 te zetten. Zie de beschrijving van deze registers voor een verklaring van het gevolg.

Verder wordt er nog het volgende gedaan:

- DTR' wordt hoog.
- RDR- en TDR-interrupts worden onmogelijk. Een eventueel al aangemelde IRQ blijft uitstaan.
- DCD'- en DSR'-interrupts worden onmogelijk. Een IRQ die was aangemeld t.g.v. DCD' of DSR' wordt afgemeld.

Klokhuis Vierlingkaart

CTR - Control Register (\$C803)



| | |
|----------|---|
| 76543210 | |
| SBN | Aantal stopbits |
| 0..... | 1 stopbit |
| 1..... | 2 stopbits |
| | 1 1/2 stopbit (WL=5, geen pariteit) |
| | 1 stopbit (WL=8 met pariteit) |
| WL | Woordlengte: aantal bits per byte. |
| .00..... | 8 bits |
| .01..... | 7 bits |
| .10..... | 6 bits |
| .11..... | 5 bits |
| RCS | Receiver Clock Source |
| ...0.... | Externe klok. |
| ...1.... | Baud Rate. Dit is de enig mogelijke waarde voor de Vierlingkaart. |
| SBR | Selected Baud Rate |
|0000 | 16 x Externe klok (niet op Vierling) |
|0001 | 50 |
|0010 | 75 |
|0011 | 109.92 |
|0100 | 134.58 |
|0101 | 150 + |
|0110 | 300 + |
|0111 | 600 + |
|1000 | 1200 + |
|1001 | 1800 |
|1010 | 2400 + |
|1011 | 3600 |
|1100 | 4800 + |
|1101 | 7200 |
|1110 | 9600 + |
|1111 | 19200 + |

De Besturingssoftware kent alleen de met + gemerkte snelheden.

Klokhuis Vierlingkaart

CMR - Command Register (\$C802)

Het Command Register bestuurt specifieke instellingen en functies.

| | | | | | | | | | | | | | | | | |
|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---|--|---|--|
| +-----+ | +-----+ | +-----+ | +-----+ | +-----+ | +-----+ | +-----+ | +-----+ | +-----+ | | | | | | | | |
| | PMC | | PME | | REM | | TIC | | IRD | | DTR | | | | | |
| +-----+ | +-----+ | +-----+ | +-----+ | +-----+ | +-----+ | +-----+ | +-----+ | +-----+ | +-----+ | +-----+ | +-----+ | +-----+ | | | | |
| | 7 | | 6 | | 5 | | 4 | | 3 | | 2 | | 1 | | 0 | |

76543210

| | |
|----------|--|
| PMC | Parity Mode Control |
| 00..... | Oneven pariteit |
| 01..... | Even pariteit |
| 10..... | Mark-pariteit; geen controle |
| 11..... | Space-pariteit; geen controle |
| PME | Parity Mode Enabled |
| ..0..... | Er wordt geen pariteitsbit gegenereerd of gecontroleerd. |
| REM | Receiver Echo Mode |
| ...0.... | Receiver Normal Mode |
| ...1.... | Receiver Echo Mode (TIC = 0) Elke ontvangen byte wordt automatisch teruggezonden, met een vertraging van een halve bittijd. |
| TIC | Transmitter Interrupt Control |
|00.. | RTS' = 1; geen interrupts. |
|01.. | RTS' = 0; interrupts toegestaan. |
|10.. | RTS' = 0; geen interrupts. |
|11.. | RTS' = 0; geen interrupts; zend Break-sigitaal. |
| IRD | Interrupt Request Disabled |
|0. | IRQ door RDR toegestaan. |
|1. | IRQ geblokkeerd. |
| DTR | Data Terminal Ready |
|0 | DTR'-signaal = 1; geen interrupts. |
|1 | DTR'-signaal = 0; interrupts toegestaan. |

Klokhuis Vierlingkaart

SR - Status Register (\$C801)

```

+-----+-----+-----+-----+-----+-----+-----+-----+
|IRQ  |DSR'|DCD'|TDRE|RDRF|OVRN| FE  | PE  |
+-----+-----+-----+-----+-----+-----+-----+-----+
|  7   |  6   |  5   |  4   |  3   |  2   |  1   |  0   |

```

| Afk. | H | P | Naam |
|------|------|-----|------------------------------|
| 7 | IRQ | 0 - | Interrupt |
| 6 | DSR' | - - | Data Set Ready |
| 5 | DCD' | - - | Data Carrier Detect |
| 4 | TDRE | 1 - | Transmit Data Register Empty |
| 3 | RDRF | 0 - | Receive Data Register Full |
| 2 | OVRN | 0 0 | Receiver Overrun |
| 1 | FE | 0 - | Framing Error |
| 0 | PE | 0 - | Parity Error |

H: Harde reset. P: Geprogrammeerde reset.

IRQ wordt 0 als het Status Register wordt gelezen.
 IRQ wordt 1 als RDRF, TDRE, DCD', of DSR' van waarde veranderen.

RDRF wordt 0 als het RDR wordt gelezen.

TDRE wordt 0 als het TDR wordt gevuld.

Afhandeling van het Status-register vereist de volgende stappen.

- Lees het Status-register. Hierdoor wordt het IRQ-bit afgezet.
- Controleer aan de hand van het IRQ-bit of er een interrupt was geweest.
- Controleer DCD' en DSR'. Deze moeten worden vergeleken met de vorige waarde. Zijn ze allebei 0, en waren ze dat al, dan moeten de overige bits worden gecontroleerd.
- Controleer of RDRF 1 is (RDR vol), en zo ja, Parity, Overrun, en Framing Error.
- Controleer of TDRE 1 is.
- Is geen van het bovenstaande van toepassing, dan is het CTS'-signaal afgevallen (hoog).

Klokhuis Vierlingkaart

9.3.3 De Klok

De klok is van het type MSM 5832. De communicatie met de klok verloopt via twee adressen. In het K-register (\$CFFE) worden opdrachten aan de klok geplaatst, of het nummer van één van de 13 interne klokregisters geplaatst. Via het D-register (\$CFFD) kunnen daarna de bijbehorende gegevens worden gelezen of geschreven. De klok wordt bereikbaar als het C-register een inhoud heeft van de vorm:

```
76543210
xxxx1110
```

De x-bits zijn hier niet van belang. Vermits de klok wordt ingeschakeld middels het C-register, is gebruik ervan vanuit een EPROM niet direct mogelijk.

De klok wordt gebruikt door de klokleesroutine (BSB-4) en het gelijkzetsprogramma (BSB-5).

De MSM 5832 bewaart de gegevens in een dertiental interne registers. Elk van deze registers is bereikbaar door het nummer ervan in het K-register te plaatsen. De registers zijn:

| REG | Naam | Max | |
|-----|--------|-----|--|
| 0 | Sec 1 | 9 | Op nul gezet als er naar de klok wordt geschreven. |
| 1 | Sec 10 | 5 | |
| 2 | Min 1 | 9 | |
| 3 | Min 10 | 5 | |
| 4 | Uur 1 | 9 | |
| 5 | Uur 10 | 2 | [1] |
| 6 | Week | 6 | |
| 7 | Dag 1 | 9 | |
| 8 | Dag 10 | 3 | [2] |
| 9 | Mnd 1 | 9 | |
| 10 | Mnd 10 | 1 | |
| 11 | Jaar 1 | 9 | |
| 12 | Jaar10 | 9 | |

Klokhuis Vierlingkaart

- [1] Verhogen met 8 voor 24-uursklok; anders met 4 als het na de middag is.
- [2] Verhogen met 4 in een schrikkeljaar. Wordt automatisch teruggezet na 29 februari.

K-register (\$CFFE)

Met het K-register kan een intern klokregister worden geselecteerd voor lezen of schrijven, de klok worden stilgezet of gelijkgezet op de dichtstbijzijnde hele minuut, en interrupts door de klok mogelijk gemaakt.

In het K-register kan alleen worden geschreven. Via het D-register (zie beneden) kan evenwel een deel van K-register worden teruggelezen.

| INT | READ | ADJ | HOLD | REG | | | |
|-----|------|-----|------|-----|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

- 7 INT 1: Maak klok-interrupts mogelijk. Welke interrupt dat is (IRQ of NMI), wordt ingesteld met SW7. Hoe vaak de klokinterrupt optreedt, wordt ingesteld met SW1.
0: Zet na een interrupt de aanvraag af.
- 6 READ Lees de klok uit.
- 5 ADJ Stel de klok in op de dichtstbijzijnde gehele minuut.
- 4 HOLD Zet de klok stil. Het duurt 150 microseconden vooraleer de gegevens van de klok veilig kunnen gelezen of geschreven worden. Dit signaal moet niet langer dan 1 seconde op staan, anders gaat de klok achter lopen.
- 3-0 REG Nummer van een intern klokregister.

Klokhuis Vierlingkaart

D-register (\$CFFD) (lezen)

Het D-register wordt gebruikt om gegevens te lezen uit één van de interne klokregisters, en voor het teruglezen van een deel van het K-register.

| | | | | | | | | |
|--|-----|--|------|--|-----|--|------|--|
| | | | | | | | | |
| | INT | | READ | | ADJ | | HOLD | |
| | | | | | | | DATA | |
| | 7 | | 6 | | 5 | | 4 | |
| | | | | | | | 3 | |
| | | | | | | | 2 | |
| | | | | | | | 1 | |
| | | | | | | | 0 | |

Bit

- 7 INT De klok heeft een interrupt-aanvraag gedaan.
- 6 READ De klok wordt uitgelezen.
- 5 ADJ De klok wordt ingesteld op de dichtstbijzijnde hele minuut.
- 4 HOLD De klok staat stil.
- 3-0 DATA Gegevens uit een intern klokregister.

D-register (\$CFFD) (schrijven)

Het D-register wordt gebruikt voor het schrijven van gegevens naar een intern klokregister. Dit register moet eerst worden geselecteerd met het K-register.

| | | | | | | | | |
|--|---|--|---|--|---|--|------|--|
| | | | | | | | | |
| | | | - | | | | DATA | |
| | 7 | | 6 | | 5 | | 4 | |
| | | | | | | | 3 | |
| | | | | | | | 2 | |
| | | | | | | | 1 | |
| | | | | | | | 0 | |

Bit

- 7-4 ---- Niet te wijzigen met het D-register.
- 3-0 DATA Gegevens voor een intern klokregister.

Klokhuis Vierlingkaart

9.3.4 EPROM

De EPROM op de Vierlingkaart kunnen zijn van het type 2716 (2K bytes), 2732 (4K bytes) of 2764 (8K bytes).

De EPROMs met gebruikersprogramma's moeten zo zijn geprogrammeerd dat ze in Extension ROM kunnen draaien. Dit zijn de 2K bytes van \$C800 tot \$CFFF die ter beschikking staan van alle uitbreidingskaarten (maar niet meer dan 1 tegelijkertijd). Er zijn echter enkele restricties.

- 1) Elke EPROM moet als vijfde byte het aantal entrypoints bevatten. Dit wordt door de Besturingssoftware gebruikt om te controleren of de gebruiker geen fout heeft gemaakt. Het volgnummer van het entrypoint wordt gebruikt als index in een entrypointtabel die op byte 5 begint. In de entrypointtabel staan de adressen van de routines die van buitenaf bereikbaar zijn. Het voordeel van deze manier van werken is dat de programmeur van de EPROM alle vrijheid heeft om de instructies te wijzigen, als de entrypointtabel dan ook maar wordt aangepast.

De indeling is dus schematisch als volgt.

| | | | |
|----|--|--------------------|--|
| 00 | | | |
| 02 | | | |
| 04 | | aantal entrypoints | |
| 05 | | adres van entry 1 | |
| 07 | | adres van entry 2 | |
| 09 | | | |
| 21 | | adres van entry 9 | |

Klokhuis Vierlingkaart

Elke adres omvat 2 bytes. Het is een 6502-adres, dus met het minst-significante deel eerst.

- 2) De EPROMs bezetten het adresgebied \$C800-\$CFFE. Daar bevinden zich ook de registers voor de ACIA en de klok. Derhalve zijn deze vanuit een programma in EPROM niet direct bereikbaar.
- 3) Een programma in de bovenste 256 bytes van een EPROM van 8K bytes kan niet gebruik maken van de Vierling-RAM.

Opmerking.

Er bestaan RAM-geheugens met dezelfde aansluitingen als de EPROMs uit de 2700-serie. Deze kunnen zonder meer in een EPROM-voetje worden gestoken. Ze zijn dan vergelijkbaar met 2K-EPROMs van het type 2716.

Klokhuis Vierlingkaart

10 De Hardware

=====

In dit hoofdstuk worden de mechanische aspecten van de Vierlingkaart besproken, t.w. DIP-schakelaars (waarmee de werking van de kaart is te beïnvloeden), connectoren (waarmee de kaart met de buitenwereld communiceert), en semipermanente doorverbindingen.

10.1 De DIP-schakelaars

Aan de bovenrand van de Vierlingkaart zitten drie blokken met elk acht DIP-schakelaars. (DIP staat voor Dual Inline Package, de benaming voor een IC-behuizing; de schakelaar heeft precies die afmetingen.) Van voor naar achter zijn het:

- SW1 Instellen van het nummer van de bank van de Besturingssoftware die bij het inschakelen actief is, en de interruptfrequentie van de klok.
- SW4 Aangeven van het gebruikte EPROM-type en de transportsnelheid van de ACIA.
- SW7 Doorgeven van interrupts van de VIA, de ACIA en de klok.

Deze schakelaars worden nu achtereenvolgens besproken.

10.1.1 SW1: BSB-instelling en Klokinterrupts

| BSB | | | | Klok-interrupt | | | | |
|-----|-----|----|----|----------------|----|----|----|------------|
| A11 | A10 | A9 | A8 | IMS | IS | IM | IH | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | Schakelaar |
| h | g | f | e | d | c | b | a | Schema |

Klokhuis Vierlingkaart

Het BSB-deel van SW1 (1-4)

De Besturingssoftware op de Vierlingkaart beslaat 8 K bytes, en is opgedeeld in 32 banken van 256 bytes. Van deze banken, de BSB's, kan er op elk moment slechts 1 actief zijn. Het wisselen van de BSB's doet de Besturingssoftware geheel ongemerkt voor de gebruiker. Met SW1(1-4) wordt ingesteld, welke BSB er actief is bij het aanzetten van de Apple, of na het indrukken van de (CONTROL-) RESET-knop. Met vier schakelaars kunnen de eerste 16 BSB's worden aangegeven.

| BSB | 1 | 2 | 3 | 4 |
|-----|-----|-----|-----|-----|
| 0 | On | On | On | On |
| 1 | On | On | On | Off |
| 2 | On | On | Off | On |
| 3 | On | On | Off | Off |
| 4 | On | Off | On | On |
| 5 | On | Off | On | Off |
| 6 | On | Off | Off | On |
| 7 | On | Off | Off | Off |
| 8 | Off | On | On | On |
| 9 | Off | On | On | Off |
| A | Off | On | Off | On |
| B | Off | On | Off | Off |
| C | Off | Off | On | On |
| D | Off | Off | On | Off |
| E | Off | Off | Off | On |
| F | Off | Off | Off | Off |

Het Interrupt-deel van SW1 (5-8)

Zoals bij de beschrijving van SW7 wordt vermeld, kan de Klok een interrupt veroorzaken. Met SW1(5-8) wordt ingesteld, om de hoeveel tijd dit gebeurt, en wel als volgt:

Klokhuis Vierlingkaart

| Sch | Naam | Omschrijving |
|-----|------|----------------------|
| 8 | IH | Elk uur |
| 7 | IM | Elke minuut |
| 6 | IS | Elke seconde |
| 5 | IMS | Elke 'milli'-seconde |

Wordt een combinatie van deze schakelaars op ON gezet, dan kan de klok worden beschadigd.

N.B. De klok kent geen echte milli-seconden: de interrupts treden niet 1000 maal maar 1024 maal per seconde op.

10.1.2 SW4: EPROM-type en S-register

```

+---+---+---+---+---+---+---+---+
|E1|E2|E3|E4|E5|  S-reg |
+---+---+---+---+---+---+---+---+
  8 7  6  5  4 | 3  2  1   Schakelaar
  a b  c  d  e | f  g  h   Schema
    
```

Het EPROM-deel van SW4 (4-8)

In de EPROM-voetjes kunnen EPROMs worden geplaatst van verschillende grootte. In verband met de verschillen tussen de aansluitingen moet er ingesteld worden of het om een kleine (2K, type 2716) of grote (4K, type 2732 of 8K, type 2764) EPROM gaat.

Een schakelaar in de ON-stand geeft een 2K EPROM aan.

Onafhankelijk hiervan moet de software ook op de hoogte zijn van de grootte van de gemonteerde EPROMs, omdat er tussen de delen van de EPROM moet worden geschakeld.

Klokhuis Vierlingkaart

Het S-register-deel van SW4 (1-3)

De resterende drie schakelaars van SW4 kunnen door een programma worden uitgelezen. Als adres hiervan fungeren \$C804 t/m \$C807, maar normaliter wordt alleen \$C804 gebruikt.

Staat een schakelaar in de ON-stand, dan leest het programma daar een 0.

Het S-register wordt gebruikt om aan de seriële interface (BSB-2) door te geven, welke transportsnelheid deze moet toepassen, uitgedrukt in bits per seconde.

| 3 | 2 | 1 | Snelheid |
|-----|-----|-----|----------|
| On | On | On | 150 |
| On | On | Off | 300 |
| On | Off | On | 600 |
| On | Off | Off | 1200 |
| Off | On | On | 2400 |
| Off | On | Off | 4800 |
| Off | Off | On | 9600 |
| Off | Off | Off | 19200 |

De schakelaars nemen drie bits van het S-register in beslag. De overige vijf bits hebben een waarde die ook is in te stellen, zij het minder snel dan met een schakelaar. Dit wordt besproken in 10.3.

10.1.3 SW7: Interrupts

| | | | | | | | | |
|---------------------------|---|------|---|-----|---|------|---|------------|
| +-----+-----+-----+-----+ | | | | | | | | |
| INT | | ACIA | | VIA | | Klok | | |
| +-----+-----+-----+-----+ | | | | | | | | |
| IRQ | | CHN | | IRQ | | NMI | | |
| +-----+-----+-----+-----+ | | | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | Schakelaar |
| h | g | f | e | d | c | b | a | Schema |

Klokhuis Vierlingkaart

Dit schakelaarblok regelt de doorgave van interrupts van klok, VIA en ACIA naar de 6502-microprocessor in de Apple. Zie hoofdstuk 10 voor een beschrijving van de mogelijke interrupts. Met SW7 kunnen de klok, de VIA en de ACIA zo worden geschakeld dat ze al dan niet een IRQ en/of een NMI kunnen veroorzaken. Dat ze alle drie dezelfde interrupt kunnen veroorzaken, lijkt een probleem te kunnen geven. In het interrupt-programma kan echter worden nagegaan, waar de interrupt vandaan komt. Zowel de klok als de VIA als de ACIA bezitten een status-register. Hierin staat een bit dat aangeeft of er een interrupt-aanvraag is gepleegd.

De schakelaars hebben de volgende betekenis.

- De NMI-schakelaars onder het hoofd ACIA, VIA en Klok verbinden de interrupt-aansluiting daarvan direct met de NMI-aansluiting van de 6502.
- De IRQ-schakelaars onder het hoofd ACIA, VIA en Klok verbinden de interrupt-aansluiting daarvan uiteindelijk met de IRQ-aansluiting van de 6502. Er zijn daarbij twee varianten:
 - De interrupt gaat direct naar de IRQ-aansluiting van de 6502 als de IRQ-schakelaar onder het hoofd INT op ON staat.
 - De interrupt wordt pas doorgegeven aan de IRQ-aansluiting van de 6502 als er geen andere kaart is die een interrupt heeft veroorzaakt, en deze kaart in een slot zit met een hoger nummer dan die van de Vierlingkaart.

Om IRQ-interrupts van de Vierlingkaart mogelijk te maken, moet in elk geval schakelaar 1 of 2 aan staan, en verder een combinatie van de schakelaars 3, 5 en/of 7.

NMI-interrupts worden mogelijk door het aanzetten van schakelaar 4, 6 en/of 8.

Zie voor de klokinterrupts ook SW1.

Klokhuis Vierlingkaart

10.2 Connectoren

Langs de bovenrand van de Vierlingkaart bevinden zich enkele connectoren.

Dit zijn van voor naar achter: de aansluiting voor de batterij, de verbinding met de VIA, en de verbinding met de ACIA.

De nummering van de pennen op de VIA- en ACIA-connectoren is:

| | | | | | |
|---|---|---|---|----|-----|
| 1 | 3 | 5 | 7 | 9 | ... |
| 2 | 4 | 6 | 8 | 10 | ... |

10.2.1 De batterijconnector

De batterijconnector heeft slechts 3 pennen.

| pen | functie |
|-----|---------|
|-----|---------|

| | |
|---|---|
| 1 | + |
| 2 | - |
| 3 | + |

Door deze opzet kan de stekker niet fout op de connector worden gestoken als elk pennetje in een gaatje zit.

De batterijspanning is 3,6 volt.

Klokhuis Vierlingkaart

10.2.1 De VIA-connector

| Connector | | | Kabel (par. printer) | |
|-----------|--------|--------|----------------------|-------------------|
| pen | in/uit | afk. | pen | functie |
| 1 | | !CB2 | 1 | Data Strobe (neg) |
| 2 | | !CB1 | 10 | Acknowledge (neg) |
| 3 | in/uit | PB7 . | 9 | D7 |
| 4 | in/uit | PB6 . | 8 | D6 |
| 5 | in/uit | PB5 . | 7 | D5 |
| 6 | in/uit | PB4 . | 6 | D4 |
| 7 | in/uit | PB3 . | 5 | D3 |
| 8 | in/uit | PB2 . | 4 | D2 |
| 9 | in/uit | PB1 . | 3 | D1 |
| 10 | in/uit | PB0 . | 2 | D0 |
| 11 | | CA1 | | |
| 12 | | CA2 | | |
| 13 | in/uit | PA7 . | | |
| 14 | in/uit | PA6 . | | |
| 15 | in/uit | PA5 . | | |
| 16 | in/uit | PA4 . | | |
| 17 | in/uit | PA3 . | | |
| 18 | in/uit | PA2 . | | |
| 19 | in/uit | PA1 . | | |
| 20 | in/uit | PA0 . | | |
| 21 | - | 0 . | 16 | GND |
| 22 | - | 0 . | | |
| 23 | in | +5V . | | |
| 24 | in | +12V . | | |
| 25 | in | -12V . | | |
| 26 | | | | |

Pennen 23, 24 en 25 zijn bij aflevering van de kaart niet direct met de bijbehorende voedingsspanningen verbonden doch via een weerstand. De bedoeling hiervan is, de stroom die maximaal geleverd kan worden, te beperken tot ongeveer 50 mA (kortsluitstroom).

In 10.3 staat hoe deze beperking kan worden opgegeven.

Klokhuis Vierlingkaart

10.2.3 De ACIA-connector

| pen | in/uit | afk | naam | kabelpen |
|-----|--------|------|---------------------|----------|
| --- | ----- | ---- | ---- | ----- |
| 1 | in | DCD' | Data Carrier Detect | 8 |
| 2 | uit | RTS' | Request To Send | 4 |
| 3 | in | CTS' | Clear To Send | 5 |
| 4 | uit | TXD | Transmit Data | 3 |
| 5 | uit | DTR' | Data Terminal Ready | 6 |
| 6 | in | DSR' | Data Set Ready | 20 |
| 7 | - | GND | Ground | 7 |
| 8 | in | RXD | Receive Data | 2 |
| 9 | - | - | Not connected | - |
| 10 | - | - | Not connected | - |

Wordt een signaalnaam gevolgd door een ', dan is het signaal ge-inverteerd. Dit houdt in dat het signaal actief is als er geen spanning op de lijn staat.

N.B. Er zijn printers die de CTS'- en DCD'-signalen niet afleveren, terwijl de ACIA ze wel nodig heeft. Om dit probleem op te lossen kunnen de pennen 4, 5 en 8 van de kabel worden doorverbonden.

10.3 Doorverbindingen

Op de kaart zijn, behalve de schakelaars, enkele instellingen mogelijk van een wat permanentere aard. Deze hebben betrekking op het S-register en de stroombeperking van enkele VIA-pennen.

S-register

Bij SW4 zitten vijf zgn. soldeereilanden. Dit zijn twee vlakjes die met een druppel soldeer doorverbonden kunnen worden. Bij aflevering is dat niet het geval.

De soldeereilanden zijn verbonden met de bovenste vijf bits van het S-register. Een open eiland levert een 1-bit op, een doorverbonden een 0-bit.

N.B. Het ligt in de bedoeling, deze in een toekomstige versie van de Besturingssoftware in gebruik te nemen. Gebruik ervan wordt derhalve afgeraden.

VIA-connector

De weerstanden die de pennen 23, 24 en 25 met een voedingsspanning verbinden, kunnen elk worden kortgesloten met een draadbrug. Zie ook 10.2.1.

Klokhuis Vierlingkaart

P R O B L E E M F O R M U L I E R
= = = = =

Met dit formulier kunt U problemen, vragen en suggesties met betrekking tot de Vierlingkaart opsturen aan:

Stichting Klokhuis
Postbus 427
3200 AK SPIJKENISSE

Betreft Vierlingkaart serienummer: . . .

Afzender

Naam :

adres :

Woonplaats :

Omschrijving:

Gaarne listing mee sturen indien mogelijk

(850607)

Klokhuis Vierlingkaart

Aanhangsel B: Geheugenindeling

=====

Deze tabel geeft de adressen die van belang zijn voor het gebruik van de Vierlingkaart. Op elke regel staat eerst het adres in hexadecimale notatie, dan eventueel tussen haakjes de decimale vorm, en tenslotte de omschrijving.

Page 0

\$3A/3B Werkrumte voor de besturingssoftware.
\$07/08 Werkrumte voor BSB-0 t.b.v. starten van EPROM.

Page 3

\$03FB JMP-instructie naar NMI-interrupt-routine.
\$03FE Adres van IRQ-interrupt-routine.

Parameters voor BSB-0, -1 en -2

\$0478+ (1144) Linkermarge
\$04F8+ (1272) Bladhoogte
\$0578 (1400) EPROM-nummer (via BSB-0).
\$0578+ (1400) Regelteller (intern)
\$05F8 (1528) EPROM-entry.
\$05F8+ (1528) Kolomteller (intern)
\$06F8+ (1784) Regelbreedte
\$0778+ (1912) (ongebruikt)
\$07F8+ (2040) Vlaggen
De adressen waar een '+' achter staat, moeten worden verhoogd met het slotnummer.

Soft Switches

\$C08x+ VIA-registers; zie de documentatie van de VIA (9.3.1).
Deze adressen moeten nog worden verhoogd met 16 maal het slotnummer van de kaart.

Klokhuis Vierlingkaart

I/O-ROM

\$C100 (49408) Slot 1.
 \$C200 (49664) Slot 2.
 \$C300 (49920) Slot 3.
 \$C400 (50176) Slot 4.
 \$C500 (50432) Slot 5.
 \$C600 (50688) Slot 6.
 \$C700 (50944) Slot 7.
 \$Cn00-FD Besturingssoftware (BSBs).
 \$CnFE (49406+256*n) B-register (banknummer).
 \$CnFF (49407+256*n) C-register (keuze uit EPROM, RAM, ACIA, en KLOK).

Extension ROM

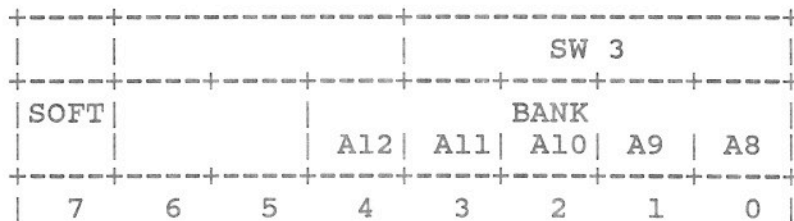
\$C800 (51200) ACIA-dataregisters (TDR/RDR).
 \$C801 (51201) ACIA-statusregister (SR).
 \$C802 (51202) ACIA-commandregister (CMR).
 \$C803 (51203) ACIA-controlregister (CTR).
 \$C804 S-register (SW4).

| | EPROM-pagina | | | | Tevens |
|-----------|--------------|----|----|----|-------------|
| \$C800 | 00 | 08 | 16 | 24 | ACIA |
| \$C900 | 01 | 09 | 17 | 25 | |
| \$CA00 | 02 | 10 | 19 | 26 | |
| \$. . . . | | | | | |
| \$CF00 | 07 | 15 | 23 | 31 | Klok of RAM |

\$CF00-FC RAM (zie C-register).
 \$CF00-3F RAM gebruikt door besturingssoftware.
 \$CFFD (53245) D-register (klok-data).
 \$CFFE (53246) K-register (klok-besturing).
 \$CFFF (53247) ROMSWITCH: schakelt Extension ROM uit (Apple Hardware-protocol).

Klokhuis Vierlingkaart

B-register: bank (\$CnFE)



bit naam

- 7 SOFT Geeft aan of het banknummer door de SW3 wordt bepaald (0) of door het programma (1).
- 4-0 BANK Nummer van de bank (256 bytes) van de Besturings-software (adreslijn A8-A12 van de besturings-EPROM).

Schakelen geschiedt door de waarde naar het gebied \$Cnxx te schrijven, dus naar de besturings-EPROM. Bij afspraak wordt echter alleen adres \$CnFE gebruikt. Adres \$CnFF kan niet gebruikt worden omdat daar het C-register zit. Om na te gaan, welke BSB er actief is, kan het B-register worden uitgelezen. Strict genomen wordt daarbij uit de Besturings-EPROM gelezen. Deze bevat op adres \$CnFE altijd het banknummer.

C-register: kaartbesturing (\$CnFF)



Het C-register kan zowel worden gelezen als geschreven. Na aanzetten van de computer of RESET staan alle bits op 0. De betekenis van de bits is:

Klokhuis Vierlingkaart

bit naam

7 RAM Geeft aan of \$CFxx naar RAM (1) of naar EPROM (0) wijst. Om de RAM te activeren, moet er een EPROM-nummer zijn ingevuld in bit 2-0.

6 RAMH Adreslijn 9 van RAM.

5 HOOG Adreslijn 12 van EPROM, adreslijn 8 van RAM.

4 LAAG Adreslijn 11 van EPROM.

3 CARD Schakelt de kaart in.

2-0 Selecteert EPROM, KLOK, of ACIA.
Dit werkt alleen als CARD (bit 3) op 1 staat.

0: geen EPROM.
1: EPROM 1.
.. ..
5: EPROM 5.
6: KLOK.
7: ACIA.

Bits 5 en 6 vormen het nummer van het actieve RAM-deel (256 bytes).

Bits 4 en 5 vormen het nummer van het actieve EPROM-deel (2048 bytes).

Hieruit volgt dat vanuit een EPROM-deel niet alle RAM-delen te gebruiken zijn. Het verband is:

| EPROM-deel | | | RAM-deel | | |
|------------|------------|------------|----------|------------|------------|
| nr | HOOG b5 | LAAG b4 | nr | RAMH b6 | HOOG b5 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 |
| 2 | 1 | 0 | 1 | 0 | 1 |
| 3 | 1 | 1 | 1 | 1 | 1 |
| | | | 3 | 0 | 1 |
| | | | 3 | 1 | 1 |

Klokhuis Vierlingkaart

KLOK

=====

K-register (\$CFFE) (schrijven)

| INT | READ | ADJ | HOLD | REG | | | |
|-----|------|-----|------|-----|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

- 7 INT 1: Interrupt mogelijk. Zie SW7.
0: Zet de interrupt-indicatie af.
- 6 READ Lees de klok uit.
- 5 ADJ Stel de klok in op de dichtstbijzijnde hele minuut.
- 4 HOLD Zet de klok stil (max 150 microseconden).
- 3-0 ADDR Nummer van intern register.

D-register (\$CFFD) (lezen)

| INT | READ | ADJ | HOLD | DATA | | | |
|-----|------|-----|------|------|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

- 7 INT Interrupt vanuit klok.
- 6 READ De klok wordt gelezen.
- 5 ADJ De klok wordt gelijkgezet.
- 4 HOLD De klok staat stil.
- 3-0 DATA Gegevens uit een intern klokregister.

D-register (\$CFFD) (schrijven)

| - | | | | DATA | | | |
|---|---|---|---|------|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

- 7-4 ---- Niet via D-register te wijzigen.
- 3-0 DATA Gegevens voor de klok.

Klokhuis Vierlingkaart

Bank-switch-routine

Deze routine is de verwezenlijking van het Bank Switchings-mechanisme dat centraal staat in de opzet van de Vierlingkaart.

Deze routine staat in elke bank van de Besturingssoftware.

De stappen die ondernomen moeten worden om een andere bank te activeren, zijn:

- 1) Zet het adres binnen de te activeren bank in byte \$3A (lage deel) en \$3B (hoge deel).
- 2) Vul de accumulator met het banknummer.
- 3) Spring naar adres \$CnF5.

De routine ziet er als volgt uit:

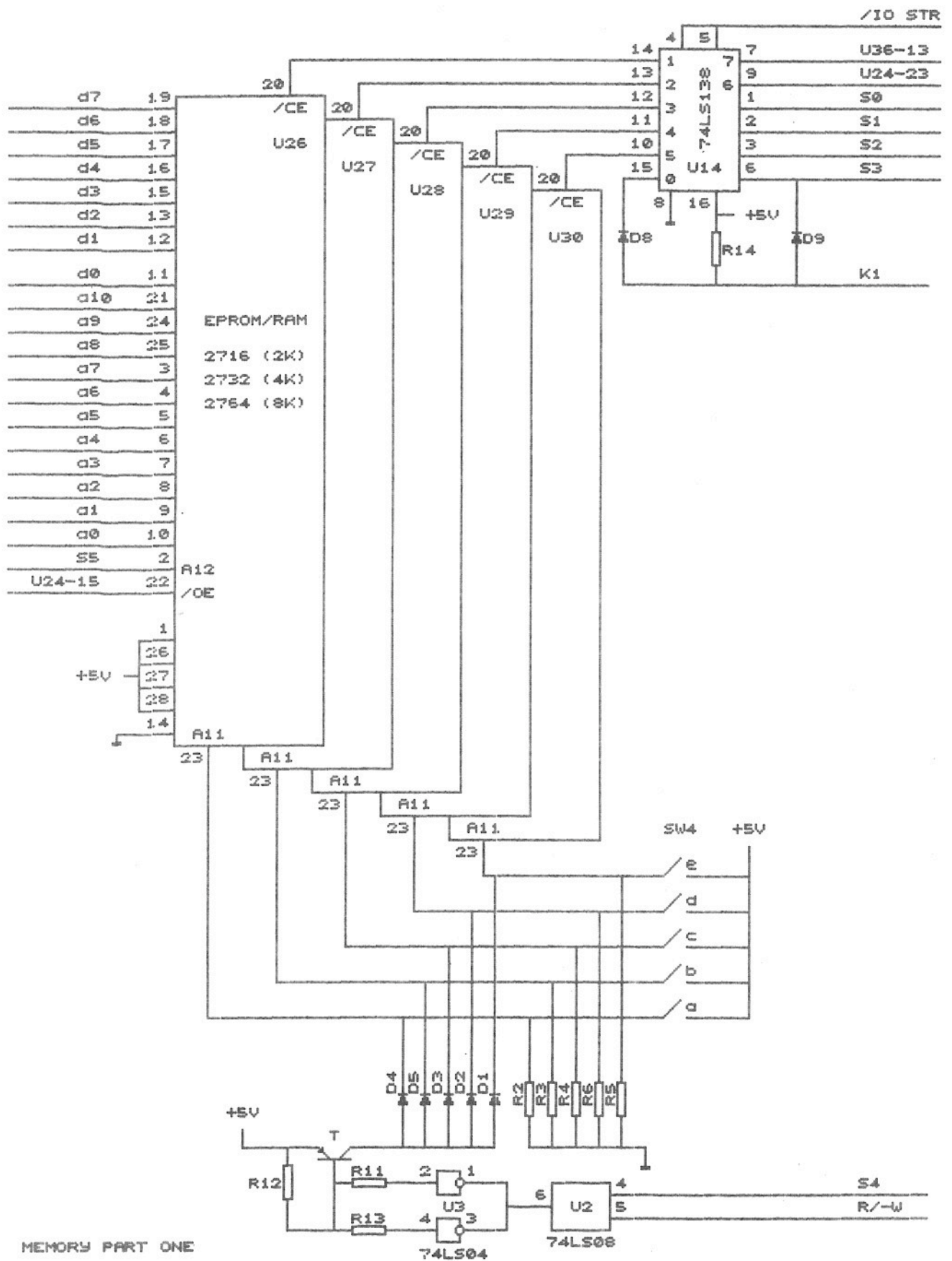
| Adres | Instructie |
|---------|---|
| \$CnF5: | LDY #0 |
| \$CnF7: | ORA #\$80 |
| \$CnF9: | STA (\$3A),Y ;schakel om door te ;schrijven naar het ;B-register. |
| | :* De volgende JMP-instructie springt naar |
| | :* het gewenste adres in de <u>nieuwe</u> bank. |
| \$CnFB: | JMP (\$3A) |
| \$CnFE: | DFB banknummer van deze bank. |
| \$CnFF: | ;niet bruikbaar. |

Klokhuis Vierlingkaart

Aanhangsel C: Schema

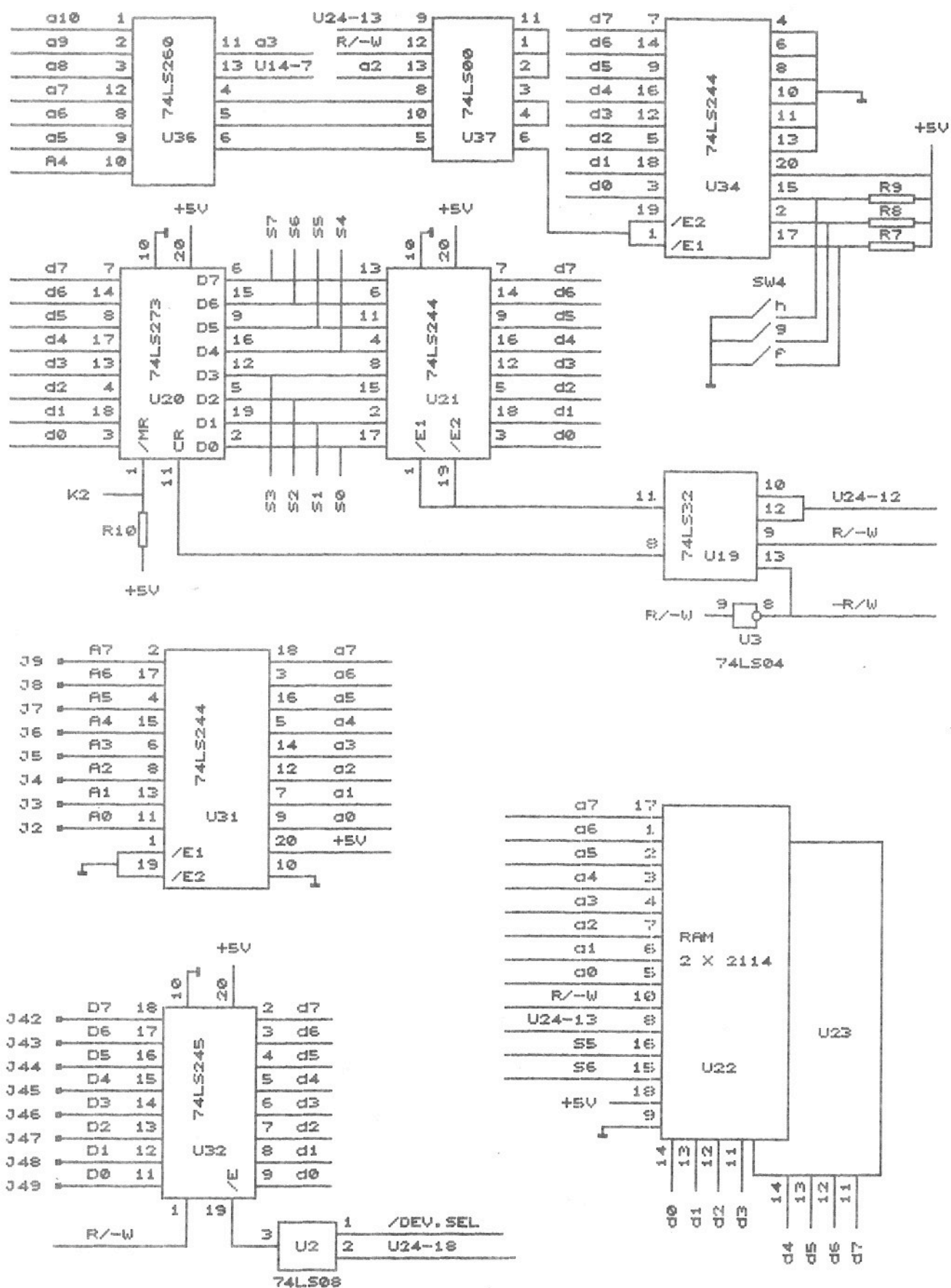
=====

Klokhuis Vierlingkaart



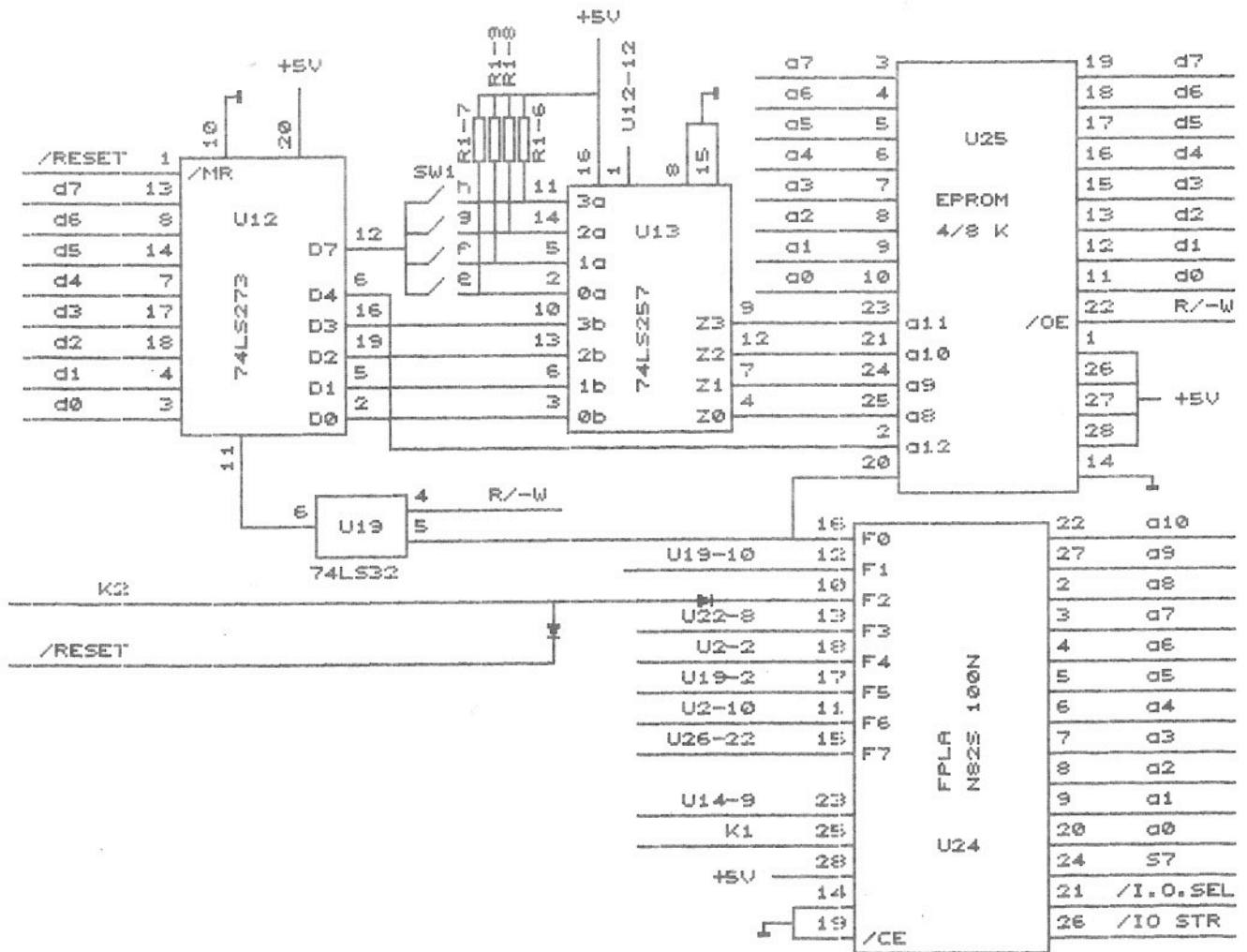
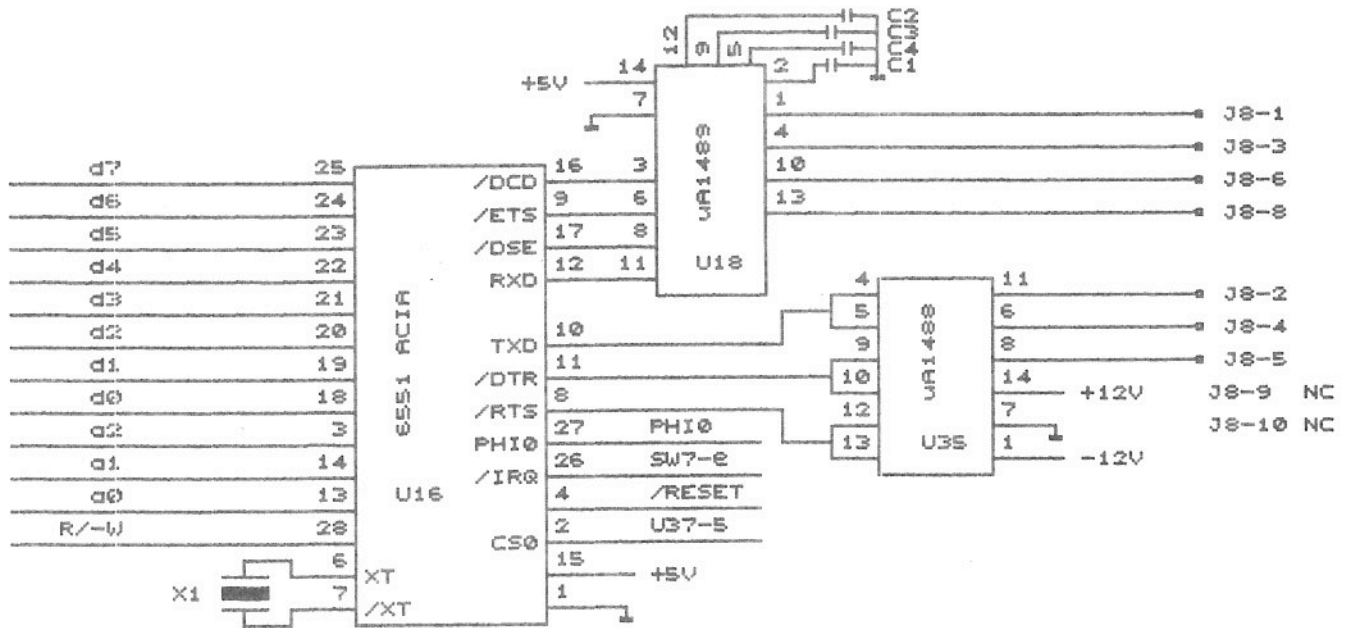
MEMORY PART ONE

Klokhuis Vierlingkaart

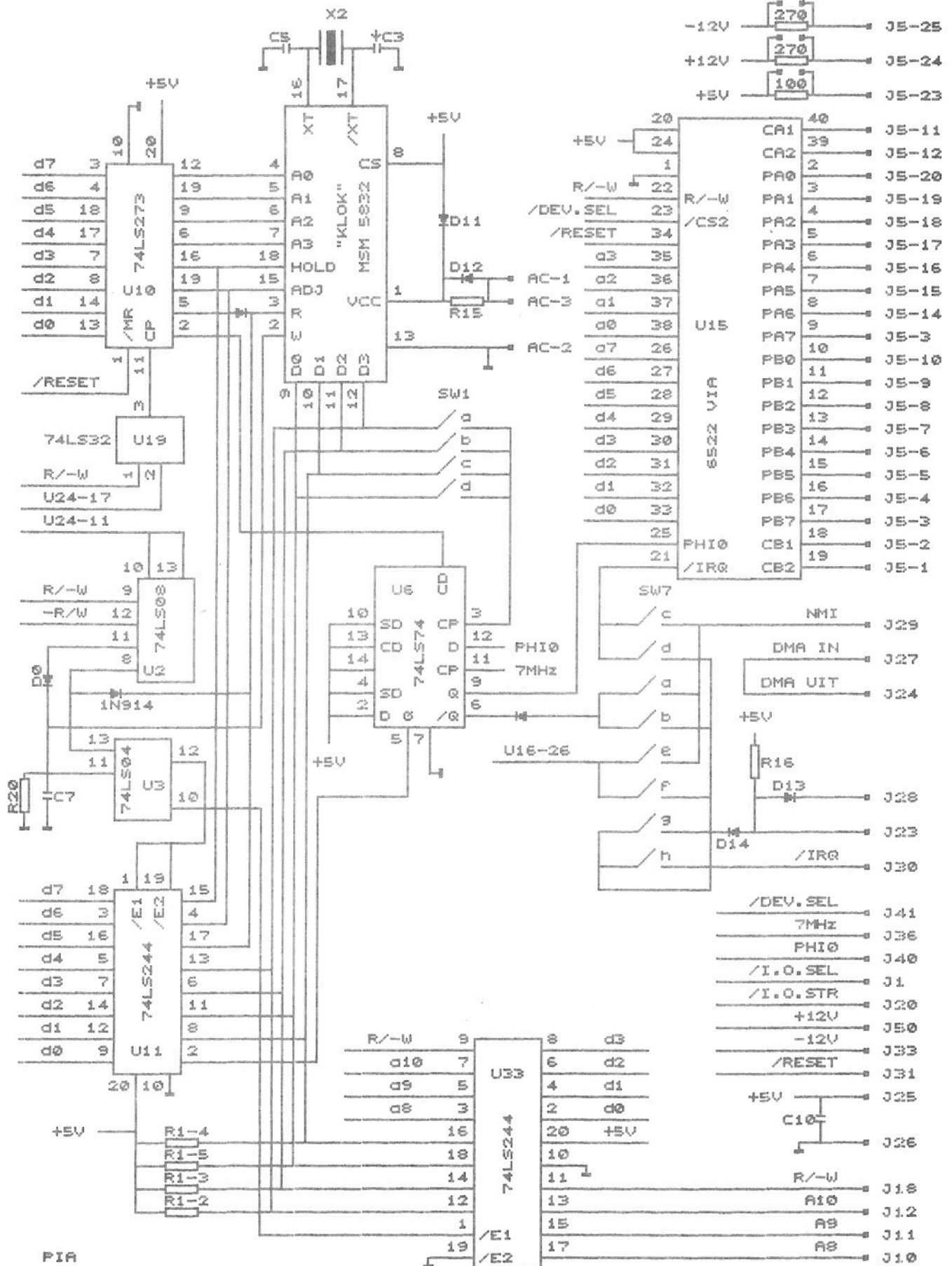


MEMORY PART TWO

Klokhuis Vierlingkaart



Klokhuis Vierlingkaart



Klokhuis Vierlingkaart

IC'S

| | | |
|-------------------------|---|----------------|
| U2 | = | 74LS08 |
| U3 | = | 74LS04 |
| U6 | = | 74LS74 |
| U9 | = | MSM832 |
| U10, U12, U20 | = | 74LS273 |
| U11, U21, U31, U33, U34 | = | 74LS244 |
| U13 | = | 74LS257 |
| U14 | = | 74LS138 |
| U15 | = | 6522 |
| U16 | = | 6551 |
| U18 | = | UA1489 |
| U19 | = | 74LS32 |
| U22, U23 | = | 2114 |
| U24 | = | N82S100 |
| U25.....U30 | = | 2716/2732/2764 |
| U32 | = | 74LS245 |
| U35 | = | UA1488 |
| U36 | = | 74LS260 |
| U37 | = | 74LS00 |

TRANSISTOR'S

T = 2N2907

DIPSWITCH

SW1, SW2, SW4 = 8* 1 POLIGE SCHAKELAARS

KRISTAL

X1 = 32.768 Mhz
X2 = 1.8432 Mhz

CONDENSATOR'S

C1.....C4 = 100 PF
C5 = 20 PF
C7 = 330 PF
C8.....C15 = 100 nF
C16 = 4u7 TANTAAL
C = INSTEL 3-18 PF

DIODE'S

D0.....D16 = 1N914

WEERSTANDEN

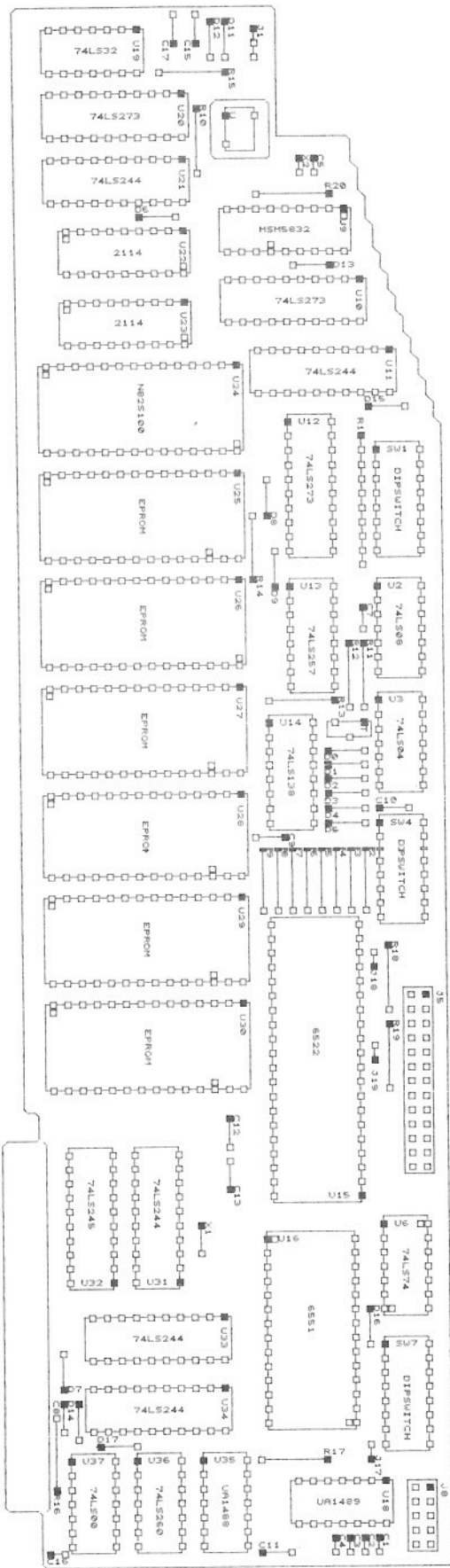
R1 = ARRAY 9*10K
R2....R6 = 1K
R7....R9, R16 = 10K
R10, R11, R13 = 1K
R12 = 330
R14 = 15K
R15, R18 = 100
R17, R19 = 270
R20 = 2K2

PENNENREKKEN

J5 = 2*13
J8 = 2*5
J1 = 1*3
J17.....J19 = 1*2

COMPONENTENLIJST

Klokhuis Vierlingkaart



VIERLINGKAART BESTURINGSSOFTWARE

```
4 *****
5 *
6 *      K L O K H U I S      *
7 *    V I E R L I N G K A A R T  *
8 *
9 *      B E S T U R I N G S S O F T W A R E      *
10 *    V E R S I E   5.0/850607      *
11 *
12 *****
13 *
14 *
15 *
16 *
17 *
18 *      O R G   $ 7 0 0 0      ;
19 *
```


VIERLINGKAART BESTURINGSSOFTWARE

```

>1      *
>2      *
>3      *****
>4      *                                     *
>5      *   A L G E M E N E   E Q U   *
>6      *                                     *
>7      *****
>8      *                                     *
>9      *           VERSIE 5.0/850607       *
>10     *                                     *
>11     *****
>12     *
>13     *
>14     *Monitor adressen
>15     *
>16     IORTS   =   $FF58           ;bekende RTS
>17     BELL    =   $FBDD           ;BELL routine
>18     COUT1   =   $FDF0           ;character op scherm
>19     COUT    =   $FDED           ;uitvoer character
>20     CROUT   =   $FD8E           ;nieuwe regel
>21     RDKEY   =   $FDOC           ;invoer character(geen ESC)
>22     RDCHAR  =   $FD35           ;invoer character(wel ESC)
>23     SETKBD  =   $FE89
>24     HOME    =   $FC58           ;maak scherm schoon
>25     WAIT    =   $FCA8           ;wacht routine
>26     *
>27     *
>28     *Algemene I/O adressen
>29     *
>30     KBD     =   $C000           ;Keyboard
>31     KBDSTRB =   $C010           ;Keyboard reset strobe
>32     CLRROM  =   $CFFF           ;Schakelt extensie ROM-en uit
>33     *
>34     *
>35     *Pagina nul
>36     *
>37     INDIRECT =   $3A           ;Indirecte adressering
>38     INDIRECL =   INDIRECT
>39     INDIRECH =   INDIRECT+1
>40     INDGOOR  =   $08           ;indirecte gore adressering (BAN
>41     INDGOORL =   INDGOOR
>42     INDGOORH =   INDGOOR+1
>43     CH       =   $24           ;horizontale positie cursor
>44     CSW      =   $36
>45     CSWL     =   CSW           ;Uitvoer vector
>46     CSWH     =   CSW+1
>47     KSW      =   $38
>48     KSWL     =   KSW           ;Invoer vector
>49     KSWH     =   KSW+1
>50     *
>51     *
>52     *Algemene adressen
>53     *
>54     STACK   =   $100           ;Systeem stack
>55     *

```

VIERLINGKAART BESTURINGSSOFTWARE

```

>56 *Screen holes
>57 *
>58 * Algemeen & werkruimte
>59 *
>60 INITF = $478 ;Initiele vlag voor par. driver
>61 EPROMNR = $578 ;Te activeren EPROM
>62 ENIRYNR = $5F8 ;Hoeveelste entrypunt in EPROM
>63 NO = $6F8 ;Slotnummer maal 16(NO)
>64 SCRAT1 = $778 ;kladruimte
>65 MSLOT = $7F8 ;Hoog adres van slotadres
>66 *
>67 * Parallele printer screen holes
>68 *
>69 P_MODE = $7F8-$C0 ;Stuurbits:B7-schermecho
>70 * ;B6-auto LF B5-FF generatie
>71 P_REGLN = $6F8-$C0 ;Regelbreedte
>72 P_IMARGE = $478-$C0 ;Linker marge
>73 P_PAGLEN = $4F8-$C0 ;#bedrukte regels per blad
>74 P_LINE = $578-$C0 ;#aantal CR's sinds laatste FF
>75 P_COL = $5F8-$C0 ;#characters sinds laatste CR
>76 *
>77 * Seriele printer screen holes
>78 *
>79 S_MODE = $7F8-$C0 ;Stuurbits:B7-schermecho
>80 * ;B6-auto LF B5-FF generatie
>81 S_REGLN = $6F8-$C0 ;Regelbreedte
>82 S_IMARGE = $478-$C0 ;Linker marge
>83 S_PAGLEN = $4F8-$C0 ;#bedrukte regels per blad
>84 S_LINE = $578-$C0 ;#aantal CR's sinds laatste FF
>85 S_COL = $5F8-$C0 ;#characters sinds laatste CR
>86 *
>87 *Printer instellingen
>88 *
>89 PAGLENR = 0 ;aantal regels per blad -8
>90 PARREGLN = $FF ;aantal karakters per regel (PAR
>91 SERREGLN = $FF ;aantal karakters per regel (SER
>92 PARMODE = %00000000 ;Geen ECHO Geen auto LF Geen FF
>93 SERMODE = %01000000 ;Geen ECHO auto LF Geen FF
>94 *
>95 *
>96 *karakters
>97 *
>98 CTRL = 0-'@' ;Gebruik: CTRL+"char"
>99 BS = CTRL+"H" ;backspace
>100 LF = $8A ;linefeed
>101 FF = $8C ;formfeed
>102 CR = $8D ;carriage return
>103 CTRL-L = CTRL+"L" ;control-L
>104 CTRL-Q = CTRL+"Q" ;control-Q
>105 CTRL-S = CTRL+"S" ;control-S
>106 ESC = $9B ;Escape
>107 SPATIE = " " ;spatie
>108 *
>109 *
>110 *VIERLINGKAART EQUATES

```

VIERLINGKAART BESTURINGSSOFTWARE

```

>111 *
>112 RETADR = $CF00 ;Terugkeer adres
>113 SLOTADR = $CF01 ;Hoge adresdeel slot
>114 RETBANK = $CF02 ;Terugkeer bank
>115 SAVCREG = $CF03 ;Bewaarpositie CREG
>116 CTRSMODE = $CF04 ;Control-S actief positie
>117 TDATA TEL = $CF05 ;te verzenden aantal klokkarakte
>118 KLOKDATA = $CF10 ;18 bytes buffer voor klokdata
>119 * ;layout:UU;MM;SS JJMMDD W<CR>
>120 *
>121 *De volgende waarden zijn allen $CnXX
>122 *en kunnen alleen via indirecte LDA en STA
>123 *bereikt worden
>124 *
>125 BANKNR = $FE ;Laag adres van banknummer
>126 CREGL = $FF ;Laag adres controleregister
>127 *
>128 *
>129 *Slot onafhankelijke adressen
>130 *
>131 *ACIA adressen
>132 *
>133 ACTIADATA = $C800 ;ACIA data register
>134 ACTIASTAT = $C801 ;ACIA status register
>135 ACTIACOMM = $C802 ;ACIA commando register
>136 ACTIACONT = $C803 ;ACIA controle register
>137 SREG = $C804 ;ACIA snelheidsregister (BPS)
>138 INITCOMN = %11001011 ;commando vulling NO PARITY
>139 INITCOME = %01101011 ;commando register EVEN PARITY
>140 INITCON7 = %00110000 ;GEEN BPS 7 bits
>141 INITCON8 = %00010000 ;geen BPS 8 bits
>142 TDRE = %00010000 ;Transmit Data Register Empty
>143 RDRF = %00001000 ;Receive Data Register Full
>144 *
>145 *KLOK adressen
>146 *
>147 DREG = $CFFD ;Data register klok
>148 KREG = $CFFE ;Klok besturings register
>149 KLOKLANG = $11 ;lengte klokstring incl CR
>150 *
>151 *GEBRUIKERS EPROM adressen
>152 *
>153 EPROM = $C800 ;gebruikers EPROM
>154 ENTRYANT = $C804 ;Aantal entrypunts in de EPROM
>155 ENTRYTAB = $C805 ;Begin entrypunt tabel
>156 *
>157 *VIA-adressen, bereikbaar via ADRES,Y
>158 * waarbij Y het slotnummer * 16 bevat,
>159 *
>160 VIA = $C080
>161 VIA_IORB = VIA+0 ;I/O-poort B
>162 VIA_IORA = VIA+1 ;I/O-poort A
>163 VIA_DDRB = VIA+2 ;Data directieregister B
>164 VIA_DDRA = VIA+3 ;Data directieregister A
>165 VIA_ACR = VIA+$B ;Auxiliary control

```

VIERLINGKAART BESTURINGSSOFTWARE

```
>166 VIA_PCR = VIA+$C ;Peripheral control
>167 VIA_IFR = VIA+$D ;Interrupt flag register
>168 VIA_IER = VIA+$E ;Interrupt enable register
>169 *
>170 SOFTBANK = %10000000 ;Bankswitching door software bit
```

VIERLINGKAART BESTURINGSSOFTWARE

```

>172 RAMACTIV = %10001110 ;RAM, klok en kaart ingeschakeld
>173 * ;en EEN EPROM RAM werkt anders n
>174 ACIA_ON = %10001111 ;RAM,ACIA en kaart aan
>175 *
>176 *
>177 *BANKNUMMERS
>178 *
>179 ENKALG = $0
>180 ENKALG1 = $1B
>181 ENKPAR = $1
>182 ENKPAR1 = $1D
>183 ENKSER = $2
>184 ENKSER1 = $1C
>185 ENKTERM = $3
>186 ENKTERML = $19
>187 ENKTERM2 = $1A
>188 ENKKLOKR = $4
>189 ENKKLOK1 = $1E
>190 ENKKLOKP = $1F
>191 ENKKLOKS = $5
>192 ENKRES6 = $6
>193 ENKRES7 = $7
>194 ENKRES8 = $8
>195 ENKRES9 = $9
>196 ENKRESA = $A
>197 ENKRESB = $B
>198 ENKRESC = $C
>199 ENKRESD = $D
>200 ENKRESE = $E
>201 ENKRESF = $F
>202 ENKRES10 = $10
>203 ENKRES11 = $11
>204 ENKRES12 = $12
>205 ENKRES13 = $13
>206 ENKRES14 = $14
>207 ENKRES15 = $15
>208 ENKRES16 = $16
>209 ENKRES17 = $17
>210 ENKRES18 = $18
>211 *

```

VIERLINGKAART BESTURINGSSOFTWARE

```

>1 *****
>2 *****
>3 ***                               ***
>4 ***   B A N K   N R   0   ***
>5 ***                               ***
>6 *****
>7 *****
>8 *                               *
>9 * bank BNKALG VERSIE 5.0/850607 *
>10 *                               *
>11 *****
>12 *
>13 *****
>14 *                               *
>15 *   A L G E M E N E   B A N K   *
>16 * G E B R U I K E R S K E U Z E *
>17 *                               *
>18 *****
>19 *
>20 *
>21 *
7000: 08   >22   PHP                               ;Bewaar status
7001: 78   >23   SEI                               ;Disable interrupten
7002: 20 58 FC >24   JSR HOME                       ;Geeft $Cn op stack
7005: BA   >25   TSX
7006: BD 00 01 >26   LDA STACK,X                   ;Pak $Cn
7009: 8D F8 07 >27   STA MSLOT                       ;Bewaar $Cn in tekstpage
700C: 85 3B   >28   STA INDIRECTH                   ;en voor indirect werk
700E: AA   >29   TAX                               ;bewaar even
700F: A9 00   >30   LDA #$0                               ;Maak indirect adres af
7011: 85 3A   >31   STA INDIRECTL
7013: A0 FF   >32   LDY #CREGL                       ;Zet de kaart en de RAM aan
7015: A9 8E   >33   LDA #RAMACTIV
7017: 91 3A   >34   STA (INDIRECT),Y
7019: 8E 01 CF >35   STX SLOTADR                       ;Bewaar in RAM
701C: 28   >36   PLP                               ;Interrupten mogen weer
>37 *
>38 *we resetten de hooks naar IN#0
701D: 20 89 FE >39   JSR SETKBD                       ;zet IN#0
>40 *
>41 *Vraag of gebruikers EPROM geactiveerd moet worden
7020: A9 97   >42   LDA #<VRAAGROM ;karakters moeten INDIRECT
7022: 85 3A   >43   STA INDIRECTL ;geladen worden
7024: A0 00   >44   LDY #0
7026: B1 3A   >45   VRAAGOP LDA (INDIRECT),Y ;verzend vraag
7028: F0 06   >46   BEQ RDRMJN
702A: 20 ED FD >47   JSR COUT
702D: C8   >48   INY
702E: D0 F6   >49   BNE VRAAGOP
>50 *
7030: 20 35 FD >51   RDRMJN JSR RDCHAR                       ;haal invoer
7033: 20 ED FD >52   JSR COUT                       ;echo letter
7036: 29 DF   >53   AND #$DF                       ;maak hoofdletter
7038: C9 CA   >54   CMP #"J"                       ;alleen op Ja controle
703A: D0 51   >55   BNE VOLGEND                       ;volgend onderdeel

```

VIERLINGKAART BESTURINGSSOFTWARE

```

>56 *
703C: A9 B8 >57 LDA #<ROMMES ;INDIRECT voorbereiden
703E: 85 3A >58 STA INDIRECL
7040: A0 00 >59 LDY #$0
7042: B1 3A >60 ROMVRAAG LDA (INDIRECT),Y ;verzend bericht
7044: F0 09 >61 BEQ RDROMNR
7046: 20 ED FD >62 JSR COUT
7049: C8 >63 INY
704A: D0 F6 >64 BNE ROMVRAAG
>65 *
704C: 20 DD FB >66 RDROMERR JSR BELL ;foutieve invoer
704F: 20 35 FD >67 RDROMNR JSR RDCHAR ;haal invoer
7052: C9 B0 >68 CMP #"0" ;controleer
7054: 90 F6 >69 BLT RDROMERR ;fout
7056: C9 B6 >70 CMP #"6"
7058: 10 F2 >71 BPL RDROMERR ;fout
705A: 8D 78 05 >72 STA EPROMNR ;bewaar
705D: 20 ED FD >73 JSR COUT ;echo invoer
7060: 29 0F >74 AND #$0F ;was het EPROM 0
7062: F0 24 >75 BEQ USERENT1 ;Ja, geen entrypunt dus
>76 *
7064: A9 D2 >77 LDA #<ENTRYMES ;INDIRECT voorbereiden
7066: 85 3A >78 STA INDIRECL
7068: A0 00 >79 LDY #$0
706A: B1 3A >80 ENTRYVR LDA (INDIRECT),Y ;verzend boodschap
706C: F0 09 >81 BEQ RENTRY
706E: 20 ED FD >82 JSR COUT
7071: C8 >83 INY
7072: D0 F6 >84 BNE ENTRYVR
>85 *
7074: 20 DD FB >86 RDENTERR JSR BELL ;fout
7077: 20 35 FD >87 RENTRY JSR RDCHAR ;haal invoer
707A: C9 B1 >88 CMP #"1" ;controleer
707C: 90 F6 >89 BLT RDENTERR
707E: C9 BA >90 CMP #"9"+1
7080: 10 F2 >91 BPL RDENTERR
7082: 8D F8 05 >92 STA ENTRYNR ;bewaar in screen hole
7085: 20 ED FD >93 JSR COUT ;echo op scherm
>94 *
>95 *Het starten van de EPROM-en wordt
>96 *in BANK A afgehandeld
7088: A9 42 >97 USERENT1 LDA #<B_EPROM
708A: D0 03 >98 BNE VOLGEND1 ;altijd
>99 *
>100 *
>101 *terugkeer punt
708C: 60 >102 ALGRIS RTS ;terug naar roeper
>103 *
>104 *in andere bank verder
708D: A9 00 >105 VOLGEND LDA #<B_ALG1 ;volgende bank entry
708F: 85 3A >106 VOLGEND1 STA INDIRECL
7091: A9 1B >107 LDA #BNKALG1 ;volgende bank
7093: D0 60 >108 BNE SCHAKELO ;op naar de volgende
>109 *opvullen tot bankswitch op zijn plaats is
>110 *

```

VIERLINGKAART BESTURINGSSOFTWARE

```

7095: FF FF >111          HEX  FFFF
      >112 *
      >113 *
      >114 *boodschappen
      >115 *
7097: 8C 8D 8D >116 VRAAGROM DFB  CTRL,CR,CR,CR
709B: C7 C5 C2 >117          ASC  "GEBRUIKERSEPROM STARTEN J/N?"
70B7: 00      >118          DFB  $0
70B8: 8D 8D   >119 ROMMES  DFB  CR,CR
70BA: C7 C5 C5 >120          ASC  "GEEF EPROM NUMMER(0-5)?"
70D1: 00      >121          DFB  $0
      >122 *
70D2: 8D 8D   >123 ENRYMES DFB  CR,CR
70D4: C7 C5 C5 >124          ASC  "GEEF ENRYPOINT NUMMER(1-9)?"
70F0: 00      >125          DFB  $0
      >126 *
      >127 *op dit punt kan de gebruiker inspringen
      >128 *om programatisch een programma
      >129 *te laten uitvoeren, mits EPROMNR en
      >130 *ENTRYNR zijn ingevuld
70F1: B0 95   >131 USERENT BCS  USERENT1
70F3: 90 93   >132          BCC  USERENT1 ;altijd genomen
      >133 *
      >134 *
      >135 *****
      >136 *
      >137 *Schakel naar andere BANK
      >138 *
      >139 *****
      >140 *
      >141 *Schakel andere bank in door het
      >142 *gewenste banknummer naar adresgebied
      >143 *$Cn00-$CnFE te schrijven, banknummer
      >144 *staat reeds in ACCU
      >145 *In adres INDIRECT staat reeds waar naar
      >146 *toe gesprongen wordt bij binnenkomst
      >147 *andere bank
      >148 *
      >149 *Y-register wordt gebruikt
      >150 *
70F5: A0 00   >151 SCHAKELO LDY  #$0
70F7: 09 80   >152          ORA  #SOFTBANK ;Zet softbankselectie bit
70F9: 91 3A   >153          STA  (INDIRECT),Y ;Schakel om
70FB: 6C 3A 00 >154          JMP  (INDIRECT) ;Spring weg bij binnenkomst
      >155 *
70FE: 00      >156          DFB  BNKALG ;Nummer van deze BANK
70FF: FF      >157          DFB  $FF ;Adres Controle Register
      >158 *

```


VIERLINGKAART BESTURINGSSOFTWARE

```

>1 *****
>2 *****
>3 *** ***
>4 *** BANK NR 1 ***
>5 *** ***
>6 *****
>7 *****
>8 * *
>9 * bank ENKPAR VERSIE 5.0/850607 *
>10 * *
>11 *****
>12 *
>13 *****
>14 * *
>15 * PRINTERDRIVER *
>16 * PARALLEL *
>17 * *
>18 *****
>19 *
>20 *
>21 *
7100: 18 >22 CLC ;clear CARRY=initiele entry
7101: B0 >23 DFB $B0 ;=BCS nooit genomen
>24 *
7102: 38 >25 P_OUT SEC ;zet carry=output
7103: 90 03 >26 BCC P_SETOVL ;initiele entry
7105: B8 >27 CLV ;niet initieel clear overflow
7106: 50 03 >28 BVC P_ENTRY ;altijd
>29 *
7108: 2C 58 FF >30 P_SETOVL BIT IORIS ;zet overflow flag=initiele entr
>31 *
710B: 48 >32 P_ENTRY PHA ;bewaar karakter
710C: 8A >33 TXA ;bewaar XREG
710D: 48 >34 PHA
710E: 98 >35 TYA ;bewaar YREG
710F: 48 >36 PHA
7110: 08 >37 PHP ;bewaar status
7111: 78 >38 SEI ;disable interrupten even
7112: 20 58 FF >39 JSR IORIS ;bekende RTS positie
7115: BA >40 TSX ;nú staat slotadres op stack($Cn
7116: BC 00 01 >41 LDY STACK,X ;pak hoog slotadres $Cn
7119: 8C F8 07 >42 STY MSLOT ;bewaar in screenhole
711C: 84 3B >43 STY INDIRECT ;en in INDIRECT register
711E: A9 00 >44 LDA #$0 ;completeer INDIRECT register
7120: 85 3A >45 STA INDIRECT
7122: 68 >46 PLA ;ga karakter halen
7123: 68 >47 PLA
7124: 68 >48 PLA
7125: 68 >49 PLA
7126: 9A >50 TXS ;zet stackpointer terug
7127: 48 >51 PHA ;bewaar karakter weer
7128: 98 >52 TYA ;ga van $Cn=slotnummer $n0 maken
7129: AA >53 TAX ;bewaar $Cn in XREG
712A: 0A >54 ASL
712B: 0A >55 ASL

```

VIERLINGKAART BESTURINGSSOFTWARE

```

712C: 0A      >56      ASL
712D: 0A      >57      ASL
712E: 8D F8 06 >58      STA NO          ;bewaar in screenhole
7131: A9 00    >59      LDA #0          ;geef aan dat dit standaard
7133: 8D 78 04 >60      STA INITF      ;niet de eerste maal is
>61      *zet de kaart en RAM aan
7136: A0 FF    >62      LDY #CREGL
7138: B1 3A    >63      LDA (INDIRECT),Y ;haal CREG op
713A: 48      >64      PHA           ;bewaar even, RAM eerst aan
713B: 8D FF CF >65      STA CLRROM    ;schakel extensie ROM-en uit
713E: A9 8E    >66      LDA #RAMACTIV ;kaart en RAM aan
7140: 91 3A    >67      STA (INDIRECT),Y ;schakel in
>68      *
7142: A0 FE    >69      LDY #BANKNR
7144: B1 3A    >70      LDA (INDIRECT),Y ;haal huidig banknummer
7146: 8D 02 CF >71      STA RETBANK   ;bewaar voor terugkeer
7149: 8E 01 CF >72      STX SLOTADR  ;bewaar $Cn ook in RAM
714C: 68      >73      PLA           ;haal oude CREG terug
714D: 8D 03 CF >74      STA SAVCREG   ;bewaar die
7150: 68      >75      PLA           ;ga status terughalen
7151: 28      >76      PLP           ;voor enable interrupt
7152: 48      >77      PHA           ;bewaar karakter toch op stack
7153: 50 4F    >78      BVC P_NOINIT  ;geen initiele entry
>79      *
>80      *initiele eerste maal binnenkomst
>81      *
7155: B8      >82      CLV           ;clear de overflow
7156: 18      >83      CLC           ;nodig om input te voorkomen
>84      *in XREG $Cn, controleer op input, zo ja eruit
7157: E4 39    >85      CPX KSWH     ;is het input
7159: F0 65    >86      BEQ P_ISINP  ;Ja, gelijk er maar weer uit
715B: E4 37    >87      CPX CSWH     ;is het wel PR#
715D: D0 04    >88      BNE P_VERDER ;nee
>89      *
715F: A9 02    >90      LDA #<P_OUT  ;pas de output hook aan
7161: 85 36    >91      STA CSWL
>92      *
7163: A9 00    >93      P_VERDER LDA #0
7165: 9D B8 04 >94      STA P_LINE,X ;#CR's sinds laatste nieuwe blad
7168: 9D B8 03 >95      STA P_IMARGE,X ;geen linker marge inspringen
716B: A9 00    >96      LDA #PARMODE ;B7=video,B6=auto LF,B5=FF gener
716D: 9D 38 07 >97      STA P_MODE,X ;video uit, auto LF uit en FF ui
7170: A9 00    >98      LDA #PAGLENNR ;aantal bedrukte regels per blad
7172: 9D 38 04 >99      STA P_PAGLEN,X
7175: A9 FF    >100     LDA #PARREGLN ;posities per regel
7177: 9D 38 06 >101     STA P_REGLEN,X ;maximale regellengte
>102     *Ga kijken of er soms toch een ACK van de
>103     *printer komt, kan bij eigen printerdrivers
>104     *en bij vlug achter elkaar inschakelen
>105     *Bij echte initiele binnenkomst staat alles
>106     *even te wachten
717A: A9 A0    >107     LDA #SAO     ;buitenste wachtloop
717C: 8D 78 04 >108     STA INITF    ;bewaar even
717F: 8A      >109     TXA         ;bewaar XREG
7180: 48      >110     PHA

```

VIERLINGKAART BESTURINGSSOFTWARE

```

7181: A2 00 >111          LDX  #$0          ;zet XREG
7183: AC F8 06 >112          LDY  NO           ;pak slot offset
7186: B9 8D C0 >113 P_WLOOP LDA  VIA_IFR,Y  ;pak interruptvlag
7189: 29 10 >114          AND  #%00010000  ;VIA al gereed
718B: D0 08 >115          ENE  P_INITEM    ;ja hoor
718D: CA >116            DEX              ;verlaag teller
718E: D0 F6 >117          ENE  P_WLOOP     ;nogmaals
7190: CE 78 04 >118          DEC  INITF       ;verminder hoofdloop
7193: D0 F1 >119          ENE  P_WLOOP     ;nogmaals
7195: 68 >120          P_INITEM PLA     ;haal XREG terug
7196: AA >121            TAX              ;
7197: A9 FF >122          LDA  #$FF        ;zet VIA-poort op output
7199: 8D 78 04 >123          STA  INITF       ;geef aan eerste maal
719C: 99 82 C0 >124          STA  VIA_DDRB,Y ;zet in VIA
>125 *PCR gebruik
>126 *
>127 * | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
>128 * |CB2 Control|CBI|CA2 Control|CA1|
>129 * bit functie
>130 * 7654
>131 * 100 Handshake: CB2:=0 bij schrijven naar ORB;
>132 * CB2:=1 bij omslaan van CBI.
>133 * 101 Pulse output: CB2:=0 (1 cyclus)
>134 * bij aanspreken van ORB.
>135 * 110 CB2:=0
>136 * 111 CB2:=1
>137 * 0 CBI Interrupt bij 1/0 op CBI
>138 * 1 CBI Interrupt bij 0/1 op CBI
>139 * Neg strobe, neg ACK or Busy
>140 * 76543210
719F: A9 E0 >141          LDA  #%11100000 ;zet de PCR juist
71A1: 99 8C C0 >142          STA  VIA_PCR,Y  ;CB2-hoog
>143 *
71A4: A9 00 >144 P_NOINIT LDA  #<B_PAR1 ;binnenkomst adres
71A6: 85 3A >145          STA  INDIRECL   ;in INDIRECT register
71A8: A9 B1 >146          LDA  #<PARRTS   ;sla terugkeer punt op
71AA: 8D 00 CF >147          STA  RETADR     ;
71AD: A9 1D >148          LDA  #ENKPAR1   ;laad volgende bank
71AF: D0 44 >149          ENE  SCHAKELL   ;en schakelen maar
>150 *
>151 *****
>152 *Hier komt de andere bank weer binnen
71B1: A9 00 >153 PARRTS LDA  #$0          ;Herstel de registers
71B3: 85 3A >154          STA  INDIRECL   ;herstel INDIRECT
71B5: AD 03 CF >155          LDA  SAVCREG    ;herstel CREG
71B8: A0 FF >156          LDY  #CREGL     ;CREG index
71BA: 91 3A >157          STA  (INDIRECT),Y ;en in CREG
71BC: BD 38 07 >158          LDA  P_MODE,X  ;moet karakter ook naar video
71BF: 0A >159          ASL              ;CARRY set is ja
71C0: 68 >160          P_ISINP PLA     ;YREG
71C1: A8 >161          TAY              ;
71C2: 68 >162          PLA              ;XREG
71C3: AA >163          TAX              ;
71C4: 68 >164          PLA              ;karakter
71C5: 90 03 >165          BCC  P_ERUIT    ;geen video echo

```

VIERLINGKAART BESTURINGSSOFTWARE

```

71C7: 4C F0 FD >166          JMP  COUT1      ;wel video echo
71CA: 60          >167  P_ERUIT  RTS
          >168  *
          >169  *vullen tot BANKSWITCH op zijn plaats komt
          >170  *
71CB: FF FF FF >171          HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
71DB: FF FF FF >172          HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
71EB: FF FF FF >173          HEX  FFFFFFFFFFFFFFFFFFFFFFFF
          >174  *
          >175  *****
          >176  *
          >177  *Schakel naar andere BANK      *
          >178  *
          >179  *****
          >180  *
          >181  *Schakel andere bank in door het
          >182  *gewenste banknummer naar adresgebied
          >183  *$Cn00-$CnFE te schrijven, banknummer
          >184  *staat reeds in ACCU
          >185  *In adres INDIRECT staat reeds waar naar
          >186  *toe gesprongen wordt bij binnenkomst
          >187  *andere bank
          >188  *
          >189  *Y-register wordt gebruikt
          >190  *
71F5: A0 00      >191  SCHAKEL1 LDY  #$0
71F7: 09 80      >192          ORA  #SOFTBANK ;Zet softbankselectie bit
71F9: 91 3A      >193          STA  (INDIRECT),Y ;Schakel om
71FB: 6C 3A 00   >194          JMP  (INDIRECT) ;Spring weg bij binnenkomst
          >195  *
71FE: 01          >196          DFB  ENKPAR      ;Nummer van deze BANK
71FF: FF          >197          DFB  $FF        ;Adres Controle Register
          >198  *

```

VIERLINGKAART BESTURINGSSOFTWARE

```

>1 *****
>2 *****
>3 ***
>4 ***   B A N K   N R   2   ***
>5 ***
>6 *****
>7 *****
>8 *
>9 * bank BNKSER VERSIE 5.0/850607 *
>10 *
>11 *****
>12 *
>13 *****
>14 *
>15 *   P R I N T E R D R I V E R   *
>16 *       S E R I E E L           *
>17 *
>18 *****
>19 *
>20 *
>21 *
7200: 2C 58 FF >22          BIT   IORTS      ;zet overflow=initiele entry
7203: 70 04   >23          BVS   S_ENTRY   ;altijd
>24 *
7205: 38     >25 S_IN   SEC           ;zet carry=input
7206: 90     >26 DFB   $90           ;=BCC
>27 *
7207: 18     >28 S_OUT  CLC           ;clear carry=output
7208: B8     >29 CLV           ;niet initieel clear overflow
>30 *
7209: 48     >31 S_ENTRY PHA          ;bewaar karakter
720A: 8A     >32 TXA          ;bewaar XREG
720B: 48     >33 PHA
720C: 98     >34 TYA          ;bewaar YREG
720D: 48     >35 PHA
720E: 08     >36 PHP           ;bewaar status
720F: 78     >37 SEI           ;disable interrupten even
7210: 20 58 FF >38 JSR   IORTS      ;bekende RTS positie
7213: BA     >39 TSX          ;nu staat slotadres op stack($Cn
7214: BC 00 01 >40 LDY   STACK,X    ;pak hoog slotadres $Cn
7217: 8C F8 07 >41 STY   MSLOT      ;bewaar in screenhole
721A: 84 3B   >42 STY   INDIRECT  ;en in INDIRECT register
721C: A9 00   >43 LDA   #$0       ;completeer INDIRECT register
721E: 85 3A   >44 STA   INDIRECTL
7220: 68     >45 PLA          ;ga karakter halen
7221: 68     >46 PLA
7222: 68     >47 PLA
7223: 68     >48 PLA
7224: 9A     >49 TXS          ;zet stackpointer terug
7225: 48     >50 PHA          ;bewaar karakter weer
7226: 98     >51 TYA          ;
7227: AA     >52 TAX          ;bewaar $Cn in XREG
>53 *zet de kaart en ACTIA aan
7228: A0 FF   >54 LDY   #CREGL
722A: B1 3A   >55 LDA   (INDIRECT),Y ;haal CREG op

```

VIERLINGKAART BESTURINGSSOFTWARE

```

722C: 48      >56      PHA          ;bewaar even, RAM eerst aan
722D: 8D FF CF >57      STA CLRROM   ;schakel extensie ROM-en uit
7230: A9 8F   >58      LDA #ACIA_ON ;ACIA, kaart en RAM aan
7232: 91 3A   >59      STA (INDIRECT),Y ;schakel in
                        >60      *
7234: A0 FE   >61      LDY #BANKNR
7236: B1 3A   >62      LDA (INDIRECT),Y;haal huidig banknummer
7238: 8D 02 CF >63      STA RETBANK  ;bewaar voor terugkeer
723B: 8E 01 CF >64      STX SLOTADR ;bewaar $Cn ook in RAM
723E: 68     >65      PLA         ;haal oude CREG terug
723F: 8D 03 CF >66      STA SAVCREG ;bewaar die
7242: 68     >67      PLA         ;ga status terughalen
7243: 28     >68      PLP        ;voor enable interrupt
7244: 48     >69      PHA        ;bewaar karakter toch op stack
7245: 50 45   >70      BVC S_NOINIT ;geen initiele entry
                        >71      *
                        >72      *initiele eerste maal binnenkomst
                        >73      *
7247: B8     >74      CLV        ;clear de overflow
7248: 18     >75      CLC        ;nodig voor INPUT bestrijding
                        >76      *in XREG $Cn, controleer op input, zo ja eruit
7249: E4 39   >77      CPX KSWH   ;is het input
724B: F0 4C   >78      BEQ S_ISINPH ;Ja, gelijk er weer uit
724D: E4 37   >79      CPX CSWH   ;is het wel PR#
724F: D0 04   >80      BNE S_VERDER
                        >81      *
7251: A9 07   >82      LDA #<S_OUT ;pas de output hook aan
7253: 85 36   >83      STA CSWL
                        >84      *
7255: A9 00   >85      S_VERDER LDA #$0
7257: 9D B8 04 >86      STA S_LINE,X ;#CR's sinds laatste nieuwe blad
725A: 9D B8 03 >87      STA S_IMARGE,X ;geen linker marge inspringen
725D: A9 40   >88      LDA #SERMODE ;B7=video,B6=auto LF,B5=FF gener
725F: 9D 38 07 >89      STA S_MODE,X ;video uit, auto LF aan en FF ui
7262: A9 00   >90      LDA #PAGLENNR ;aantal bedrukte regels per blad
7264: 9D 38 04 >91      STA S_PAGLEN,X
7267: A9 FF   >92      LDA #SERREGLN ;posities per regel
7269: 9D 38 06 >93      STA S_REGLN,X ;maximale regellengte
                        >94      *
726C: AD 01 C8 >95      ACIAVRLJ LDA ACIASTAT ;haal ACIA status
726F: 29 10   >96      AND #TDRE   ;transmit leeg?
7271: F0 F9   >97      BEQ ACIAVRLJ ;nee
7273: 8D 01 C8 >98      STA ACIASTAT ;reset de ACIA
7276: A9 9B   >99      LDA #<S_BAUD ;indirect naar BPS tabel
7278: 85 3A   >100     STA INDIRECL
727A: AD 04 C8 >101     LDA SREG    ;pak BPS switchen
727D: 29 07   >102     AND #%00000111 ;Masker SW4 (1-3) uit
727F: A8     >103     TAY        ;haal naar YREG
7280: B1 3A   >104     LDA (INDIRECT),Y
7282: 09 10   >105     ORA #INITCON8 ;gooi standaard bitten om
7284: 8D 03 C8 >106     STA ACIACONT
7287: A9 CB   >107     LDA #INITCOMM ;initialiseer commandoreg (no pa
7289: 8D 02 C8 >108     STA ACIACOMM
                        >109     *
728C: A9 00   >110     S_NOINIT LDA #<B_SER1 ;binnenkomst adres

```

VIERLINGKAART BESTURINGSSOFTWARE

```

728E: 85 3A >111 STA INDIRECL ;in INDIRECT register
7290: A9 A3 >112 LDA #<SERRTS ;sla het terugkeer adres op
7292: 8D 00 CF >113 STA RETADR
7295: A9 1C >114 LDA #BNKSER1 ;volgend banknr erin
7297: D0 5C >115 BNE SCHAKEL2 ;en schakelen maar
7299: 90 1C >116 S_ISINPH BCC S_ISINP ;tussenjumpje
>117 *
>118 *Tabel voor de BPS format van ACIACONTREG
>119 *
729B: 05 >120 S_BAUD DFB %00000101 ;150 BPS
729C: 06 >121 DFB %00000110 ;300 BPS
729D: 07 >122 DFB %00000111 ;600 BPS
729E: 08 >123 DFB %00001000 ;1200 bps
729F: 0A >124 DFB %00001010 ;2400 BPS
72A0: 0C >125 DFB %00001100 ;4800 BPS
72A1: 0E >126 DFB %00001110 ;9600 BPS
72A2: 0F >127 DFB %00001111 ;19200 BPS
>128 *
>129 *****
>130 *Hier komt de andere bank weer binnen
72A3: A9 00 >131 SERRIS LDA #$0 ;Herstel de registers
72A5: 85 3A >132 STA INDIRECL ;herstel INDIRECT
72A7: AD 03 CF >133 LDA SAVCREG ;herstel CREG
72AA: 29 F8 >134 AND #%11111000 ;ACIA uit
72AC: 8D 03 CF >135 STA SAVCREG ;bewaar
72AF: A0 FF >136 LDY #CREGL ;CREG index
72B1: 91 3A >137 STA (INDIRECT),Y;en in CREG
72B3: BD 38 07 >138 LDA S_MODE,X ;moet karakter ook naar video
72B6: 0A >139 ASL ;CARRY set is ja
72B7: 68 >140 S_ISINP PLA ;YREG
72B8: A8 >141 TAY
72B9: 68 >142 PLA ;XREG
72BA: AA >143 TAX
72BB: 68 >144 PLA ;karakter
72BC: 90 03 >145 BCC S_ERUIT ;geen video echo
72BE: 4C F0 FD >146 JMP COUT1 ;wel video echo
72C1: 60 >147 S_ERUIT RTS
>148 *
>149 *vullen tot BANKSWITCH op zijn plaats komt
>150 *
72C2: FF FF FF >151 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
72D2: FF FF FF >152 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
72E2: FF FF FF >153 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
72F2: FF FF FF >154 HEX FFFFFFFF
>155 *
>156 *****
>157 * *
>158 *Schakel naar andere BANK *
>159 * *
>160 *****
>161 *
>162 *Schakel andere bank in door het
>163 *gewenste banknummer naar adresgebied
>164 *$Cn00-$CnFE te schrijven, banknummer
>165 *staat reeds in ACCU

```

VIERLINGKAART BESTURINGSSOFTWARE

```

>166 *In adres INDIRECT staat reeds waar naar
>167 *toe gesprongen wordt bij binnenkomst
>168 *andere bank
>169 *
>170 *Y-register wordt gebruikt
>171 *
72F5: A0 00 >172 SCHAKEL2 LDY #$0
72F7: 09 80 >173          ORA #SOFTBANK ;Zet softbankselectie bit
72F9: 91 3A >174          STA (INDIRECT),Y ;Schakel om
72FB: 6C 3A 00 >175        JMP (INDIRECT) ;Spring weg bij binnenkomst
>176 *
72FE: 02 >177          DFB BNKSER ;Nummer van deze BANK
72FF: FF >178          DFB $FF ;Adres Controle Register
>179 *

```


VIERLINGKAART BESTURINGSSOFTWARE

```

>1 *****
>2 *****
>3 ***
>4 ***   B A N K   N R   3   ***
>5 ***
>6 *****
>7 *****
>8 *
>9 * bank BNKTERM VERSIE 5.0/850607*
>10 *
>11 *****
>12 *
>13 *****
>14 *
>15 *T E R M I N A L E M U L A T I E*
>16 *
>17 *****
>18 *
>19 *
>20 *
7300: 08 >21 PHP ;Bewaar status
7301: 78 >22 SEI ;Disable interrupten
7302: 2C FF CF >23 BIT CLRROM ;Disable extended ROM's
7305: 20 58 FC >24 JSR HOME ;Geeft $Cn op stack
7308: BA >25 TSX
7309: BD 00 01 >26 LDA STACK,X ;Pak $Cn
730C: 8D F8 07 >27 STA MSLOT ;Bewaar $Cn in tekstpage
730F: 85 3B >28 STA INDIRECH ;en voor indirect werk
7311: AA >29 TAX ;bewaar ook even
7312: 28 >30 PLP ;Interrupten weer mogelijk
>31 *Zet de ACIA aan
7313: A9 00 >32 LDA #$0 ;reset INDIRECT
7315: 85 3A >33 STA INDIRECL
7317: A0 FF >34 LDY #CREGL
7319: B1 3A >35 LDA (INDIRECT),Y ;Haal CREG
731B: 48 >36 PHA ;bewaar even
731C: 8D FF CF >37 STA CLRROM ;schakel extensie ROM-en uit
731F: A9 8F >38 LDA #ACIA_ON
7321: 91 3A >39 STA (INDIRECT),Y
7323: 68 >40 PLA
7324: 8D 03 CF >41 STA SAVCREG ;en bewaar inhoud
7327: 8A >42 TXA ;bewaar $Cn ook in RAM
7328: 8D 01 CF >43 STA SLOTADR
>44 *
>45 *we resetten de hooks naar IN#0
>46 *dan weten we tenminste waar we aan toe zijn
732B: 20 89 FE >47 JSR SETKBD ;reset IN#0
>48 *
732E: A9 7F >49 LDA #<SIGON ;Maak indirect gereed voor SIGON
7330: 85 3A >50 STA INDIRECL
7332: A0 00 >51 LDY #$0
7334: B1 3A >52 SIGMES LDA (INDIRECT),Y ;verzend sig_on en
7336: F0 09 >53 BEQ RDBPS ;BPS vraag
7338: 20 ED FD >54 JSR COUT
733B: C8 >55 INY

```

VIERLINGKAART BESTURINGSSOFTWARE

```

733C: D0 F6 >56 BNE SIGMES
>57 *
733E: 20 DD FB >58 RDBPSERR JSR BELL ;Foute invoer
7341: 20 35 FD >59 RDBPS JSR RDCHAR ;Lees waarde in
7344: C9 B1 >60 CMP #"1" ;controleer invoer
7346: 90 F6 >61 BLT RDBPSERR
7348: C9 B9 >62 CMP #"9"
734A: 10 F2 >63 BPL RDBPSERR
734C: 20 ED FD >64 JSR COUT ;Zet op scherm
734F: 29 0F >65 AND #$0F ;maak er decimaal van
7351: A8 >66 TAY ;bewaar waarde
7352: 88 >67 DEY ;nummer van 0-6
7353: A9 75 >68 LDA #<T_BPSTAB ;bereid INDIRECT voor op
7355: 85 3A >69 STA INDIRECL ;baudmasker tabel
7357: B1 3A >70 LDA (INDIRECT),Y;Zet juiste BPS bitten
7359: 48 >71 PHA ;bewaar op stack
>72 *
>73 *De rest gaat in de andere banken
>74 *Tref de nodige voorbereidingen voor
>75 *vertrek en terugkomst
735A: A9 00 >76 LDA #$0
735C: 85 3A >77 STA INDIRECL ;herstel INDIRECT
735E: A0 FE >78 LDY #BANKNR ;haal huidige banknr
7360: B1 3A >79 LDA (INDIRECT),Y
7362: 8D 02 CF >80 STA RETBANK
7365: A9 72 >81 LDA #<TERMRTS ;pak returnadres
7367: 8D 00 CF >82 STA RETADR ;en zet klaar
736A: A9 00 >83 LDA #<B_TERM1 ;Pak spronadresL
736C: 85 3A >84 STA INDIRECL ;zet in jumpadres
736E: A9 19 >85 LDA #BNKTERM1 ;Volgende bank
7370: D0 0B >86 BNE T_TUSJMP ;bankswitchen via tussen stap
>87 *
>88 *Opvullen zodat bankswitch op zijn plaats is
>89 *
>90 *Terugkeer punt geef return
7372: A9 8D >91 TERMRTS LDA #CR ;neem CR in ACCU bij RTS
7374: 60 >92 RTS ;Klaar eruit
>93 *
>94 *
>95 *
>96 * BPS tabel voor CONTROLEREGISTER ACIA
>97 *
7375: 15 >98 T_BPSTAB DFB %00010101 ;150 BPS
7376: 16 >99 DFB %00010110 ;300 BPS
7377: 17 >100 DFB %00010111 ;600 BPS
7378: 18 >101 DFB %00011000 ;1200 BPS
7379: 1A >102 DFB %00011010 ;2400 BPS
737A: 1C >103 DFB %00011100 ;4800 BPS
737B: 1E >104 DFB %00011110 ;9600 BPS
737C: 1F >105 DFB %00011111 ;19200BPS
>106 *
737D: D0 76 >107 T_TUSJMP BNE SCHAKEL3
>108 *
>109 *opvullen zodat bankswitch op zijn plaats is
>110 *

```

VIERLINGKAART BESTURINGSSOFTWARE

```

>111 *
>112 *
>113 *Boodschappen
737F: 8C 8D 8D >114 SIGON DFB CTRL,CR,CR,CR
7383: D6 C9 C5 >115 ASC "VIERLINGKAART TERMINAL EMULATIE V5.0"
73A7: 8D 8D >116 DFB CR,CR
73A9: C7 C5 C5 >117 ASC "GEEF LIJNSNELHEID: 1=150 2=300 3=600"
73CE: 8D >118 DFB CR
73CF: B4 BD B1 >119 ASC "4=1200 5=2400 6=4800 7=9600 8=19200 ?"
73F4: 00 >120 DFB $0
>121 *
>122 *
>123 *
>124 *****
>125 * *
>126 *Schakel naar andere BANK *
>127 * *
>128 *****
>129 *
>130 *Schakel andere bank in door het
>131 *gewenste banknummer naar adresgebied
>132 *$Cn00-$CnFE te schrijven, banknummer
>133 *staat reeds in ACCU
>134 *In adres INDIRECT staat reeds waar naar
>135 *toe gesprongen wordt bij binnenkomst
>136 *andere bank
>137 *
>138 *Y-register wordt gebruikt
>139 *
73F5: A0 00 >140 SCHAKEL3 LDY #$0
73F7: 09 80 >141 ORA #SOFTBANK ;Zet softbankselectie bit
73F9: 91 3A >142 STA (INDIRECT),Y ;Schakel om
73FB: 6C 3A 00 >143 JMP (INDIRECT) ;Spring weg bij binnenkomst
>144 *
73FE: 03 >145 DFB BNKIERM ;Nummer van deze BANK
73FF: FF >146 DFB $FF ;Adres Controle Register
>147 *

```

VIERLINGKAART BESTURINGSSOFTWARE

```

>1 *****
>2 *****
>3 ***                               ***
>4 ***   B A N K   N R   4   ***
>5 ***                               ***
>6 *****
>7 *****
>8 *                               *
>9 * bank BNKKLOK VERSIE 5.0/850607*
>10 *                               *
>11 *****
>12 *
>13 *****
>14 *                               *
>15 *   K L O K D R I V E R   *
>16 *                               *
>17 *****
>18 *
>19 *
>20 *Klokdriver voor de 4-lingkaart
>21 *ontworpen alleen voor het inlezen van
>22 *de tijd
>23 *
7400: 2C 58 FF >24 BIT IORIS ;zet overflow=initiele entry
7403: 70 05 >25 BVS K_ENTRY ;altijd
>26 *
7405: 38 >27 K_IN SEC ;zet carry=input
7406: B0 01 >28 BCS CLEARINI ;verder voor beide
>29 *
7408: 18 >30 K_OUT CLC ;clear carry=output
7409: B8 >31 CLEARINI CLV ;niet initieel clear overflow
>32 *
740A: 8D 78 07 >33 K_ENTRY STA SCRAT1 ;bewaar ACCU
740D: 8A >34 TXA ;bewaar XREG
740E: 48 >35 PHA
740F: 98 >36 TYA ;bewaar YREG
7410: 48 >37 PHA
7411: 08 >38 PHP ;bewaar status
7412: 78 >39 SEI ;disable interrupten even
7413: 20 58 FF >40 JSR IORIS ;bekende RIS positie
7416: BA >41 TSX ;nu staat slotadres op stack($Cn
7417: BD 00 01 >42 LDA STACK,X ;pak hoog slotadres $Cn
741A: 8D F8 07 >43 STA MSLOT ;bewaar in screenhole
741D: 85 3B >44 STA INDIRECH ;en in INDIRECT register
741F: AA >45 TAX ;bewaar in XREG
7420: A9 00 >46 LDA #$0 ;completeer INDIRECT register
7422: 85 3A >47 STA INDIRECL
>48 *zet de kaart en RAM aan
7424: A0 FF >49 LDY #CREGL
7426: B1 3A >50 LDA (INDIRECT),Y ;haal CREG op
7428: 48 >51 PHA ;bewaar even, RAM eerst aan
7429: 8D FF CF >52 STA CLRROM ;schakel extensie ROM-en uit
742C: A9 8E >53 LDA #RAMACTIV ;kaart en RAM aan
742E: 91 3A >54 STA (INDIRECT),Y ;schakel in
7430: 68 >55 PLA ;haal oude CREG terug

```

VIERLINGKAART BESTURINGSSOFTWARE

```

7431: 8D 03 CF >56          STA  SAVCREG      ;bewaar die
7434: 8E 01 CF >57          STX  SLOTADR     ;bewaar $Cn in RAM
7437: 28                   >58          PLP                    ;enable interrupt
7438: 50 43                   >59          BVC  K_NOINIH    ;geen initiele entry
>60          *
>61          *initiele eerste maal binnenkomst
>62          *
743A: B8                   >63          CLV                    ;clear de overflow
>64          *in XREG $Cn, controleer op output, zo ja reset
743B: E4 37                   >65          CPX  CSWH         ;is het output
743D: D0 06                   >66          BNE  K_ISINP     ;nee
743F: A9 08                   >67          LDA  #<K_OUT     ;laad output vector
7441: 85 36                   >68          STA  CSWL        ;zet hook
7443: D0 7F                   >69          BNE  K_EXITIH    ;altijd
>70          *
7445: E4 39                   >71          K_ISINP CPX  KSWH      ;IN#n ??
7447: D0 0A                   >72          BNE  K_PAKTYD    ;nee dus geen IO hook zetten
7449: A9 05                   >73          LDA  #<K_IN      ;pas input hook aan
744B: 85 38                   >74          STA  KSWL        ;
744D: A9 08                   >75          LDA  #<K_OUT     ;ook output ombuigen
744F: 85 36                   >76          STA  CSWL        ;ter bestrijding van echo
7451: 86 37                   >77          STX  CSWH
>78          *
7453: A9 C6                   >79          K_PAKTYD LDA  #<KLOKRIS ;laad terugkeer adres
7455: 8D 00 CF >80          STA  RETADR      ;zet klaar
7458: A0 FE                   >81          LDY  #BANKNR     ;haal huidig banknr op
745A: B1 3A                   >82          LDA  (INDIRECT),Y ;banknummer
745C: 8D 02 CF >83          STA  RETBANK     ;bewaar
745F: A9 00                   >84          LDA  #<B_KLOK1  ;sprongadres in bank BNKKLOK1
7461: 85 3A                   >85          STA  INDIRECL
7463: A9 1E                   >86          LDA  #BNKKLOK1  ;volgende bank erin
7465: D0 5B                   >87          BNE  TUSKLOK    ;altijd
>88          *
>89          *opvullen tot bankswitch op zijn plaats
>90          *
7467: FF FF FF >91          HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7477: FF FF FF >92          HEX  FFFFFFFFFF
>93          *
747D: 50 48                   >94          K_NOINIH BVC  K_NOINIT ;tussenjumpje
>95          *
747F: 4A 76 44 >96          K_STAMP HEX  4A7644550607
>97          *
>98          *
>99          *****
>100         *PRODOS ENTRY PUNT
>101         *
7485: 8D 78 07 >102         K_PRODOS STA  SCRAT1    ;nep bewaar aktie
7488: 8A                   >103         TXA                    ;bewaar registers
7489: 48                   >104         PHA
748A: 98                   >105         TYA
748B: 48                   >106         PHA
748C: 08                   >107         PHP
748D: 78                   >108         SEI                    ;disable even interrupten
>109         *Bij PRODOS niet de monitor gebruiken
>110         *werkt wat makkelijker in de PRODOS driver

```

VIERLINGKAART BESTURINGSSOFTWARE

```

>111 *voor IORTS wordt daarom een eigen RTS gebruikt
748E: A9 60 >112 LDA #$60 ;RTS opcode
7490: 85 3A >113 STA INDIRECL ;zet op adres 0
7492: 20 3A 00 >114 JSR INDIRECL
7495: BA >115 TSX ;stackpointer naar XREG
7496: BD 00 01 >116 LDA STACK,X ;hier is $Cn
7499: 85 3B >117 STA INDIRECH ;in indirect
749B: A9 00 >118 LDA #$0
749D: 85 3A >119 STA INDIRECL
749F: 28 >120 PLP ;interrupten mogen weer
74A0: A0 FF >121 LDY #CREGL ;CREG bewaren
74A2: B1 3A >122 LDA (INDIRECT),Y ;haal CREG op
74A4: 48 >123 PHA ;bewaar even, RAM eerst aan
74A5: 8D FF CF >124 STA CLRROM ;schakel extensie ROM-en uit
74A8: A9 8E >125 LDA #RAMACTIV ;kaart en RAM aan
74AA: 91 3A >126 STA (INDIRECT),Y ;schakel in
>127 *
74AC: 68 >128 PLA ;haal oude CREG terug
74AD: 8D 03 CF >129 STA SAVCREG
>130 *ga naar bank BNKKLOKP om verder te werken
74B0: A0 FE >131 LDY #BANKNR
74B2: B1 3A >132 LDA (INDIRECT),Y;haal huidig banknummer
74B4: 8D 02 CF >133 STA RETBANK ;bewaar voor terugkeer
74B7: A9 DE >134 LDA #<K_EXIT1 ;terugkeer punt
74B9: 8D 00 CF >135 STA RETADR
74BC: A9 00 >136 LDA #<B_KLOKP ;start adres in nieuwe bank
74BE: 85 3A >137 STA INDIRECL
74C0: A9 1F >138 LDA #BNKKLOKP ;volgendebanknr erin
74C2: D0 31 >139 TUSKLOK BNE SCHAKEL4 ;altijd
>140 *
>141 *
74C4: D0 18 >142 K_EXIT1H BNE K_EXIT1 ;altijd
>143 *
>144 *****
>145 *Terugkeerpunt nadat klokdata is ingelezen+geconverteerd
74C6: 38 >146 KLOKRTS SEC ;zorg zeker voor INPUT
>147 *****
>148 *
>149 *input binnenkomst NIET eerste maal
74C7: 90 15 >150 K_NOINIT BCC K_EXIT1 ;output dus gelijk eruit
74C9: AE 05 CF >151 LDX TDATATEL
74CC: EE 05 CF >152 INC TDATATEL ;verhoog pointer vast
74CF: BD 10 CF >153 LDA KLOKDATA,X ;pak karakter
74D2: C9 8D >154 CMP #CR ;CR?
74D4: D0 05 >155 BNE K_EXIT ;nee
>156 *
74D6: 20 89 FE >157 JSR SETKBD ;reset IN#0
74D9: A9 8D >158 LDA #CR ;karakter weer in ACCU
>159 *
74DB: 8D 78 07 >160 K_EXIT STA SCRAT1 ;bewaar uittevoeren kar even
74DE: A0 FF >161 K_EXIT1 LDY #CREGL ;herstel CREG in oude staat
74E0: A9 00 >162 LDA #$0
74E2: 85 3A >163 STA INDIRECL ;reset INDIRECT
74E4: AD 03 CF >164 LDA SAVCREG
74E7: 91 3A >165 STA (INDIRECT),Y ;CREG hersteld

```

VIERLINGKAART BESTURINGSSOFTWARE

```

74E9: 68      >166      PLA
74EA: A8      >167      TAY          ;restore YREG
74EB: 68      >168      PLA
74EC: AA      >169      TAX          ;restore XREG
74ED: AD 78 07 >170      LDA SCRAT1   ;pak karakter weer
74F0: 60      >171      RTS
              >172      *
              >173      *entrypunt voor PROdos
              >174      *
74F1: B0 92    >175      BCS K_PRODOS
74F3: 90 90    >176      BCC K_PRODOS
              >177      *
              >178      *
              >179      *****
              >180      *
              >181      *Schakel naar andere BANK
              >182      *
              >183      *****
              >184      *
              >185      *Schakel andere bank in door het
              >186      *gewenste banknummer naar adresgebied
              >187      *$Cn00-$CnFE te schrijven, banknummer
              >188      *stat reeds in ACCU
              >189      *In adres INDIRECT staat reeds waar naar
              >190      *toe gesprongen wordt bij binnenkomst
              >191      *andere bank
              >192      *
              >193      *Y-register wordt gebruikt
              >194      *
74F5: A0 00    >195      SCHAKEL4 LDY #0
74F7: 09 80    >196      ORA #SOFTBANK ;Zet softbankselectie bit
74F9: 91 3A    >197      STA (INDIRECT),Y ;Schakel om
74FB: 6C 3A 00 >198      JMP (INDIRECT) ;Spring weg bij binnenkomst
              >199      *
74FE: 04      >200      DFB BNKKLOKR ;Nummer van deze BANK
74FF: FF      >201      DFB $FF      ;Adres Controle Register
              >202      *

```

VIERLINGKAART BESTURINGSSOFTWARE

```

>1 *****
>2 *****
>3 ***                               ***
>4 ***   B A N K   N R   5   ***
>5 ***                               ***
>6 *****
>7 *****
>8 *                               *
>9 *bank BNKKLOKS VERSIE 5.0/850607*
>10 *                               *
>11 *****
>12 *
>13 *****
>14 *                               *
>15 *   K L O K   Z E T T E N   *
>16 *                               *
>17 *****
>18 *
>19 *
>20 *klokgelijkzetten op interactieve wijze
>21 *
>22 *
7500: 08      >23      PHP                ;bewaar status
7501: 78      >24      SEI                ;disable interrupts
7502: 20 58 FC >25      JSR HOME          ;$Cn staat nu op stack
7505: BA      >26      TSX                ;stack pointer in XREG
7506: BD 00 01 >27      LDA STACK,X      ;pak $Cn
7509: 8D F8 07 >28      STA MSLOT        ;bewaar $Cn in screenhole
750C: 85 3B   >29      STA INDIRECT     ;ook in INDIRECT
750E: AA      >30      TAX                ;bewaar ook in XREG
750F: A9 00   >31      LDA #$0          ;zet INDIRECThapadres
7511: 85 3A   >32      STA INDIRECTL    ;rest van INDIRECT
7513: 28      >33      PLP                ;interrupts weer mogelijk
7514: A0 FF   >34      LDY #CREGL        ;
7516: B1 3A   >35      LDA (INDIRECT),Y ;haal CREG
7518: 48      >36      PHA                ;bewaar
7519: A9 8E   >37      LDA #RAMACTIV    ;zet kaart en RAM aan
751B: 8D FF CF >38      STA CLRROM      ;schakel extensie ROM-en uit
751E: 91 3A   >39      STA (INDIRECT),Y
7520: 68      >40      PLA                ;haal CREG terug
7521: 8D 03 CF >41      STA SAVCREG      ;bewaar
7524: 8E 01 CF >42      STX SLOTADR
7527: A0 FE   >43      LDY #BANKNR      ;haal banknummer
7529: B1 3A   >44      LDA (INDIRECT),Y
752B: 8D 02 CF >45      STA RETBANK     ;bewaar
>46 *reset de eventuele IN#n
752E: 20 89 FE >47      JSR SETKBD
>48 *
7531: A9 B6   >49      LDA #<Z_SIGON ;haal boodschap adres
7533: 85 3A   >50      STA INDIRECTL
7535: A0 00   >51      LDY #$0
7537: B1 3A   >52      Z_MEERMS LDA (INDIRECT),Y ;PAK BOODSCHAP
7539: F0 06   >53      BEQ Z_LSKLOK
753B: 20 ED FD >54      JSR COUT        ;schrijf
753E: C8      >55      INY

```


VIERLINGKAART BESTURINGSSOFTWARE

```

753F: D0 F6 >56 BNE Z_MEERMS ;altijd
7541: A9 4E >57 Z_LSKLOK LDA #<Z_EDIT ;terugkeer punt
7543: A2 00 >58 LDX #<B_KLOK1 ;adres in andere bank
7545: 8D 00 CF >59 Z_SWITCH STA RETADR
7548: 86 3A >60 STX INDIRECL ;store adres
754A: A9 1E >61 LDA #ENKKLOK1 ;klok bank
754C: D0 66 >62 BNE TUSKLOKS
>63 *
>64 *****
>65 *Hier komt de ingelezen klok weer terug
754E: A2 00 >66 Z_EDIT LDX #$0 ;reset INDIRECT
7550: 86 3A >67 STX INDIRECL
7552: A0 FF >68 LDY #CREGL ;neem CREG in YREG
7554: A9 8E >69 Z_REGEL LDA #RAMACTIV
7556: 8D FF CF >70 STA CLRROM ;ivm 80 kol schakel ext. uit
7559: 91 3A >71 STA (INDIRECT),Y ;vierling weer aan
755B: BD 10 CF >72 LDA KLOKDATA,X
755E: C9 8D >73 CMP #CR ;einde bereikt
7560: F0 06 >74 BEQ Z_WYZIG
7562: 20 ED FD >75 JSR COUT ;zet op scherm
7565: E8 >76 INX
7566: D0 EC >77 BNE Z_REGEL ;altijd
>78 *
7568: A2 00 >79 Z_WYZIG LDX #$0
756A: 86 24 >80 Z_WYZIGM STX CH ;reset CH
756C: 20 0C FD >81 JSR RDKEY ;lees karakter
756F: C9 8D >82 CMP #CR ;<cr>?
7571: F0 14 >83 BEQ Z_CARDON ;ja
7573: C9 88 >84 CMP #BS ;BS?
7575: D0 05 >85 BNE Z_CLJFER ;nee
7577: CA >86 DEX ;BS dus positie terug
7578: 30 1F >87 BMI Z_INC ;voorbij linker marge
757A: 10 1E >88 BPL Z_VGL ;gedaan doorgaan
757C: C9 B0 >89 Z_CLJFER CMP #"0" ;controleer op cijfer
757E: 30 19 >90 BMI Z_INC
7580: C9 BA >91 CMP #"9"+1 ;< OF = negen
7582: 10 15 >92 BPL Z_INC ;nee
7584: 20 ED FD >93 JSR COUT ;schrijf uit
7587: 48 >94 Z_CARDON PHA ;bewaar character even
7588: A0 FF >95 LDY #CREGL ;verzielt door RDCHAR
758A: A9 8E >96 LDA #RAMACTIV
758C: 8D FF CF >97 STA CLRROM ;ivm 80kol schakel ext.rom uit
758F: 91 3A >98 STA (INDIRECT),Y ;zet 4ling weer aan
7591: 68 >99 PLA ;haal character weer terug
7592: C9 8D >100 CMP #CR
7594: F0 08 >101 BEQ Z_SCHRYF
7596: 9D 10 CF >102 STA KLOKDATA,X ;zet in klok RAM
7599: E8 >103 Z_INC INX
759A: E0 11 >104 Z_VGL CPX #KLOKLANG ;einde string
759C: D0 CC >105 BNE Z_WYZIGM ;nee
>106 *
759E: A9 A4 >107 Z_SCHRYF LDA #<Z_EXIT ;return adres
75A0: A2 A5 >108 LDX #<K_SCHRYF ;entry adres(<0)
75A2: D0 A1 >109 BNE Z_SWITCH ;altijd
>110 *

```

VIERLINGKAART BESTURINGSSOFTWARE

```

>111 *****
>112 *binnenkomstpunt na het weg schrijven
>113 *van de gewijzigde klokdata
75A4: A9 00 >114 Z_EXIT LDA #0 ;reset INDIRECT
75A6: 85 3A >115 STA INDIRECL
75A8: A0 FF >116 LDY #CREGL ;herstel CREG
75AA: AD 03 CF >117 LDA SAVCREG
75AD: 91 3A >118 STA (INDIRECT),Y
75AF: A9 8D >119 LDA #CR ;zet een CR in ACCU
75B1: A2 00 >120 LDX #0 ;zet XREG op 0
75B3: 60 >121 RTS ;terug naar aanroeper
>122 *
75B4: D0 3F >123 TUSKLOKS BNE SCHAKEL5 ;tussenjumpje
>124 *
>125 *boodschap en heading van de klokuitlezing
>126 *
75B6: 8C 8D 8D >127 Z_SIGON DFB CTRL,CR,CR,CR
75BA: D6 C9 C5 >128 ASC "VIERLINGKAART KLOK ZETTEN V5.0"
75D8: 8D 8D 8D >129 DFB CR,CR,CR
75DB: D5 D5 BA >130 ASC "UU:MM:SS JJMMDD W"
75EC: 8D 00 >131 DFB CR,$0
>132 *
>133 *opvullen tot bankswitch op plaats
>134 *
75EE: FF FF FF >135 HEX FFFFFFFFFFFFFFFF
>136 *
>137 *****
>138 * *
>139 *Schakel naar andere BANK *
>140 * *
>141 *****
>142 *
>143 *Schakel andere bank in door het
>144 *gewenste banknummer naar adresgebied
>145 *$Cn00-$CnFE te schrijven, banknummer
>146 *staat reeds in ACCU
>147 *In adres INDIRECT staat reeds waar naar
>148 *toe gesprongen wordt bij binnenkomst
>149 *andere bank
>150 *
>151 *Y-register wordt gebruikt
>152 *
75F5: A0 00 >153 SCHAKEL5 LDY #0
75F7: 09 80 >154 ORA #SOFTBANK ;Zet softbankselectie bit
75F9: 91 3A >155 STA (INDIRECT),Y ;Schakel om
75FB: 6C 3A 00 >156 JMP (INDIRECT) ;Spring weg bij binnenkomst
>157 *
75FE: 05 >158 DFB ENKKLOKS ;Nummer van deze BANK
75FF: FF >159 DFB $FF ;Adres Controle Register
>160 *

```

VIERLINGKAART BESTURINGSSOFTWARE

```

>1 *****
>2 *****
>3 *** ***
>4 *** BANK NR 6 ***
>5 *** ***
>6 *****
>7 *****
>8 * *
>9 *bank BNKRES6 VERSIE 5.0/850607*
>10 * *
>11 *****
>12 *
>13 *
7600: FF FF FF >14 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7610: FF FF FF >15 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7620: FF FF FF >16 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7630: FF FF FF >17 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7640: FF FF FF >18 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7650: FF FF FF >19 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7660: FF FF FF >20 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7670: FF FF FF >21 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7680: FF FF FF >22 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7690: FF FF FF >23 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
76A0: FF FF FF >24 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
76B0: FF FF FF >25 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
76C0: FF FF FF >26 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
76D0: FF FF FF >27 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
76E0: FF FF FF >28 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
76F0: FF FF FF >29 HEX FFFFFFFF
>30 *
>31 *****
>32 * *
>33 *Schakel naar andere BANK *
>34 * *
>35 *****
>36 *
>37 *Schakel andere bank in door het
>38 *gewenste banknummer naar adresgebied
>39 *$Cn00-$CnFE te schrijven, banknummer
>40 *staat reeds in ACCU
>41 *In adres INDIRECT staat reeds waar naar
>42 *toe gesprongen wordt bij binnenkomst
>43 *andere bank
>44 *
>45 *Y-register wordt gebruikt
>46 *
76F5: A0 00 >47 SCHAKEL6 LDY #$0
76F7: 09 80 >48 ORA #SOFTBANK ;Zet softbankselectie bit
76F9: 91 3A >49 STA (INDIRECT),Y ;Schakel om
76FB: 6C 3A 00 >50 JMP (INDIRECT) ;Spring weg bij binnenkomst
>51 *
76FE: 06 >52 DFB BNKRES6 ;Nummer van deze BANK
76FF: FF >53 DFB $FF ;Adres Controle Register
>54 *

```

VIERLINGKAART BESTURINGSSOFTWARE

```

>1 *****
>2 *****
>3 *** ***
>4 *** BANK NR 7 ***
>5 *** ***
>6 *****
>7 *****
>8 * *
>9 *bank BNKRES7 VERSIE 5.0/850607*
>10 * *
>11 *****
>12 *
>13 *
7700: FF FF FF >14 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7710: FF FF FF >15 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7720: FF FF FF >16 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7730: FF FF FF >17 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7740: FF FF FF >18 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7750: FF FF FF >19 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7760: FF FF FF >20 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7770: FF FF FF >21 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7780: FF FF FF >22 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7790: FF FF FF >23 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
77A0: FF FF FF >24 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
77B0: FF FF FF >25 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
77C0: FF FF FF >26 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
77D0: FF FF FF >27 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
77E0: FF FF FF >28 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
77F0: FF FF FF >29 HEX FFFFFFFF
>30 *
>31 *****
>32 * *
>33 *Schakel naar andere BANK *
>34 * *
>35 *****
>36 *
>37 *Schakel andere bank in door het
>38 *gewenste banknummer naar adresgebied
>39 *$Cn00-$CnFE te schrijven, banknummer
>40 *staat reeds in ACCU
>41 *In adres INDIRECT staat reeds waar naar
>42 *toe gesprongen wordt bij binnenkomst
>43 *andere bank
>44 *
>45 *Y-register wordt gebruikt
>46 *
77F5: A0 00 >47 SCHAKEL7 LDY #$0
77F7: 09 80 >48 ORA #SOFTBANK ;Zet softbankselectie bit
77F9: 91 3A >49 STA (INDIRECT),Y ;Schakel om
77FB: 6C 3A 00 >50 JMP (INDIRECT) ;Spring weg bij binnenkomst
>51 *
77FE: 07 >52 DFB BNKRES7 ;Nummer van deze BANK
77FF: FF >53 DFB $FF ;Adres Controle Register
>54 *

```

VIERLINGKAART BESTURINGSSOFTWARE

```

>1 *****
>2 *****
>3 ***                               ***
>4 ***   B A N K   N R   8   ***
>5 ***                               ***
>6 *****
>7 *****
>8 *                                 *
>9 *bank BNKRES8  VERSIE 5.0/850607*
>10 *                                 *
>11 *****
>12 *
>13 *
7800: FF FF FF >14   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7810: FF FF FF >15   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7820: FF FF FF >16   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7830: FF FF FF >17   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7840: FF FF FF >18   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7850: FF FF FF >19   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7860: FF FF FF >20   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7870: FF FF FF >21   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7880: FF FF FF >22   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7890: FF FF FF >23   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
78A0: FF FF FF >24   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
78B0: FF FF FF >25   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
78C0: FF FF FF >26   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
78D0: FF FF FF >27   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
78E0: FF FF FF >28   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
78F0: FF FF FF >29   HEX  FFFFFFFFFF
>30 *
>31 *****
>32 *                                 *
>33 *Schakel naar andere BANK      *
>34 *                                 *
>35 *****
>36 *
>37 *Schakel andere bank in door het
>38 *gewenste banknummer naar adresgebied
>39 *$Cn00-$CnFE te schrijven, banknummer
>40 *staat reeds in ACCU
>41 *In adres INDIRECT staat reeds waar naar
>42 *toe gesprongen wordt bij binnenkomst
>43 *andere bank
>44 *
>45 *Y-register wordt gebruikt
>46 *
78F5: A0 00 >47   SCHAKELS LDY  #$0
78F7: 09 80 >48   ORA  #SOFTBANK ;Zet softbankselectie bit
78F9: 91 3A >49   STA  (INDIRECT),Y ;Schakel om
78FB: 6C 3A 00 >50   JMP  (INDIRECT) ;Spring weg bij binnenkomst
>51 *
78FE: 08 >52   DFB  ENKRES8 ;Nummer van deze BANK
78FF: FF >53   DFB  $FF ;Adres Controle Register
>54 *

```

VIERLINGKAART BESTURINGSSOFTWARE

```

>1 *****
>2 *****
>3 *** ***
>4 *** BANK NR 9 ***
>5 *** ***
>6 *****
>7 *****
>8 * *
>9 *bank ENKRES9 VERSIE 5.0/850607*
>10 * *
>11 *****
>12 *
>13 *
7900: FF FF FF >14 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7910: FF FF FF >15 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7920: FF FF FF >16 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7930: FF FF FF >17 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7940: FF FF FF >18 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7950: FF FF FF >19 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7960: FF FF FF >20 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7970: FF FF FF >21 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7980: FF FF FF >22 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7990: FF FF FF >23 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
79A0: FF FF FF >24 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
79B0: FF FF FF >25 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
79C0: FF FF FF >26 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
79D0: FF FF FF >27 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
79E0: FF FF FF >28 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
79F0: FF FF FF >29 HEX FFFFFFFF
>30 *
>31 *****
>32 * *
>33 *Schakel naar andere BANK *
>34 * *
>35 *****
>36 *
>37 *Schakel andere bank in door het
>38 *gewenste banknummer naar adresgebied
>39 *$Cn00-$CnFE te schrijven, banknummer
>40 *staat reeds in ACCU
>41 *In adres INDIRECT staat reeds waar naar
>42 *toe gesprongen wordt bij binnenkomst
>43 *andere bank
>44 *
>45 *Y-register wordt gebruikt
>46 *
79F5: A0 00 >47 SCHAKEL9 LDY #$0
79F7: 09 80 >48 ORA #SOFTBANK ;Zet softbankselectie bit
79F9: 91 3A >49 STA (INDIRECT),Y ;Schakel om
79FB: 6C 3A 00 >50 JMP (INDIRECT) ;Spring weg bij binnenkomst
>51 *
79FE: 09 >52 DFB ENKRES9 ;Nummer van deze BANK
79FF: FF >53 DFB $FF ;Adres Controle Register
>54 *

```

VIERLINGKAART BESTURINGSSOFTWARE

```

>1 *****
>2 *****
>3 ***                               ***
>4 ***   B A N K   N R   A   ***
>5 ***                               ***
>6 *****
>7 *****
>8 *                                 *
>9 *bank BNKRESA  VERSIE 5.0/850607*
>10 *                                 *
>11 *****
>12 *
>13 *
7A00: FF FF FF >14   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7A10: FF FF FF >15   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7A20: FF FF FF >16   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7A30: FF FF FF >17   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7A40: FF FF FF >18   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7A50: FF FF FF >19   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7A60: FF FF FF >20   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7A70: FF FF FF >21   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7A80: FF FF FF >22   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7A90: FF FF FF >23   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7AA0: FF FF FF >24   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7AB0: FF FF FF >25   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7AC0: FF FF FF >26   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7AD0: FF FF FF >27   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7AE0: FF FF FF >28   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7AF0: FF FF FF >29   HEX  FFFFFFFFFF
>30 *
>31 *****
>32 *                                 *
>33 *Schakel naar andere BANK      *
>34 *                                 *
>35 *****
>36 *
>37 *Schakel andere bank in door het
>38 *gewenste banknummer naar adresgebied
>39 *$Cn00-$CnFE te schrijven, banknummer
>40 *staat reeds in ACCU
>41 *In adres INDIRECT staat reeds waar naar
>42 *toe gesprongen wordt bij binnenkomst
>43 *andere bank
>44 *
>45 *Y-register wordt gebruikt
>46 *
7AF5: A0 00 >47   SCHAKELA LDY  #$0
7AF7: 09 80 >48           ORA  #SOFTRANK ;Zet softbankselectie bit
7AF9: 91 3A >49           STA  (INDIRECT),Y ;Schakel om
7AFB: 6C 3A 00 >50          JMP  (INDIRECT) ;Spring weg bij binnenkomst
>51 *
7AFE: 0A >52           DFB  BNKRESA ;Nummer van deze BANK
7AFF: FF >53           DFB  $FF ;Adres Controle Register
>54 *

```

VIERLINGKAART BESTURINGSSOFTWARE

```

>1 *****
>2 *****
>3 ***                               ***
>4 ***   B A N K   N R   B   ***
>5 ***                               ***
>6 *****
>7 *****
>8 *                                 *
>9 *bank BNKRESB VERSIE 5.0/850607*
>10 *                                 *
>11 *****
>12 *
>13 *
7B00: FF FF FF >14   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7B10: FF FF FF >15   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7B20: FF FF FF >16   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7B30: FF FF FF >17   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7B40: FF FF FF >18   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7B50: FF FF FF >19   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7B60: FF FF FF >20   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7B70: FF FF FF >21   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7B80: FF FF FF >22   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7B90: FF FF FF >23   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7BA0: FF FF FF >24   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7BB0: FF FF FF >25   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7BC0: FF FF FF >26   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7BD0: FF FF FF >27   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7BE0: FF FF FF >28   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7BF0: FF FF FF >29   HEX  FFFFFFFFFF
>30 *
>31 *****
>32 *                                 *
>33 *Schakel naar andere BANK       *
>34 *                                 *
>35 *****
>36 *
>37 *Schakel andere bank in door het
>38 *gewenste banknummer naar adresgebied
>39 *$Cn00-$CnFE te schrijven, banknummer
>40 *staat reeds in ACCU
>41 *In adres INDIRECT staat reeds waar naar
>42 *toe gesprongen wordt bij binnenkomst
>43 *andere bank
>44 *
>45 *Y-register wordt gebruikt
>46 *
7BF5: A0 00 >47   SCHAKELB LDY  #$0
7BF7: 09 80 >48   ORA  #SOFTBANK ;Zet softbankselectie bit
7BF9: 91 3A >49   STA  (INDIRECT),Y ;Schakel om
7BFB: 6C 3A 00 >50   JMP  (INDIRECT) ;Spring weg bij binnenkomst
>51 *
7BFE: 0B >52   DFB  BNKRESB ;Nummer van deze BANK
7BFF: FF >53   DFB  $FF ;Adres Controle Register
>54 *

```


VIERLINGKAART BESTURINGSSOFTWARE

```

>1 *****
>2 *****
>3 ***                               ***
>4 ***   B A N K   N R   C   ***
>5 ***                               ***
>6 *****
>7 *****
>8 *                               *
>9 *bank BNKRESC VERSIE 5.0/850607*
>10 *                               *
>11 *****
>12 *
>13 *
7C00: FF FF FF >14   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7C10: FF FF FF >15   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7C20: FF FF FF >16   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7C30: FF FF FF >17   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7C40: FF FF FF >18   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7C50: FF FF FF >19   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7C60: FF FF FF >20   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7C70: FF FF FF >21   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7C80: FF FF FF >22   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7C90: FF FF FF >23   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7CA0: FF FF FF >24   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7CB0: FF FF FF >25   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7CC0: FF FF FF >26   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7CD0: FF FF FF >27   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7CE0: FF FF FF >28   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7CF0: FF FF FF >29   HEX  FFFFFFFF
>30 *
>31 *****
>32 *                               *
>33 *Schakel naar andere BANK   *
>34 *                               *
>35 *****
>36 *
>37 *Schakel andere bank in door het
>38 *gewenste banknummer naar adresgebied
>39 *$Cn00-$CnFE te schrijven, banknummer
>40 *staat reeds in ACCU
>41 *In adres INDIRECT staat reeds waar naar
>42 *toe gesprongen wordt bij binnenkomst
>43 *andere bank
>44 *
>45 *Y-register wordt gebruikt
>46 *
7CF5: A0 00 >47   SCHAKELC LDY  #$0
7CF7: 09 80 >48           ORA  #SOFTBANK ;Zet softbankselectie bit
7CF9: 91 3A >49           STA  (INDIRECT),Y ;Schakel om
7CFB: 6C 3A 00 >50          JMP  (INDIRECT) ;Spring weg bij binnenkomst
>51 *
7CFE: 0C >52           DFB  BNKRESC ;Nummer van deze BANK
7CFF: FF >53           DFB  $FF ;Adres Controle Register
>54 *

```

VIERLINGKAART BESTURINGSSOFTWARE

```

>1 *****
>2 *****
>3 *** ***
>4 *** BANK NR D ***
>5 *** ***
>6 *****
>7 *****
>8 * *
>9 *bank BNKRESD VERSIE 5.0/850607*
>10 * *
>11 *****
>12 *
>13 *
7D00: FF FF FF >14 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7D10: FF FF FF >15 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7D20: FF FF FF >16 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7D30: FF FF FF >17 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7D40: FF FF FF >18 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7D50: FF FF FF >19 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7D60: FF FF FF >20 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7D70: FF FF FF >21 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7D80: FF FF FF >22 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7D90: FF FF FF >23 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7DA0: FF FF FF >24 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7DB0: FF FF FF >25 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7DC0: FF FF FF >26 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7DD0: FF FF FF >27 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7DE0: FF FF FF >28 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7DF0: FF FF FF >29 HEX FFFFFFFF
>30 *
>31 *****
>32 * *
>33 *Schakel naar andere BANK *
>34 * *
>35 *****
>36 *
>37 *Schakel andere bank in door het
>38 *gewenste banknummer naar adresgebied
>39 *$Cn00-$CnFE te schrijven, banknummer
>40 *staat reeds in ACCU
>41 *In adres INDIRECT staat reeds waar naar
>42 *toe gesprongen wordt bij binnenkomst
>43 *andere bank
>44 *
>45 *Y-register wordt gebruikt
>46 *
7DF5: A0 00 >47 SCHAKELD LDY #$0
7DF7: 09 80 >48 ORA #SOFTBANK ;Zet softbankselectie bit
7DF9: 91 3A >49 STA (INDIRECT),Y ;Schakel om
7DFB: 6C 3A 00 >50 JMP (INDIRECT) ;Spring weg bij binnenkomst
>51 *
7DFE: 0D >52 DFB BNKRESD ;Nummer van deze BANK
7DFE: FF >53 DFB $FF ;Adres Controle Register
>54 *

```

VIERLINGKAART BESTURINGSSOFTWARE

```

>1 *****
>2 *****
>3 *** ***
>4 *** BANK NRE ***
>5 *** ***
>6 *****
>7 *****
>8 * *
>9 *bank BNKRESE VERSIE 5.0/850607*
>10 * *
>11 *****
>12 *
>13 *
7E00: FF FF FF >14 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7E10: FF FF FF >15 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7E20: FF FF FF >16 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7E30: FF FF FF >17 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7E40: FF FF FF >18 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7E50: FF FF FF >19 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7E60: FF FF FF >20 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7E70: FF FF FF >21 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7E80: FF FF FF >22 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7E90: FF FF FF >23 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7EA0: FF FF FF >24 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7EB0: FF FF FF >25 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7EC0: FF FF FF >26 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7ED0: FF FF FF >27 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7EE0: FF FF FF >28 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7EF0: FF FF FF >29 HEX FFFFFFFF
>30 *
>31 *****
>32 * *
>33 *Schakel naar andere BANK *
>34 * *
>35 *****
>36 *
>37 *Schakel andere bank in door het
>38 *gewenste banknummer naar adresgebied
>39 *$Cn00-$CnFE te schrijven, banknummer
>40 *staat reeds in ACCU
>41 *In adres INDIRECT staat reeds waar naar
>42 *toe gesprongen wordt bij binnenkomst
>43 *andere bank
>44 *
>45 *Y-register wordt gebruikt
>46 *
7EF5: A0 00 >47 SCHAKELE LDY #$0
7EF7: 09 80 >48 ORA #SOFTBANK ;Zet softbankselectie bit
7EF9: 91 3A >49 STA (INDIRECT),Y ;Schakel om
7EFB: 6C 3A 00 >50 JMP (INDIRECT) ;Spring weg bij binnenkomst
>51 *
7EFE: 0E >52 DFB BNKRESE ;Nummer van deze BANK
7EFF: FF >53 DFB $FF ;Adres Controle Register
>54 *

```

VIERLINGKAART BESTURINGSSOFTWARE

```

>1 *****
>2 *****
>3 ***                               ***
>4 ***   B A N K   N R   F   ***
>5 ***                               ***
>6 *****
>7 *****
>8 *                               *
>9 *bank BNKRESF VERSIE 5.0/850607*
>10 *                               *
>11 *****
>12 *
>13 *
7F00: FF FF FF >14   HEX   FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7F10: FF FF FF >15   HEX   FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7F20: FF FF FF >16   HEX   FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7F30: FF FF FF >17   HEX   FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7F40: FF FF FF >18   HEX   FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7F50: FF FF FF >19   HEX   FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7F60: FF FF FF >20   HEX   FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7F70: FF FF FF >21   HEX   FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7F80: FF FF FF >22   HEX   FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7F90: FF FF FF >23   HEX   FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7FA0: FF FF FF >24   HEX   FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7FB0: FF FF FF >25   HEX   FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7FC0: FF FF FF >26   HEX   FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7FD0: FF FF FF >27   HEX   FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7FE0: FF FF FF >28   HEX   FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
7FF0: FF FF FF >29   HEX   FFFFFFFF
>30 *
>31 *****
>32 *                               *
>33 *Schakel naar andere BANK   *
>34 *                               *
>35 *****
>36 *
>37 *Schakel andere bank in door het
>38 *gewenste banknummer naar adresgebied
>39 *$Cn00-$CnFE te schrijven, banknummer
>40 *staat reeds in ACCU
>41 *In adres INDIRECT staat reeds waar naar
>42 *toe gesprongen wordt bij binnenkomst
>43 *andere bank
>44 *
>45 *Y-register wordt gebruikt
>46 *
7FF5: A0 00 >47   SCHAKELF IDY   #$0
7FF7: 09 80 >48   ORA   #SOFTBANK ;Zet softbankselectie bit
7FF9: 91 3A >49   STA   (INDIRECT),Y ;Schakel om
7FFB: 6C 3A 00 >50   JMP   (INDIRECT) ;Spring weg bij binnenkomst
>51 *
7FFE: 0F >52   DFB   BNKRESF ;Nummer van deze BANK
7FFF: FF >53   DFB   $FF ;Adres Controle Register
>54 *

```

VIERLINGKAART BESTURINGSSOFTWARE

```

>1 *****
>2 *****
>3 ***                                     ***
>4 ***      B A N K   N R   1 0   ***
>5 ***                                     ***
>6 *****
>7 *****
>8 *                                     *
>9 *bank BNKRES10 VERSIE 5.0/850607*
>10 *                                     *
>11 *****
>12 *
>13 *
8000: FF FF FF >14      HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8010: FF FF FF >15      HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8020: FF FF FF >16      HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8030: FF FF FF >17      HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8040: FF FF FF >18      HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8050: FF FF FF >19      HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8060: FF FF FF >20      HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8070: FF FF FF >21      HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8080: FF FF FF >22      HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8090: FF FF FF >23      HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
80A0: FF FF FF >24      HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
80B0: FF FF FF >25      HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
80C0: FF FF FF >26      HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
80D0: FF FF FF >27      HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
80E0: FF FF FF >28      HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
80F0: FF FF FF >29      HEX FFFFFFFF
>30 *
>31 *****
>32 *                                     *
>33 *Schakel naar andere BANK          *
>34 *                                     *
>35 *****
>36 *
>37 *Schakel andere bank in door het
>38 *gewenste banknummer naar adresgebied
>39 *$Cn00-$CnFE te schrijven, banknummer
>40 *staat reeds in ACCU
>41 *In adres INDIRECT staat reeds waar naar
>42 *toe gesprongen wordt bij binnenkomst
>43 *andere bank
>44 *
>45 *Y-register wordt gebruikt
>46 *
80F5: A0 00 >47      SCHAKEL1OLDY  #$0
80F7: 09 80 >48      ORA  #SOFTBANK ;Zet softbankselectie bit
80F9: 91 3A >49      STA  (INDIRECT),Y ;Schakel om
80FB: 6C 3A 00 >50     JMP  (INDIRECT) ;Spring weg bij binnenkomst
>51 *
80FE: 10 >52      DFB  BNKRES10 ;Nummer van deze BANK
80FF: FF >53      DFB  $FF ;Adres Controle Register
>54 *

```

VIERLINGKAART BESTURINGSSOFTWARE

```

>1 *****
>2 *****
>3 ***                                     ***
>4 ***   B A N K   N R   1 1   ***
>5 ***                                     ***
>6 *****
>7 *****
>8 *                                       *
>9 *bank ENKRES11 VERSIE 5.0/850607*
>10 *                                       *
>11 *****
>12 *
>13 *
8100: FF FF FF >14   HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8110: FF FF FF >15   HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8120: FF FF FF >16   HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8130: FF FF FF >17   HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8140: FF FF FF >18   HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8150: FF FF FF >19   HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8160: FF FF FF >20   HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8170: FF FF FF >21   HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8180: FF FF FF >22   HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8190: FF FF FF >23   HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
81A0: FF FF FF >24   HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
81B0: FF FF FF >25   HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
81C0: FF FF FF >26   HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
81D0: FF FF FF >27   HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
81E0: FF FF FF >28   HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
81F0: FF FF FF >29   HEX FFFFFFFFFF
>30 *
>31 *****
>32 *                                       *
>33 *Schakel naar andere BANK          *
>34 *                                       *
>35 *****
>36 *
>37 *Schakel andere bank in door het
>38 *gewenste banknummer naar adresgebied
>39 *$Cn00-$CnFE te schrijven, banknummer
>40 *staat reeds in ACCU
>41 *In adres INDIRECT staat reeds waar naar
>42 *toe gesprongen wordt bij binnenkomst
>43 *andere bank
>44 *
>45 *Y-register wordt gebruikt
>46 *
81F5: A0 00 >47   SCHAKEL11LDY  #$0
81F7: 09 80 >48   ORA  #SOFTBANK ;Zet softbankselectie bit
81F9: 91 3A >49   STA  (INDIRECT),Y ;Schakel om
81FB: 6C 3A 00 >50   JMP  (INDIRECT) ;Spring weg bij binnenkomst
>51 *
81FE: 11 >52   DFB  ENKRES11 ;Nummer van deze BANK
81FF: FF >53   DFB  $FF ;Adres Controle Register
>54 *

```

VIERLINGKAART BESTURINGSSOFTWARE

```

>1 *****
>2 *****
>3 ***                                     ***
>4 ***      B A N K   N R   1 2   ***
>5 ***                                     ***
>6 *****
>7 *****
>8 *                                       *
>9 *bank BNKRES12 VERSIE 5.0/850607*
>10 *                                       *
>11 *****
>12 *
>13 *
8200: FF FF FF >14   HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8210: FF FF FF >15   HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8220: FF FF FF >16   HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8230: FF FF FF >17   HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8240: FF FF FF >18   HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8250: FF FF FF >19   HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8260: FF FF FF >20   HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8270: FF FF FF >21   HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8280: FF FF FF >22   HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8290: FF FF FF >23   HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
82A0: FF FF FF >24   HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
82B0: FF FF FF >25   HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
82C0: FF FF FF >26   HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
82D0: FF FF FF >27   HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
82E0: FF FF FF >28   HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
82F0: FF FF FF >29   HEX FFFFFFFF
>30 *
>31 *****
>32 *                                       *
>33 *Schakel naar andere BANK          *
>34 *                                       *
>35 *****
>36 *
>37 *Schakel andere bank in door het
>38 *gewenste banknummer naar adresgebied
>39 *$Cn00-$CnFE te schrijven, banknummer
>40 *staat reeds in ACCU
>41 *In adres INDIRECT staat reeds waar naar
>42 *toe gesprongen wordt bij binnenkomst
>43 *andere bank
>44 *
>45 *Y-register wordt gebruikt
>46 *
82F5: A0 00 >47   SCHAKEL12LDY   #$0
82F7: 09 80 >48   ORA   #SOFTBANK ;Zet softbankselectie bit
82F9: 91 3A >49   STA   (INDIRECT),Y ;Schakel om
82FB: 6C 3A 00 >50   JMP   (INDIRECT) ;Spring weg bij binnenkomst
>51 *
82FE: 12 >52   DFB   BNKRES12 ;Nummer van deze BANK
82FF: FF >53   DFB   $FF ;Adres Controle Register
>54 *

```

VIERLINGKAART BESTURINGSSOFTWARE

```

>1 *****
>2 *****
>3 ***                               ***
>4 ***   B A N K   N R   1 3   ***
>5 ***                               ***
>6 *****
>7 *****
>8 *                                 *
>9 *bank BNKRES13 VERSIE 5.0/850607*
>10 *                                 *
>11 *****
>12 *
>13 *
8300: FF FF FF >14   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8310: FF FF FF >15   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8320: FF FF FF >16   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8330: FF FF FF >17   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8340: FF FF FF >18   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8350: FF FF FF >19   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8360: FF FF FF >20   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8370: FF FF FF >21   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8380: FF FF FF >22   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8390: FF FF FF >23   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
83A0: FF FF FF >24   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
83B0: FF FF FF >25   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
83C0: FF FF FF >26   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
83D0: FF FF FF >27   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
83E0: FF FF FF >28   HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
83F0: FF FF FF >29   HEX  FFFFFFFFFF
>30 *
>31 *****
>32 *                                 *
>33 *Schakel naar andere BANK       *
>34 *                                 *
>35 *****
>36 *
>37 *Schakel andere bank in door het
>38 *gewenste banknummer naar adresgebied
>39 *$Cn00-$CnFE te schrijven, banknummer
>40 *staat reeds in ACCU
>41 *In adres INDIRECT staat reeds waar naar
>42 *toe gesprongen wordt bij binnenkomst
>43 *andere bank
>44 *
>45 *Y-register wordt gebruikt
>46 *
83F5: A0 00 >47   SCHAKEL13LDY  #$0
83F7: 09 80 >48   ORA  #SOFTBANK ;Zet softbankselectie bit
83F9: 91 3A >49   STA  (INDIRECT),Y ;Schakel om
83FB: 6C 3A 00 >50   JMP  (INDIRECT) ;Spring weg bij binnenkomst
>51 *
83FE: 13 >52   DFB  BNKRES13 ;Nummer van deze BANK
83FF: FF >53   DFB  $FF ;Adres Controle Register
>54 *

```


VIERLINGKAART BESTURINGSSOFTWARE

```

>1 *****
>2 *****
>3 *** ***
>4 *** BANK NR 14 ***
>5 *** ***
>6 *****
>7 *****
>8 * *
>9 *bank BNKRES14 VERSIE 5.0/850607*
>10 * *
>11 *****
>12 *
>13 *
8400: FF FF FF >14 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8410: FF FF FF >15 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8420: FF FF FF >16 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8430: FF FF FF >17 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8440: FF FF FF >18 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8450: FF FF FF >19 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8460: FF FF FF >20 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8470: FF FF FF >21 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8480: FF FF FF >22 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8490: FF FF FF >23 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
84A0: FF FF FF >24 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
84B0: FF FF FF >25 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
84C0: FF FF FF >26 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
84D0: FF FF FF >27 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
84E0: FF FF FF >28 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
84F0: FF FF FF >29 HEX FFFFFFFF
>30 *
>31 *****
>32 * *
>33 *Schakel naar andere BANK *
>34 * *
>35 *****
>36 *
>37 *Schakel andere bank in door het
>38 *gewenste banknummer naar adresgebied
>39 *$Cn00-$CnFE te schrijven, banknummer
>40 *staat reeds in ACCU
>41 *In adres INDIRECT staat reeds waar naar
>42 *toe gesprongen wordt bij binnenkomst
>43 *andere bank
>44 *
>45 *Y-register wordt gebruikt
>46 *
84F5: A0 00 >47 SCHAKEL14LDY #$0
84F7: 09 80 >48 ORA #SOFTBANK ;Zet softbankselectie bit
84F9: 91 3A >49 STA (INDIRECT),Y ;Schakel om
84FB: 6C 3A 00 >50 JMP (INDIRECT) ;Spring weg bij binnenkomst
>51 *
84FE: 14 >52 DFB BNKRES14 ;Nummer van deze BANK
84FF: FF >53 DFB $FF ;Adres Controle Register
>54 *

```

VIERLINGKAART BESTURINGSSOFTWARE

```

>1 *****
>2 *****
>3 *** ***
>4 *** BANK NR 15 ***
>5 *** ***
>6 *****
>7 *****
>8 * *
>9 *bank BNKRES15 VERSIE 5.0/850607*
>10 * *
>11 *****
>12 *
>13 *
8500: FF FF FF >14 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8510: FF FF FF >15 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8520: FF FF FF >16 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8530: FF FF FF >17 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8540: FF FF FF >18 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8550: FF FF FF >19 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8560: FF FF FF >20 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8570: FF FF FF >21 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8580: FF FF FF >22 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8590: FF FF FF >23 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
85A0: FF FF FF >24 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
85B0: FF FF FF >25 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
85C0: FF FF FF >26 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
85D0: FF FF FF >27 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
85E0: FF FF FF >28 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
85F0: FF FF FF >29 HEX FFFFFFFF
>30 *
>31 *****
>32 * *
>33 *Schakel naar andere BANK *
>34 * *
>35 *****
>36 *
>37 *Schakel andere bank in door het
>38 *gewenste banknummer naar adresgebied
>39 *$Cn00-$CnFE te schrijven, banknummer
>40 *staat reeds in ACCU
>41 *In adres INDIRECT staat reeds waar naar
>42 *toe gesprongen wordt bij binnenkomst
>43 *andere bank
>44 *
>45 *Y-register wordt gebruikt
>46 *
85F5: A0 00 >47 SCHAKEL15LDY #$0
85F7: 09 80 >48 ORA #SOFTBANK ;Zet softbankselectie bit
85F9: 91 3A >49 STA (INDIRECT),Y ;Schakel om
85FB: 6C 3A 00 >50 JMP (INDIRECT) ;Spring weg bij binnenkomst
>51 *
85FE: 15 >52 DFB BNKRES15 ;Nummer van deze BANK
85FF: FF >53 DFB $FF ;Adres Controle Register
>54 *

```

VIERLINGKAART BESTURINGSSOFTWARE

```

>1 *****
>2 *****
>3 ***                               ***
>4 ***   B A N K   N R   1 6   ***
>5 ***                               ***
>6 *****
>7 *****
>8 *                                 *
>9 *bank BNKRES16 VERSIE 5.0/850607*
>10 *                                 *
>11 *****
>12 *
>13 *
8600: FF FF FF >14          HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8610: FF FF FF >15          HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8620: FF FF FF >16          HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8630: FF FF FF >17          HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8640: FF FF FF >18          HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8650: FF FF FF >19          HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8660: FF FF FF >20          HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8670: FF FF FF >21          HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8680: FF FF FF >22          HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8690: FF FF FF >23          HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
86A0: FF FF FF >24          HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
86B0: FF FF FF >25          HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
86C0: FF FF FF >26          HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
86D0: FF FF FF >27          HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
86E0: FF FF FF >28          HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
86F0: FF FF FF >29          HEX FFFFFFFF
>30 *
>31 *****
>32 *                                 *
>33 *Schakel naar andere BANK      *
>34 *                                 *
>35 *****
>36 *
>37 *Schakel andere bank in door het
>38 *gewenste banknummer naar adresgebied
>39 *$Cn00-$CnFE te schrijven, banknummer
>40 *staat reeds in ACCU
>41 *In adres INDIRECT staat reeds waar naar
>42 *toe gesprongen wordt bij binnenkomst
>43 *andere bank
>44 *
>45 *Y-register wordt gebruikt
>46 *
86F5: A0 00 >47 SCHAKEL16LDY #0
86F7: 09 80 >48          ORA #SOFTBANK ;Zet softbankselectie bit
86F9: 91 3A >49          STA (INDIRECT),Y ;Schakel om
86FB: 6C 3A 00 >50          JMP (INDIRECT) ;Spring weg bij binnenkomst
>51 *
86FE: 16 >52          DFB BNKRES16 ;Nummer van deze BANK
86FF: FF >53          DFB $FF ;Adres Controle Register
>54 *

```

VIERLINGKAART BESTURINGSSOFTWARE

```

>1 *****
>2 *****
>3 *** ***
>4 ***   B A N K   N R   1 7   ***
>5 *** ***
>6 *****
>7 *****
>8 * *
>9 *bank ENKRES17 VERSIE 5.0/850607*
>10 * *
>11 *****
>12 *
>13 *
8700: FF FF FF >14      HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8710: FF FF FF >15      HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8720: FF FF FF >16      HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8730: FF FF FF >17      HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8740: FF FF FF >18      HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8750: FF FF FF >19      HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8760: FF FF FF >20      HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8770: FF FF FF >21      HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8780: FF FF FF >22      HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8790: FF FF FF >23      HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
87A0: FF FF FF >24      HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
87B0: FF FF FF >25      HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
87C0: FF FF FF >26      HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
87D0: FF FF FF >27      HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
87E0: FF FF FF >28      HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
87F0: FF FF FF >29      HEX  FFFFFFFFFF
>30 *
>31 *****
>32 * *
>33 *Schakel naar andere BANK *
>34 * *
>35 *****
>36 *
>37 *Schakel andere bank in door het
>38 *gewenste banknummer naar adresgebied
>39 *$Cn00-$CnFE te schrijven, banknummer
>40 *staat reeds in ACCU
>41 *In adres INDIRECT staat reeds waar naar
>42 *toe gesprongen wordt bij binnenkomst
>43 *andere bank
>44 *
>45 *Y-register wordt gebruikt
>46 *
87F5: A0 00 >47 SCHAKEL17LDY #$0
87F7: 09 80 >48      ORA  #SOFTBANK ;Zet softbankselectie bit
87F9: 91 3A >49      STA  (INDIRECT),Y ;Schakel om
87FB: 6C 3A 00 >50      JMP  (INDIRECT) ;Spring weg bij binnenkomst
>51 *
87FE: 17 >52      DFB  ENKRES17 ;Nummer van deze BANK
87FF: FF >53      DFB  $FF ;Adres Controle Register
>54 *

```

VIERLINGKAART BESTURINGSSOFTWARE

```

>1 *****
>2 *****
>3 *** ***
>4 ***   B A N K   N R   1 8   ***
>5 *** ***
>6 *****
>7 *****
>8 * *
>9 *bank BNKRES18 VERSIE 5.0/850607*
>10 * *
>11 *****
>12 *
>13 *
8800: FF FF FF >14      HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8810: FF FF FF >15      HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8820: FF FF FF >16      HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8830: FF FF FF >17      HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8840: FF FF FF >18      HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8850: FF FF FF >19      HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8860: FF FF FF >20      HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8870: FF FF FF >21      HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8880: FF FF FF >22      HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8890: FF FF FF >23      HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
88A0: FF FF FF >24      HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
88B0: FF FF FF >25      HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
88C0: FF FF FF >26      HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
88D0: FF FF FF >27      HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
88E0: FF FF FF >28      HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
88F0: FF FF FF >29      HEX  FFFFFFFFFF
>30 *
>31 *****
>32 * *
>33 *Schakel naar andere BANK *
>34 * *
>35 *****
>36 *
>37 *Schakel andere bank in door het
>38 *gewenste banknummer naar adresgebied
>39 *$Cn00-$CnFE te schrijven, banknummer
>40 *staat reeds in ACCU
>41 *In adres INDIRECT staat reeds waar naar
>42 *toe gesprongen wordt bij binnenkomst
>43 *andere bank
>44 *
>45 *Y-register wordt gebruikt
>46 *
88F5: A0 00 >47  SCHAKEL18LDY  #$0
88F7: 09 80 >48      ORA  #SOFTBANK ;Zet softbankselectie bit
88F9: 91 3A >49      STA  (INDIRECT),Y ;Schakel om
88FB: 6C 3A 00 >50     JMP  (INDIRECT) ;Spring weg bij binnenkomst
>51 *
88FE: 18 >52      DFB  BNKRES18 ;Nummer van deze BANK
88FF: FF >53      DFB  $FF ;Adres Controle Register
>54 *

```

VIERLINGKAART BESTURINGSSOFTWARE

```

>1 *****
>2 *****
>3 ***                                     ***
>4 ***      B A N K   N R   1 9      ***
>5 ***                                     ***
>6 *****
>7 *****
>8 *                                     *
>9 *bank BNKTERML VERSIE 5.0/850607*
>10 *                                     *
>11 *****
>12 *
>13 *
>14 *Vervolg op bank BNKTERM terminal emulatie
>15 *vervolg initialisatie
>16 *
>17 *
8900: A9 78 >18 B_TERM1 LDA #<DATFMES ;zet INDIRECL voor boodschap
8902: 85 3A >19 STA INDIRECL ;gelijk al op indirect laden
8904: A0 00 >20 LDY #$0
8906: B1 3A >21 FMIMES LDA (INDIRECT),Y ;Vraag dataformat op
8908: F0 09 >22 BEQ RDDATAFM
890A: 20 ED FD >23 JSR COUT
890D: C8 >24 INY
890E: D0 F6 >25 BNE FMIMES
>26 *
8910: 20 DD FB >27 RDDATERR JSR BELL ;foute invoer
8913: 20 35 FD >28 RDDATAFM JSR RDCHAR ;haal invoer
8916: C9 B1 >29 CMP #"1" ;controleer
8918: 90 F6 >30 BLT RDDATERR
891A: C9 B6 >31 CMP #"6"
891C: 10 F2 >32 BPL RDDATERR
891E: 20 ED FD >33 JSR COUT ;echo op scherm
8921: 29 0F >34 AND #$0F ;maak decimaal
8923: A8 >35 TAY ;bewaar in YREG
8924: 88 >36 DEY ;maar wel vorm 0 t/m 4
>37 *
8925: 68 >38 PLA ;haal BPS terug(is 8 bits)
8926: C0 02 >39 CPY #$2 ;7 bits data?
8928: B0 02 >40 BCS T_ZETBPS ;nee
892A: 09 20 >41 ORA #%00100000 ;masker 7 bits data bij BPS in
892C: 8D 01 C8 >42 T_ZETBPS STA ACIASTAT ;RESET ACIA door write naar stat
892F: 8D 03 C8 >43 STA ACIACONT ;zet ACIA controlereg op BPS+bit
>44 *
8932: A9 73 >45 LDA #<T_COMMAN ;zet INDIRECT goed voor masker
8934: 85 3A >46 STA INDIRECL
8936: B1 3A >47 LDA (INDIRECT),Y;Haal dataformat bits
8938: 8D 02 C8 >48 STA ACIACOMM ;zet in ACIA commando register
>49 *
893B: A9 BA >50 LDA #<FHDUPMES ;zet INDIRECT voor boodschap
893D: 85 3A >51 STA INDIRECL
893F: A0 00 >52 LDY #$0
8941: B1 3A >53 DUPMES LDA (INDIRECT),Y ;vraag DUPLEX mode
8943: F0 09 >54 BEQ RDDUPLEX
8945: 20 ED FD >55 JSR COUT

```

VIERLINGKAART BESTURINGSSOFTWARE

```

8948: C8      >56      INY
8949: D0 F6   >57      BNE  DUPMES
      >58      *
894B: 20 DD FB >59      RDDUPERR JSR  BELL      ;foute invoer
894E: 20 35 FD >60      RDDUPLEX JSR  RDCHAR    ;haal invoer
8951: AE F8 07 >61      LDX  MSLOT    ;herstel XREG met $Cn
8954: 29 DF   >62      AND  #$DF     ;converteer naar UPCASE
8956: C9 C8   >63      CMP  #"H"     ;controleer op H of F
8958: F0 08   >64      BEQ  HDUPLEX
895A: C9 C6   >65      CMP  #"F"
895C: D0 ED   >66      BNE  RDDUPERR ;fout
895E: A0 80   >67      LDY  %10000000 ;8-e bit=1 geen echo=fullduplex
8960: D0 02   >68      BNE  DUPECHO  ;altijd genomen
8962: A0 00   >69      HDUPLEX LDY  %00000000 ;8-e bit=0 echo=half duplex
8964: 20 ED FD >70      DUPECHO JSR  COUT     ;echo ingetypte karakter
8967: 98      >71      TYA
8968: 9D 38 07 >72      STA  S MODE,X ;bewaars duplex mode
      >73      *De rest gaat in een andere bank
      >74      *Tref de nodige voorbereidingen voor
      >75      *vertrek
896B: A9 00   >76      LDA  #B_TERM2 ;haal sprongadres
896D: 85 3A   >77      STA  INDIRECL
896F: A9 1A   >78      LDA  #BNKTERM2 ;Nieuw banknummer
8971: D0 6A   >79      BNE  TUSTERM1 ;altijd genomen
      >80      *
      >81      *
      >82      *Tabel
      >83      *
      >84      *
8973: 6B      >85      T_COMMAN DFB  %01101011 ;7E+geen interrupts
8974: 2B      >86      DFB  %00101011 ;70+geen interrupts
8975: 0B      >87      DFB  %00001011 ;8N+geen interrupts
8976: 6B      >88      DFB  %01101011 ;8E+geen interrupts
8977: 2B      >89      DFB  %00101011 ;80+geen interrupts
      >90      *
      >91      *Boodschappen
      >92      *
8978: 8D 8D   >93      DATFMES DFB  CR,CR
897A: C7 C5 C5 >94      ASC  "GEEF DATAFORMAAT: 1=7-EVEN 2=7-ODD "
899D: 8D      >95      DFB  CR
899E: B3 BD B8 >96      ASC  "3=8-NONE 4=8-EVEN 5=8-ODD ?"
89B9: 00      >97      DFB  $0
      >98      *
89BA: 8D 8D   >99      FHDUPMES DFB  CR,CR
89BC: C7 C5 C5 >100     ASC  "GEEF DUPLEXMODE: H=HALF F=FULL ?"
89DC: 00      >101     DFB  $0
      >102     *
89DD: D0 16   >103     TUSTERM1 BNE  SCHAKEL19
      >104     *
      >105     *opvulling tot bankswitch op zijn plaats is
      >106     *
89DF: FF FF FF >107     HEX  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
89EF: FF FF FF >108     HEX  FFFFFFFFFF
      >109     *
      >110     *

```

VIERLINGKAART BESTURINGSSOFTWARE

```

>111 *
>112 *****
>113 *
>114 *Schakel naar andere BANK *
>115 *
>116 *****
>117 *
>118 *Schakel andere bank in door het
>119 *gewenste banknummer naar adresgebied
>120 *$Cn00-$CnFE te schrijven, banknummer
>121 *staat reeds in ACCU
>122 *In adres INDIRECT staat reeds waar naar
>123 *toe gesprongen wordt bij binnenkomst
>124 *andere bank
>125 *
>126 *Y-register wordt gebruikt
>127 *
89F5: A0 00 >128 SCHAKEL19LDY #$0
89F7: 09 80 >129 ORA #SOFTBANK ;Zet softbankselectie bit
89F9: 91 3A >130 STA (INDIRECT),Y ;Schakel om
89FB: 6C 3A 00 >131 JMP (INDIRECT) ;Spring weg bij binnenkomst
>132 *
89FE: 19 >133 DFB BNKTERML ;Nummer van deze BANK
89FF: FF >134 DFB $FF ;Adres Controle Register
>135 *

```


VIERLINGKAART BESTURINGSSOFTWARE

```

>1 *****
>2 *****
>3 ***                                     ***
>4 ***      B A N K   N R   1 A      ***
>5 ***                                     ***
>6 *****
>7 *****
>8 *                                     *
>9 *bank BNKTERM2 VERSIE 5.0/850607*
>10 *                                     *
>11 *****
>12 *
>13 *
>14 *Vervolg op BNKTERM1 terminal emulatie
>15 *In deze bank wordt de datacommunicatie
>16 *uitgevoerd.
>17 *
>18 *
>19 *EQU voor deze bank
>20 DEL      =      $FF      ;DEL karakter
>21 CURSOR   =      " _ "    ;Cursor
>22 *
8A00: AE F8 07 >23 B_TERM2  LDX  MSL0T
8A03: A9 00   >24          LDA  #$0      ;Clear de CTR-S vlag
8A05: 8D 04 CF >25          STA  CTRSMODE
8A08: 20 58 FC >26          JSR  HOME
8A0B: A9 8C   >27          LDA  #CTRL    ;ook 80 kolom schoon
8A0D: 20 ED FD >28          JSR  COUT
>29 *Beginpunt van alle toetsenbord akties
8A10: AD 04 CF >30 KARIN   LDA  CTRSMODE ;Is CTRL S aktief
8A13: D0 43   >31          BNE  ACIAINH  ;Ja geen input toegestaan
8A15: A9 DF   >32          LDA  #CURSOR ;Plaats cursor
8A17: 20 ED FD >33          JSR  COUT
8A1A: A9 88   >34          LDA  #BS     ;En 1 positie weer terug
8A1C: 20 ED FD >35          JSR  COUT
8A1F: AD 00 C0 >36          LDA  KBD     ;is er een toets ingedrukt
8A22: 10 34   >37          BPL  ACIAINH  ;Nee op naar ACIA
8A24: 8D 10 C0 >38          STA  KBDSTRB ;Ja, reset strobe
8A27: C9 9B   >39          CMP  #ESC    ;Is het ESC
8A29: D0 40   >40          BNE  ECHOIN  ;Nee ga evt echoen
>41 *Er is een ESC ingedrukt dit is de voorloper
>42 *voor een commando letter
8A2B: AD 00 C0 >43 COMKAR  LDA  KBD     ;Volgende toets
8A2E: 10 FB   >44          BPL  COMKAR  ;Wachten tot het er is
8A30: 8D 10 C0 >45          STA  KBDSTRB ;reset toetsenbord
8A33: C9 C5   >46          CMP  #"E"    ;Einde?
8A35: D0 15   >47          BNE  ISCOMD  ;Nee
>48 *Einde bereikt dus weer terug naar bank BNKTERM eerst ACI
8A37: AD 02 CF >49          LDA  RETBANK ;Pak terugkeerbank
8A3A: 48     >50          PHA
8A3B: AD 00 CF >51          LDA  RETADR  ;haal terugkeer adres
8A3E: 48     >52          PHA     ;bewaars even
8A3F: AD 03 CF >53          LDA  SAVCREG ;haal oude waarde CREG
8A42: A0 FF   >54          LDY  #CREGL
8A44: 91 3A   >55          STA  (INDIRECT),Y ;en plaats die weer

```

VIERLINGKAART BESTURINGSSOFTWARE

```

8A46: 68      >56      PLA
8A47: 85 3A   >57      STA  INDIRECL ;pas terugkeer adres aan
8A49: 68      >58      PLA ;en terugkeer bank
8A4A: B0 5A   >59      BCS  TUSTERM2 ;Spring in tweekeer naar schakel
8A4C: C9 C4   >60      ISCOMD CMP  #"D" ;Duplexmode wisselen
8A4E: D0 0D   >61      BNE  ISCOML ;Nee
      >62      *Wissel de duplexmode
8A50: BD 38 07 >63      LDA  S_MODE,X
8A53: 49 80   >64      EOR  #%10000000 ;wissel waarde
8A55: 9D 38 07 >65      STA  S_MODE,X ;Pas duplexmode aan
8A58: 18      >66      ACIAINH CLC
8A59: 90 4D   >67      BCC  ACIAIN ;altijd
      >68      *
8A5B: F0 B3   >69      KARINH BEQ  KARIN ;Tussen jumpje
      >70      *
8A5D: C9 CC   >71      ISCOML CMP  #"L" ;Auto linefeed mode wisselen
8A5F: D0 0A   >72      BNE  ECHOIN ;nee
8A61: BD 38 07 >73      LDA  S_MODE,X ;Haal de mode
8A64: 49 40   >74      EOR  #%01000000 ;flip auto LF bit
8A66: 9D 38 07 >75      STA  S_MODE,X ;bewaar
8A69: B0 3D   >76      BCS  ACIAIN ;altijd
      >77      *Echo de invoer als half duplex
8A6B: 48      >78      ECHOIN PHA ;Bewaar karakter
8A6C: BC 38 07 >79      LDY  S_MODE,X ;Pak duplex mode
8A6F: 30 1B   >80      EMI  ACIAOUT ;Full duplex
      >81      *Het is half duplex pas cursor bij BS,CR en LF aan
8A71: C9 88   >82      CMP  #BS ;Is het een BS
8A73: F0 08   >83      BEQ  HDAANPAS ;Ja
8A75: C9 8D   >84      CMP  #CR ;Is het een <cr>
8A77: F0 04   >85      BEQ  HDAANPAS ;Ja
8A79: C9 8A   >86      CMP  #LF ;Is het een <lf>
8A7B: D0 0C   >87      BNE  HDECHO ;Nee
8A7D: 48      >88      HDAANPAS PHA ;Bewaar karakter
8A7E: A9 A0   >89      LDA  #SPATIE
8A80: 20 ED FD >90      JSR  COUT ;verwijder cursor
8A83: A9 88   >91      LDA  #BS ;en 1 positie terug
8A85: 20 ED FD >92      JSR  COUT
8A88: 68      >93      PLA ;karakter terug
      >94      *
8A89: 20 ED FD >95      HDECHO JSR  COUT ;echo half duplex
      >96      *
8A8C: AD 01 C8 >97      ACIAOUT LDA  ACIASTAT ;Haal status
8A8F: 29 10   >98      AND  #TDRE ;Is transmitregister empty
8A91: F0 F9   >99      BEQ  ACIAOUT ;Nee wacht
8A93: 68      >100     PLA ;Haal karakter terug
8A94: 8D 00 C8 >101     STA  ACIADATA ;Laat het wegschrijven
8A97: C9 8D   >102     CMP  #CR ;controle op auto LF
8A99: D0 0D   >103     BNE  ACIAIN
8A9B: BD 38 07 >104     LDA  S_MODE,X ;haal mode
8A9E: 0A      >105     ASL ;7-e bit is auto LF mode
8A9F: 10 07   >106     BPL  ACIAIN ;Geen auto LF
8AA1: A9 8A   >107     LDA  #LF ;auto LF verzend LF
8AA3: 48      >108     PHA
8AA4: D0 E6   >109     BNE  ACIAOUT ;altijd
      >110     *

```

VIERLINGKAART BESTURINGSSOFTWARE

```

8AA6: B0 4D >111 TUSTERM2 BCS SCHAKELIA ;tussen jumpje
>112 *
>113 *Het volgende deel is de input van de ACIA
>114 *
8AA8: AD 01 C8 >115 ACIAIN LDA ACIASTAT ;Haal ACIA status
8AAB: 29 08 >116 AND #RDRF ;Is receive register full
8AAD: F0 AC >117 BEQ KARINH ;Nee ga op toetsenbord kijken
8AAF: AD 00 C8 >118 LDA ACIADATA ;Pak karakter
8AB2: 09 80 >119 ORA #$80 ;Zet 8-e bit op
8AB4: C9 A0 >120 CMP #SPATIE ;Eerste niet control karakter
8AB6: 90 06 >121 BLT CTRLKAR ;Het is controlkarakter
8AB8: C9 FF >122 CMP #DEL ;Is het een DEL
8ABA: D0 32 >123 BNE BEELDAF ;Nee druk maar af
8ABC: A9 88 >124 LDA #BS ;Ja maak er BS van
>125 *
8ABE: C9 88 >126 CTRLKAR CMP #BS ;<BS>?
8AC0: F0 08 >127 BEQ INAAPAS ;Ja
8AC2: C9 8D >128 CMP #CR ;<CR>?
8AC4: F0 04 >129 BEQ INAAPAS ;Ja
8AC6: C9 8A >130 CMP #LF ;<lf>?
8AC8: D0 0E >131 BNE ISCTRLS ;Nee
8ACA: 48 >132 INAAPAS PHA ;Bewaar karakter
8ACB: A9 A0 >133 LDA #SPATIE
8ACD: 20 ED FD >134 JSR COUT ;Verwijder cursor
8AD0: A9 88 >135 LDA #BS ;en 1 positie terug
8AD2: 20 ED FD >136 JSR COUT
8AD5: 68 >137 PLA ;Karakter terug
8AD6: D0 16 >138 BNE BEELDAF ;altijd
>139 *
>140 *
8AD8: C9 93 >141 ISCTRLS CMP #CTRLS ;Is het ctrl-S
8ADA: D0 07 >142 BNE ISCTRLQ ;Nee
8ADC: A9 01 >143 LDA #$01 ;Zet CTRL-S mode=stop uitvoer
8ADE: 8D 04 CF >144 STA CTRSMODE
8AE1: D0 C5 >145 BNE ACIAIN ;Ga volgend ACIA-kar halen
8AE3: C9 91 >146 ISCTRLQ CMP #CTRLQ ;Is het ctrl-Q
8AE5: D0 07 >147 BNE BEELDAF ;Nee
8AE7: A9 00 >148 LDA #$0 ;Ja clear CTR-S mode
8AE9: 8D 04 CF >149 STA CTRSMODE
8AEC: F0 BA >150 BEQ ACIAIN ;Ga op ACIA kijken
>151 *
8AEE: 20 ED FD >152 BEELDAF JSR COUT ;Beeld karakter op scherm af
8AF1: 18 >153 CLC
8AF2: 90 B4 >154 BCC ACIAIN ;kijk naar ACIA
>155 *
>156 *Opvulling om bankswitch op juiste plaats te zetten
>157 *
8AF4: FF >158 HEX FF
>159 *
>160 *
>161 *****
>162 * *
>163 *Schakel naar andere BANK *
>164 * *
>165 *****

```

VIERLINGKAART BESTURINGSSOFTWARE

```

>166 *
>167 *Schakel andere bank in door het
>168 *gewenste banknummer naar adresgebied
>169 *$Cn00-$CnFE te schrijven, banknummer
>170 *staat reeds in ACCU
>171 *In adres INDIRECT staat reeds waar naar
>172 *toe gesprongen wordt bij binnenkomst
>173 *andere bank
>174 *
>175 *Y-register wordt gebruikt
>176 *
8AF5: A0 00 >177 SCHAKEL1ALDY #$0
8AF7: 09 80 >178          ORA #SOFTBANK ;Zet softbankselectie bit
8AF9: 91 3A >179          STA (INDIRECT),Y ;Schakel om
8AFB: 6C 3A 00 >180        JMP (INDIRECT) ;Spring weg bij binnenkomst
>181 *
8AFE: 1A >182          DFB BNKTERM2 ;Nummer van deze BANK
8AFF: FF >183          DFB $FF ;Adres Controle Register
>184 *

```

VIERLINGKAART BESTURINGSSOFTWARE

```

>1 *****
>2 *****
>3 *** ***
>4 ***   B A N K   N R   1 B   ***
>5 *** ***
>6 *****
>7 *****
>8 * *
>9 * bank ENKALG1 VERSIE 5.0/850607*
>10 * *
>11 *****
>12 *
>13 *
>14 *vervolg van de bank BNKALG
>15 *In deze bank wordt een speciale bank geactiveerd
>16 *In deze bank wordt ook een EPROM geactiveerd
>17 *
>18 *
8B00: A9 C4 >19 B_ALG1 LDA #<BANKMES ;zet INDIRECT op voor boodschap
8B02: 85 3A >20 STA INDIRECL
>21 *vraag op welke bank geactiveerd moet worden
8B04: A0 00 >22 LDY #$0
8B06: B1 3A >23 ASKBANK1 LDA (INDIRECT),Y ;verzend boodschap
8B08: F0 06 >24 BEQ RESINDL
8B0A: 20 ED FD >25 JSR COUT
8B0D: C8 >26 INY
8B0E: D0 F6 >27 BNE ASKBANK1
>28 *Zet in INDIRECT een sprong naar een bekende RTS
8B10: A9 8E >29 RESINDL LDA #<CROUT ;bereid uitspringen via CROUT
8B12: 85 3A >30 STA INDIRECL ;voor, via de monitor
8B14: A9 FD >31 LDA #>CROUT
8B16: 85 3B >32 STA INDIRECH
8B18: D0 03 >33 BNE RDBANKNR ;altijd
>34 *
8B1A: 20 DD FB >35 RDBANKER JSR BELL
8B1D: 20 35 FD >36 RDBANKNR JSR RDCHAR ;haal invoer
8B20: C9 8D >37 CMP #CR
8B22: F0 1A >38 BEQ ERUITCR
8B24: C9 B1 >39 CMP #"1" ;controleer 1 t/m 9
8B26: 90 F2 >40 BLT RDBANKER ;fout
8B28: C9 BA >41 CMP #"9"+1
8B2A: 30 0B >42 BMI ECHOBANK ;banknr 1...9
8B2C: C9 C1 >43 CMP #"A" ;controleer A t/m F
8B2E: 90 EA >44 BLT RDBANKER ;fout
8B30: C9 C7 >45 CMP #"F"+1
8B32: 10 E6 >46 BPL RDBANKER
8B34: 18 >47 CLC ;maak van ASCII hexadecimaal
8B35: 69 09 >48 ADC #$09
8B37: 20 ED FD >49 ECHOBANK JSR COUT ;echo goede invoer
8B3A: 29 0F >50 AND #$0F ;maak er een getal van
8B3C: 10 02 >51 BPL ERUIT ;gereed
>52 *
8B3E: A9 00 >53 ERUITCR LDA #$0 ;terug naar bank 0
8B40: 10 72 >54 ERUIT BPL TUSALG1 ;CREG is niet gesaved
>55 *

```

VIERLINGKAART BESTURINGSSOFTWARE

```

>56 *****
>57 *
>58 *Hier komt vervolg van gebruikersEPROM
>59 *activeren, vervolg bank-0
8B42: A9 00 >60 B_EPROM LDA #0 ;reset INDIRECT
8B44: 85 3A >61 STA INDIRECL
8B46: AD 78 05 >62 LDA EPROMNR ;Maak getal uit evt ASCII
8B49: 29 0F >63 AND #$0F
8B4B: 8D 78 05 >64 STA EPROMNR
8B4E: AD F8 05 >65 LDA ENTRYNR ;maak getal uit evt ASCII
8B51: 29 0F >66 AND #$0F
8B53: 8D F8 05 >67 STA ENTRYNR
>68 *
8B56: A0 FF >69 LDY #CREGL ;haal CREG op
8B58: B1 3A >70 LDA (INDIRECT),Y
8B5A: 48 >71 PHA ;bewaar even
8B5B: 8D FF CF >72 STA CLRROM ;schakel extensie ROM uit
8B5E: A9 8E >73 LDA #RAMACTIV
8B60: 91 3A >74 STA (INDIRECT),Y ;zet kaart en RAM aan
8B62: 68 >75 PLA
8B63: 8D 03 CF >76 STA SAVCREG ;bewaar in RAM
8B66: 29 F8 >77 AND #$F8 ;maak EPROMnr schoon
8B68: 0D 78 05 >78 ORA EPROMNR ;EPROMnr erin
8B6B: 09 8E >79 ORA #RAMACTIV ;RAM en kaart aan
8B6D: 91 3A >80 STA (INDIRECT),Y ;activeer juiste EPROM
8B6F: AD 78 05 >81 LDA EPROMNR ;haal EPROMnr terug
8B72: F0 2A >82 BEQ EPROMRTS ;EPROMnr nul dus eruit
>83 *bepaal juist entrypnt
8B74: A0 FE >84 LDY #BANKNR ;Haal huidig banknummer vast
8B76: B1 3A >85 LDA (INDIRECT),Y
8B78: AA >86 TAX ;bewaar
8B79: AD F8 05 >87 LDA ENTRYNR ;vergelijk gevraagd entrypnt
8B7C: CD 04 C8 >88 CMP ENTRYANT ;met max aantal EP's
8B7F: B0 16 >89 BCS ENTERR ;foute boel
8B81: 0A >90 ASL ;entrypnt maal 2
8B82: A8 >91 TAY
8B83: B9 03 C8 >92 LDA ENTRYTAB-2,Y ;pak laag adres entrypnt
8B86: 85 3A >93 STA INDIRECL ;in INDIRECT
8B88: C8 >94 INY
8B89: B9 03 C8 >95 LDA ENTRYTAB-2,Y ;pak hoog adres entrypnt
8B8C: 85 3B >96 STA INDIRECH ;ook in in DIRECT
8B8E: A9 8B >97 LDA #>EPROMRTS-1 ;terugkeer adres hoog
8B90: 48 >98 PHA ;op stack
8B91: A9 9D >99 LDA #<EPROMRTS-1 ;terugkeer adres laag
8B93: 48 >100 PHA ;ook op stack, terugkeer via RTS
8B94: 8A >101 TXA ;banknummer terugkomst
8B95: D0 1D >102 BNE TUSALG1 ;altijd (banknummer <> nul)
>103 *
8B97: A2 9B >104 ENTERR LDX #<ERRRTS ;foute terugkeer
8B99: D0 05 >105 BNE EPROMRT1
>106 *
8B9B: 20 DD FB >107 ERRRTS JSR BELL ;fout
8B9E: A2 8C >108 EPROMRTS LDX #<ALGRTS ;Pak terugkeer adres in bank0
8BA0: A9 00 >109 EPROMRT1 LDA #$0 ;zet INDIRECT juist
8BA2: 85 3A >110 STA INDIRECL

```

VIERLINGKAART BESTURINGSSOFTWARE

```

8BA4: AD 01 CF >111      LDA  SLOTADR
8BA7: 85 3B      >112      STA  INDIRECH
8BA9: A0 FF      >113      LDY  #CREGL
8BAB: AD 03 CF >114      LDA  SAVCREG      ;zet CREG terug
8BAE: 91 3A      >115      STA  (INDIRECT),Y
8BB0: 86 3A      >116      STX  INDIRECL      ;zet INDIRECT op return adres
8BB2: A9 00      >117      LDA  #$0          ;returnbank
8BB4: 48          >118      TUSALG1 PHA      ;bewaar ACCU
8BB5: A9 8C      >119      LDA  #CTRL      ;maak scherm 80kol schoon
8BB7: 20 ED FD >120      JSR  COUT
8BBA: 20 58 FC >121      JSR  HOME      ;maak scherm schoon
8BBD: 68          >122      PLA          ;haal banknummer terug
8BBE: 10 2C      >123      BPL  SCHAKEL1B ;schakel naar terugkeer bank
>124 *
8BC0: FF FF FF >125      HEX  FFFFFFFF
>126 *
>127 *boodschappen
>128 *
>129 *
8BC4: 8D 8D      >130      BANKMES DFB  CR,CR
8BC6: C7 C5 C5 >131      ASC  "GEEF NUMMER IN TE SCHAKELEN BANK 1-F?"
8BEB: 00          >132      DFB  $0
>133 *
>134 *.....DIT IS HEEL ERG GOOR.....
>135 *DEZE ENE MAAL WORDT VIA INDGOOR GESCHAKELD
>136 *EN NIET VIA INDIRECT
>137 *oorzaak op INDIRECT staat een wild adres van
>138 *een EPROM of een bank in, en hoeft
>139 *absoluut NIET naar $CnXX te wijzen.
>140 *
8BEC: AE F8 07 >141      SCHAKEL1BLDX MSLOT      ;hoog slotadres
8BEF: 86 09      >142      STX  INDGOORH      ;zet weg
8BF1: A2 00      >143      LDX  #$0          ;laag slotadres
8BF3: 86 08      >144      STX  INDGOORL
>145 *
>146 *
>147 *****
>148 *
>149 *Schakel naar andere BANK
>150 *
>151 *****
>152 *
>153 *Schakel andere bank in door het
>154 *gewenste banknummer naar adresgebied
>155 *$Cn00-$CnFE te schrijven, banknummer
>156 *staat reeds in ACCU
>157 *In adres INDIRECT staat reeds waar naar
>158 *toe gesprongen wordt bij binnenkomst
>159 *andere bank
>160 *
>161 *Y-register wordt gebruikt
>162 *
8BF5: A0 00      >163      LDY  #$0
8BF7: 09 80      >164      ORA  #SOFTBANK ;Zet softbankselectie bit
8BF9: 91 08      >165      STA  (INDGOOR),Y ;Schakel om

```

VIERLINGKAART BESTURINGSSOFTWARE

| | | | |
|----------------|--------|----------------|-----------------------------|
| 8BFB: 6C 3A 00 | >166 | JMP (INDIRECT) | ;Spring weg bij binnenkomst |
| | >167 * | | |
| 8BFE: 1B | >168 | DFB BNKALG1 | ;Nummer van deze BANK |
| 8BFF: FF | >169 | DFB \$FF | ;Adres Controle Register |
| | >170 * | | |

VIERLINGKAART BESTURINGSSOFTWARE

```

>1 *****
>2 *****
>3 *** ***
>4 ***   B A N K   N R   1 C   ***
>5 *** ***
>6 *****
>7 *****
>8 * *
>9 * bank BNKSER1 VERSIE 5.0/850607*
>10 * *
>11 *****
>12 *
>13 *
>14 *Vervolg van de seriele printer driver bank BNKSER
>15 *
>16 *
8C00: BD 38 07 >17 B_SER1 LDA S_MODE,X ;haal MODE op
8C03: 29 E0 >18 AND #11100000 ;zet B0=spatie, B1=LF EN B2=CR
8C05: 9D 38 07 >19 STA S_MODE,X ;B3=FF generatie af
8C08: A8 >20 TAY ;bewaar voor later
8C09: 30 49 >21 BMI NAARACIA ;video staat aan GEEN controle
8C0B: 68 >22 PLA ;haal karakter even terug
8C0C: 48 >23 PHA ;terug op stack weer
8C0D: C9 8C >24 CMP #FF ;is het een Formfeed
8C0F: D0 0F >25 BNE S_KOLCON ;nee
8C11: 98 >26 TYA ;haal MODE
8C12: 29 20 >27 AND #00100000 ;moet FF gegenereerd worden
8C14: F0 0A >28 BEQ S_KOLCON ;nee
8C16: 98 >29 TYA ;pas MODE aan
8C17: 09 0A >30 ORA #00001010 ;zet FF en LF generatie bit
8C19: 9D 38 07 >31 STA S_MODE,X ;bewaar
8C1C: 68 >32 PLA ;verwijder karakter van stack
8C1D: A9 8A >33 LDA #LF ;vervang FF door LF
8C1F: 48 >34 PHA ;karakter weer op stack
8C20: BD 38 06 >35 S_KOLCON LDA S_REGLEN,X ;haal gebruikers regelbreedte
8C23: C9 FF >36 CMP #FF ;regelbreedte controle gewenst
8C25: F0 2D >37 BEQ NAARACIA ;nee, karakter printen
8C27: DD 38 05 >38 CMP S_COL,X ;is regel vol
8C2A: D0 0E >39 BNE S_TABCON ;nee
8C2C: BD 38 07 >40 LDA S_MODE,X ;haal MODE
8C2F: 09 04 >41 ORA #00000100 ;zet CR-generatie bit aan
8C31: 9D 38 07 >42 STA S_MODE,X
8C34: A9 8D >43 LDA #CR ;genereer CR, regel is vol
8C36: D0 18 >44 BNE S_KARPHA ;altijd
>45 *
8C38: D0 C6 >46 BEGSER1 BNE B_SER1 ;tussenjumpje
>47 *
8C3A: BD 38 05 >48 S_TABCON LDA S_COL,X
8C3D: DD B8 03 >49 CMP S_LMARGE,X ;linker marge generatie
8C40: 90 04 >50 BCC S_DOESPA ;ja
8C42: C5 24 >51 CMP CH ;BASIC TAB of ,
8C44: B0 0B >52 BCS S_KARINC ;nee
>53 *
8C46: BD 38 07 >54 S_DOESPA LDA S_MODE,X ;haal MODE
8C49: 09 01 >55 ORA #00000001 ;zet spatiegeneratie bit aan

```

VIERLINGKAART BESTURINGSSOFTWARE

```

8C4B: 9D 38 07 >56          STA  S_MODE,X      ;bewaar
8C4E: A9 A0      >57          LDA  #SPATIE       ;spatie
8C50: 48         >58          S_KARPHA PHA      ;karakter op stack
8C51: FE 38 05 >59          S_KARINC INC      S_COL,X      ;verhoog kolomteller, gebeurt
      >60          *              ;overbodig voor CR,LF generatie
      >61          *
8C54: AD 01 C8 >62          NAARACIA LDA  ACIASTAT  ;haal status
8C57: 29 08      >63          AND  #RDRF        ;is er een karakter binnen gekom
8C59: F0 19      >64          BEQ  S_ACIAOUT    ;nee
      >65          *
8C5B: AD 00 C8 >66          LDA  ACIADATA     ;pak binnegekomen data
8C5E: 09 80      >67          ORA  #$80         ;8-e bit op
8C60: C9 93      >68          CMP  #CTRLS      ;control-S?
8C62: D0 10      >69          BNE  S_ACIAOUT    ;nee, doorgaan
      >70          *
8C64: AD 01 C8 >71          WACHTQ LDA  ACIASTAT  ;er was ctrl-S dus wachten op ct
8C67: 29 08      >72          AND  #RDRF
8C69: F0 F9      >73          BEQ  WACHTQ
8C6B: AD 00 C8 >74          LDA  ACIADATA     ;pak binnengekomen data
8C6E: 09 80      >75          ORA  #$80         ;zet 8-e bit aan
8C70: C9 91      >76          CMP  #CTRLQ      ;control-Q?
8C72: D0 F0      >77          BNE  WACHTQ      ;nee
      >78          *
8C74: AD 01 C8 >79          S_ACIAOUTLDA  ACIASTAT  ;statusregister
8C77: 29 10      >80          AND  #TDRE       ;mag er uitgevoerd worden
8C79: F0 D9      >81          BEQ  NAARACIA    ;nee, wachten
8C7B: 68         >82          PLA             ;pak karakter van stack
8C7C: 8D 00 C8 >83          STA  ACIADATA     ;plaats in ACIA
      >84          *
      >85          *
8C7F: C9 8D      >86          CMP  #CR         ;was het een CR
8C81: F0 15      >87          BEQ  S_WASCR     ;ja
8C83: C9 8A      >88          CMP  #LF        ;was het LF
8C85: F0 23      >89          BEQ  S_WASLFF   ;ja
8C87: BD 38 07 >90          S_EXIT  LDA  S_MODE,X  ;haal MODE
8C8A: 29 05      >91          AND  #%00000101 ;moet er nog wat gebeuren
8C8C: D0 AA      >92          BNE  BEGSERI    ;verzonden karakter was hulpkara
      >93          *spring nu terug naar bank BNKSER
8C8E: AD 00 CF >94          LDA  RETADR      ;plaats indirect sprong adres
8C91: 85 3A      >95          STA  INDIRECL
8C93: AD 02 CF >96          LDA  RETBANK     ;terugkeer banknummer
8C96: D0 5D      >97          BNE  SCHAKELIC  ;altijd
      >98          *
      >99          *
8C98: BD 38 07 >100         S_WASCR LDA  S_MODE,X  ;CR, LF erachter nodig
8C9B: 0A         >101         ASL                                     ;dan B6=1
8C9C: 10 0C      >102         BPL  S_WASLFF   ;nee, doe alsof LF
8C9E: BD 38 07 >103         S_GENLF LDA  S_MODE,X  ;pas MODE aan voor LF generatie
8CA1: 09 02      >104         ORA  #%00000010 ;zet LF bit
8CA3: 9D 38 07 >105         STA  S_MODE,X  ;zet terug
8CA6: A9 8A      >106         LDA  #LF        ;laad LF
8CA8: D0 A6      >107         BNE  S_KARPHA   ;altijd, printem
      >108         *
8CAA: BD 38 04 >109         S_WASLFF LDA  S_PAGLEN,X ;paginering gewenst
8CAD: F0 21      >110         BEQ  S_CHNUL    ;nee

```

VIERLINGKAART BESTURINGSSOFTWARE

```

8CAF: 38      >111      SEC
8CB0: FE B8 04 >112      INC S_LINE,X ;verhoog regelteller
8CB3: FD B8 04 >113      SBC S_LINE,X ;blad vol?
8CB6: 30 09    >114      EMI S_BLZVOL ;ja
8CB8: BD 38 07 >115      LDA S_MODE,X ;haal MODE
8CBB: 29 08    >116      AND #%00001000 ;Formfeed generatie
8CBD: F0 11    >117      BEQ S_CHNUL ;nee
8CBF: D0 DD    >118      BNE S_GENLF ;genereer meer LF's
8CC1: 49 F8    >119      S_BLZVOL EOR #$0-$8 ;al op volgend blad
8CC3: D0 D9    >120      BNE S_GENLF ;nee, genereer meer LF's
8CC5: 9D B8 04 >121      STA S_LINE,X ;reset regelteller
8CC8: BD 38 07 >122      LDA S_MODE,X
8CCB: 29 F5    >123      AND #%11110101 ;reset FF en LF generatie
8CCD: 9D 38 07 >124      STA S_MODE,X
>125      *
8CD0: BD 38 07 >126      S_CHNUL LDA S_MODE,X ;haal MODE
8CD3: A8      >127      TAY ;bewaars even
8CD4: 29 ED    >128      AND #%11101101 ;reset LF mode
8CD6: 9D 38 07 >129      STA S_MODE,X
8CD9: 98      >130      TYA ;haal terug
8CDA: 0A      >131      ASL ;video echo bit in CARRY
8CDB: 29 04    >132      AND #%00000010*2 ;echte LF
8CDD: F0 A8    >133      BEQ S_EXIT ;ja geen teller resetten
8CDF: A9 00    >134      LDA #$0
8CE1: 9D 38 05 >135      STA S_COL,X ;reset 4-ling kolomteller
8CE4: B0 A1    >136      BCS S_EXIT ;als video aan afblijven van CH
8CE6: 85 24    >137      STA CH
8CE8: 90 9D    >138      BCC S_EXIT ;altijd
>139      *
>140      *opvullen om bankswitch op zijn plaats te zetten
8CEA: FF FF FF >141      HEX FFFFFFFFFFFFFFFFFFFFFFFF
>144      *****
>145      *
>146      *Schakel naar andere BANK *
>147      *
>148      *****
>149      *
>150      *Schakel andere bank in door het
>151      *gewenste banknummer naar adresgebied
>152      *$Cn00-$CnFE te schrijven, banknummer
>153      *staat reeds in ACCU
>154      *In adres INDIRECT staat reeds waar naar
>155      *toe gesprongen wordt bij binnenkomst
>156      *andere bank
>157      *
>158      *Y-register wordt gebruikt
>159      *
8CF5: A0 00    >160      SCHAKEL1CLDY #$0
8CF7: 09 80    >161      ORA #SOFTBANK ;Zet softbankselectie bit
8CF9: 91 3A    >162      STA (INDIRECT),Y ;Schakel om
8CFB: 6C 3A 00 >163      JMP (INDIRECT) ;Spring weg bij binnenkomst
>164      *
8CFE: 1C      >165      DFB BNKSER1 ;Nummer van deze BANK
8CFF: FF      >166      DFB $FF ;Adres Controle Register

```

VIERLINGKAART BESTURINGSSOFTWARE

```

>1 *****
>2 *****
>3 ***          ***
>4 ***      B A N K   N R   1 D   ***
>5 ***          ***
>6 *****
>7 *****
>8 *
>9 * bank ENKPAR1 VERSIE 5.0/850607*
>10 *
>11 *****
>12 *
>13 *
>14 *Vervolg van de parolle printer driver bank ENKPAR
>15 *
>16 *
8D00: BD 38 07 >17 B_PAR1 LDA P_MODE,X ;haal MODE op
8D03: 29 E0 >18 AND #11100000 ;zet B0=spatie, B1=LF EN B2=CR
8D05: 9D 38 07 >19 STA P_MODE,X ;B3=FF generatie af
8D08: A8 >20 TAY ;bewaars voor later
8D09: 30 49 >21 BMI NAARVIA ;video staat aan GEEN controle
8D0B: 68 >22 PLA ;haal karakter even terug
8D0C: 48 >23 PHA ;en weer terug op stack
8D0D: C9 8C >24 CMP #FF ;is het een Formfeed
8D0F: D0 0F >25 BNE P_KOLCON ;nee
8D11: 98 >26 TYA ;haal MODE
8D12: 29 20 >27 AND #00100000 ;moet FF gegenereerd worden
8D14: F0 0A >28 BEQ P_KOLCON ;nee
8D16: 98 >29 TYA ;pas MODE aan
8D17: 09 0A >30 ORA #00001010 ;zet FF en LF generatie bit
8D19: 9D 38 07 >31 STA P_MODE,X ;bewaars
8D1C: 68 >32 PLA ;verwijder oud karakter van stack
8D1D: A9 8A >33 LDA #LF ;vervang FF door LF
8D1F: 48 >34 PHA ;karakter weer op stack
8D20: BD 38 06 >35 P_KOLCON LDA P_REGLN,X ;haal gebruikers regelbreedte
8D23: C9 FF >36 CMP #FF ;regelbreedte controle gewenst
8D25: F0 2D >37 BEQ NAARVIA ;nee, karakter printen
8D27: DD 38 05 >38 CMP P_COL,X ;is regel vol
8D2A: D0 0E >39 BNE P_TABCON ;nee
8D2C: BD 38 07 >40 LDA P_MODE,X ;haal MODE
8D2F: 09 04 >41 ORA #00000100 ;zet CR-generatie bit aan
8D31: 9D 38 07 >42 STA P_MODE,X
8D34: A9 8D >43 LDA #CR ;genereer CR, regel is vol
8D36: D0 18 >44 BNE P_KARPHA ;altijd
>45 *
8D38: D0 C6 >46 BEGPARI BNE B_PAR1 ;tussen jumpje
>47 *
8D3A: BD 38 05 >48 P_TABCON LDA P_COL,X
8D3D: DD B8 03 >49 CMP P_IMARGE,X ;linker marge generatie
8D40: 90 04 >50 BCC P_DOESPA ;ja
8D42: C5 24 >51 CMP CH ;BASIC TAB of ,
8D44: B0 0B >52 BCS P_KARINC ;nee
>53 *
8D46: BD 38 07 >54 P_DOESPA LDA P_MODE,X ;haal MODE
8D49: 09 01 >55 ORA #00000001 ;zet spatiegeneratie bit aan

```

VIERLINGKAART BESTURINGSSOFTWARE

```

8D4B: 9D 38 07 >56          STA  P_MODE,X      ;bewaar
8D4E: A9 A0      >57          LDA  #SPATIE       ;spatie
8D50: 48         >58          P_KARPHA PHA       ;karakter op stack
8D51: FE 38 05 >59          P_KARINC INC  P_COL,X ;verhoog kolomteller, gebeurt
      >60          *                ;overbodig voor CR,LF generatie
      >61          *
8D54: AC F8 06 >62          NAARVIA LDY  NO      ;haal slot maal 16
8D57: AD 78 04 >63          LDA  INITF        ;is dit de eerste maal
8D5A: F0 07      >64          BEQ  ACKLOOP      ;nee, kijken of VIA gereed is
8D5C: A9 00      >65          LDA  #$0          ;reset de initiele vlag
8D5E: 8D 78 04 >66          STA  INITF
8D61: F0 07      >67          BEQ  P_DOORG      ;altijd, niet kijken meer
8D63: B9 8D C0 >68          ACKLOOP LDA  VIA_IFR,Y ;haal interrupt vlag register
8D66: 29 10      >69          AND  #%00010000 ;VIA al gereed
8D68: F0 F9      >70          BEQ  ACKLOOP      ;nee, wachten
      >71          *karakter kan verstuurd worden
8D6A: 68         >72          P_DOORG PLA       ;pak karakter
8D6B: 48         >73          PHA       ;bewaar weer
8D6C: 99 80 C0 >74          STA  VIA_IORB,Y  ;zet kar in data registerB
      >75          *                ;dit reset tevens de IFR vlag
8D6F: B9 8C C0 >76          LDA  VIA_PCR,Y  ;ga een write strobe geven
8D72: 49 20      >77          EOR  #%00100000 ;flip CB-bit om
8D74: 99 8C C0 >78          STA  VIA_PCR,Y  ;geef puls
8D77: B9 8C C0 >79          LDA  VIA_PCR,Y  ;haal PCR waarde weer
8D7A: 49 20      >80          EOR  #%00100000 ;flip CB bit weer om
8D7C: 99 8C C0 >81          STA  VIA_PCR,Y  ;zet puls weer af
8D7F: 68         >82          PLA       ;pak karakter weer terug
      >83          *
      >84          *
8D80: C9 8D      >85          CMP  #CR         ;was het een CR
8D82: F0 15      >86          BEQ  P_WASCR     ;ja
8D84: C9 8A      >87          CMP  #LF         ;was het LF
8D86: F0 23      >88          BEQ  P_WASLF     ;ja
8D88: BD 38 07 >89          P_EXIT LDA  P_MODE,X ;haal MODE
8D8B: 29 05      >90          AND  #%00000101 ;moet er nog wat gebeuren
8D8D: D0 A9      >91          BNE  BEGPARI    ;verzonden karakter was hulpkara
      >92          *spring nu terug naar bank BNKPAR
8D8F: AD 00 CF >93          LDA  RETADR      ;plaats indirect sprong adres
8D92: 85 3A      >94          STA  INDIRECL
8D94: AD 02 CF >95          LDA  RETBANK     ;terugkeer banknummer
8D97: D0 5C      >96          BNE  SCHAKEL1D  ;altijd
      >97          *
      >98          *
8D99: BD 38 07 >99          P_WASCR LDA  P_MODE,X ;CR, LF erachter nodig
8D9C: 0A         >100         ASL                ;dan B6=1
8D9D: 10 0C      >101         BPL  P_WASLF     ;nee, doe alsof LF
8D9F: BD 38 07 >102         P_GENLF LDA  P_MODE,X ;pas MODE aan voor LF generatie
8DA2: 09 02      >103         ORA  #%00000010 ;zet LF bit
8DA4: 9D 38 07 >104         STA  P_MODE,X   ;zet terug
8DA7: A9 8A      >105         LDA  #LF        ;laad LF
8DA9: D0 A5      >106         BNE  P_KARPHA   ;altijd, printem
      >107         *
8DAB: BD 38 04 >108         P_WASLF LDA  P_PAGLEN,X ;paginering gewenst
8DAE: F0 21      >109         BEQ  P_CHNUL     ;nee
8DB0: 38         >110         SEC

```

VIERLINGKAART BESTURINGSSOFTWARE

```

8DB1: FE B8 04 >111      INC  P_LINE,X      ;verhoog regelteller
8DB4: FD B8 04 >112      SBC  P_LINE,X      ;blad vol?
8DB7: 30 09      >113      EMI  P_BLZVOL      ;ja
8DB9: BD 38 07 >114      LDA  P_MODE,X      ;haal MODE
8DBC: 29 08      >115      AND  #%00001000    ;Formfeed generatie
8DBE: F0 11      >116      BEQ  P_CHNUL       ;nee
8DC0: D0 DD      >117      BNE  P_GENLF       ;genereer meer LF's
8DC2: 49 F8      >118      P_BLZVOL EOR  #$0-$8    ;al op volgend blad
8DC4: D0 D9      >119      BNE  P_GENLF       ;nee, genereer meer LF's
8DC6: 9D B8 04 >120      STA  P_LINE,X      ;reset regelteller
8DC9: BD 38 07 >121      LDA  P_MODE,X      ;
8DCC: 29 F5      >122      AND  #%11110101    ;reset FF en LF generatie
8DCE: 9D 38 07 >123      STA  P_MODE,X      ;
      >124      *
8DD1: BD 38 07 >125      P_CHNUL LDA  P_MODE,X      ;haal MODE
8DD4: A8          >126      TAY          ;bewaar even
8DD5: 29 ED      >127      AND  #%11101101    ;reset LF mode
8DD7: 9D 38 07 >128      STA  P_MODE,X      ;
8DDA: 98          >129      TYA          ;haal terug
8ddb: 0A          >130      ASL          ;video echo bit in CARRY
8DDC: 29 04      >131      AND  #%00000010*2 ;echte LF
8DDE: F0 A8      >132      BEQ  P_EXIT       ;ja geen teller resetten
8DE0: A9 00      >133      LDA  #$0          ;
8DE2: 9D 38 05 >134      STA  P_COL,X      ;reset 4-ling kolomteller
8DE5: B0 A1      >135      BCS  P_EXIT       ;als video aan afblijven van CH
8DE7: 85 24      >136      STA  CH           ;
8DE9: 90 9D      >137      BCC  P_EXIT       ;altijd
      >138      *
      >139      *opvullen om bankswitch op zijn plaats te zetten
8DEB: FF FF FF >140      HEX  FFFFFFFFFFFFFFFFFF
      >142      *
      >143      *****
      >144      *
      >145      *Schakel naar andere BANK      *
      >146      *
      >147      *****
      >148      *
      >149      *Schakel andere bank in door het
      >150      *gewenste banknummer naar adresgebied
      >151      *$Cn00-$CnFE te schrijven, banknummer
      >152      *staat reeds in ACCU
      >153      *In adres INDIRECT staat reeds waar naar
      >154      *toe gesprongen wordt bij binnenkomst
      >155      *andere bank
      >156      *
      >157      *Y-register wordt gebruikt
      >158      *
8DF5: A0 00      >159      SCHAKEL1LDLY  #$0
8DF7: 09 80      >160      ORA  #SOFTBANK    ;Zet softbankselectie bit
8DF9: 91 3A      >161      STA  (INDIRECT),Y ;Schakel om
8DFB: 6C 3A 00 >162      JMP  (INDIRECT)   ;Spring weg bij binnenkomst
      >163      *
8DFE: 1D          >164      DFB  BNKPAR1     ;Nummer van deze BANK
8DFF: FF          >165      DFB  $FF         ;Adres Controle Register

```

VIERLINGKAART BESTURINGSSOFTWARE

```

>1 *****
>2 *****
>3 *** ***
>4 ***   B A N K   N R   1 E   ***
>5 *** ***
>6 *****
>7 *****
>8 * *
>9 *bank BNKKLOK1 VERSIE 5.0/850607*
>10 * *
>11 *****
>12 *
>13 *
>14 * K L O K   U I T L E Z I N G
>15 *
>16 *Na aanroep staat in RAM (KLOKDATA) de
>17 *uitlezing in de volgende layout (in ASCII ):
>18 *UU;MM;SS JJMMDD W<CR>
>19 *adres KLOKDATA=$CF10 t/m $CF21
>20 *
>21 *
>22 *eigen EQU
>23 *
>24 *Beschrijving KREG
>25 *bit0...3:adreslijnen op klok
>26 *bit 4   :HOLD mode
>27 *bit 5   :ADJUST
>28 *bit 6   :READ mode
>29 *bit 7   :INIERRUPT enable
>30 *           76543210 bitnummers
>31 HOLD    =   %00011111 ;HOLD lijn KREG
>32 CLEAR   =   %10001111 ;geen selectie wel interrupts
>33 HOLD_RD =   %01010000 ;HOLD en READ mode
>34 HOLD_WR =   %00010000 ;schrijf masker
>35 KLOK_AAN = %10001110 ;Zet de klok op de 4ling aan
>36 M_ASCII =   $B0      ;maak van decimaal getal ASCII
>37 M_DECIMAL = $0F      ;masker voor ASCII> binair
>38 *
8E00: A9 00 >39 B_KLOK1 LDA #$0
8E02: 85 3A >40 STA INDIRECL ;herstel INDIRECT
8E04: A0 FF >41 LDY #CREGL ;bewaar CREG
8E06: B1 3A >42 LDA (INDIRECT),Y
8E08: 8D 78 07 >43 STA SCRAT1 ;SAVCREG is al ingebruik
8E0B: 8A >44 TXA ;bewaar XREG
8E0C: 48 >45 PHA ;even op stack
8E0D: A9 8E >46 LDA #KLOK_AAN ;selekteer de klok
8E0F: 91 3A >47 STA (INDIRECT),Y
8E11: AD FD CF >48 LDA DREG ;bewaar status informatie
8E14: 48 >49 PHA ;zonder interrupt info
8E15: A9 1F >50 LDA #HOLD ;HOLD zetten
8E17: 8D FE CF >51 STA KREG ;zet in klok
8E1A: A9 09 >52 LDA #$9 ;wacht >150 muSEC (=175 muSEC)
8E1C: 38 >53 SEC
8E1D: 48 >54 K_WAIT2 PHA
8E1E: E9 01 >55 K_WAIT3 SBC #$01

```

VIERLINGKAART BESTURINGSSOFTWARE

```

8E20: D0 FC >56      BNE K_WAIT3
8E22: 68 >57        PLA
8E23: E9 01 >58      SBC #$01
8E25: D0 F6 >59      BNE K_WAIT2
>60      *lees alle data uit de klok
8E27: A0 0D >61      LDY #$0C+1 ;maximaal KLOK commando nummer
>62      *
8E29: 88 >63      VOLG_COM DEY ;verlaag commandonummer
8E2A: 30 22 >64      BMI GELEZEN ;klaar
8E2C: A9 86 >65      LDA #<PLATAB ;bereid INDIRECT voor op
8E2E: 85 3A >66      STA INDIRECL ;inlezen tabelwaarde
8E30: B1 3A >67      LDA (INDIRECT),Y ;plaats in RAM data
8E32: AA >68        TAX ;zet in XREG
8E33: A9 93 >69      LDA #<STRIPTAB ;bereid INDIRECT voor op
8E35: 85 3A >70      STA INDIRECL ;afstrippen bitten
8E37: 98 >71        TYA ;haal gevraagde klokcommando
8E38: 09 50 >72      ORA #HOLD_RD ;voeg juiste status bitten toe
8E3A: 8D FE CF >73    STA KREG ;zet in klokcommandoregister
8E3D: 8D FE CF >74    STA KREG ;keertje extra voor timing
8E40: 68 >75        PLA ;wacht 6 micro sec
8E41: 48 >76        PHA
8E42: AD FD CF >77    LDA DREG ;pak klokdata
8E45: 31 3A >78      AND (INDIRECT),Y ;strip ongewenste bitten af
8E47: 09 B0 >79      ORA #M_ASCII ;maak er ASCII van
8E49: 9D 10 CF >80    STA KLOKDATA,X ;sla op in RAM
8E4C: D0 DB >81      BNE VOLG_COM ;altijd
>82      *klok ingelezen voer de extra leestekens toe
8E4E: A9 BB >83      GELEZEN LDA #";"
8E50: 8D 12 CF >84    STA KLOKDATA+$2 ;tussen UU, MM en SS
8E53: 8D 15 CF >85    STA KLOKDATA+$5
8E56: A9 A0 >86      LDA #SPATIE
8E58: 8D 18 CF >87    STA KLOKDATA+$8 ;en er nog wat spaties in
8E5B: 8D 1F CF >88    STA KLOKDATA+$0F
8E5E: A9 8D >89      LDA #CR
8E60: 8D 21 CF >90    STA KLOKDATA+$11
>91      *
8E63: A9 0F >92      K_TERUG LDA #%00001111 ;na HOLD niet meteen interrupten
8E65: 8D FE CF >93    STA KREG
8E68: 68 >94        PLA ;zet status in originele staat
8E69: 09 8F >95      ORA #%10001111 ;nu de interrupt erbij
8E6B: 8D FE CF >96    STA KREG
8E6E: A9 00 >97      LDA #$0 ;zet aantal verzonden karakters
8E70: 8D 05 CF >98    STA TDATATEL
>99      *
>100     *keer terug naar aanroeper
>101     *
8E73: 68 >102      PLA ;haal XREG terug
8E74: AA >103      TAX
8E75: A0 FF >104      LDY #CREGL
8E77: AD 78 07 >105    LDA SCRAT1 ;haal bewaarde CREG op
8E7A: 91 3A >106      STA (INDIRECT),Y ;klok uit
8E7C: AD 00 CF >107    LDA RETADR ;pak returnadres laag
8E7F: 85 3A >108      STA INDIRECL ;zet in INDIRECT
8E81: AD 02 CF >109    LDA RETBANK ;terugkeer banknr erin
8E84: D0 6F >110     BNE SCHAKELLE ;altijd

```


VIERLINGKAART BESTURINGSSOFTWARE

```

>111 *
>112 *tabel PLATAB geeft de plaats waar
>113 *een bepaald dataitem gestoord wordt in RAM (KLOKDATA)
8E86: 07 >114 PLATAB DFB $7 ;=sec eenheid
8E87: 06 >115 DFB $6 ;=sec tiental
8E88: 04 >116 DFB $4 ;=min eenheid
8E89: 03 >117 DFB $3 ;=min tiental
8E8A: 01 >118 DFB $1 ;=uur eenheid
8E8B: 00 >119 DFB $0 ;=uur tiental
8E8C: 10 >120 DFB $10 ;=dag van de week
8E8D: 0E >121 DFB $0E ;=dag eenheid
8E8E: 0D >122 DFB $0D ;=dag tiental
8E8F: 0C >123 DFB $0C ;=maand eenheid
8E90: 0B >124 DFB $0B ;=maand tiental
8E91: 0A >125 DFB $0A ;=jaar eenheid
8E92: 09 >126 DFB $9 ;=jaar tiental
>127 *tabel STRIPTAB geeft aan welke bits
>128 *bij dit opdrachtnummer afgestript
>129 *moeten worden
8E93: 0F >130 STRIPTAB DFB $0F
8E94: 07 >131 DFB $07
8E95: 0F >132 DFB $0F
8E96: 07 >133 DFB $07
8E97: 0F >134 DFB $0F
8E98: 03 >135 DFB $03
8E99: 07 >136 DFB $07
8E9A: 0F >137 DFB $0F
8E9B: 03 >138 DFB $03
8E9C: 0F >139 DFB $0F
8E9D: 01 >140 DFB $01
8E9E: 0F >141 DFB $0F
8E9F: 0F >142 DFB $0F
>143 *
>144 *opvullen tot bankswitch op zijn plaatst
8EA0: FF FF FF >145 HEX FFFFFFFF
>146 *

```

VIERLINGKAART BESTURINGSSOFTWARE

```

>148 *
>149 *
>150 *****
>151 * *
>152 * K L O K   I N V U L L I N G *
>153 * *
>154 *****
>155 *
>156 *
>157 *De complete tijd aanduiding moet in RAM (KLOKDATA) staan
>158 *in de vorm (in ASCII ):
>159 *UU;MM;SS JJMMDD W<CR>
>160 *de seconden worden ook geschreven
>161 *maar hebben geen zin, wordt door de
>162 *klok niet gezet
>163 *
8EA5: A9 00 >164 K_SCHRYF LDA  #$0
8EA7: 85 3A >165          STA  INDIRECL ;herstel INDIRECT
8EA9: A0 FF >166          LDY  #CREGL  ;bewaar CREG
8EAB: B1 3A >167          LDA  (INDIRECT),Y
8EAD: 8D 78 07 >168         STA  SCRAT1  ;SAVCREG is al ingebruik
8EB0: 8A      >169         TXA          ;bewaar XREG
8EB1: 48      >170         PHA          ;even op stack
8EB2: A9 8E   >171         LDA  #KLOK_AAN ;selekteer de klok
8EB4: 91 3A   >172         STA  (INDIRECT),Y
8EB6: AD FD CF >173         LDA  DREG    ;bewaar status informatie
8EB9: 48      >174         PHA
8EBA: A9 1F   >175         LDA  #HOLD    ;HOLD zetten
8EBC: 8D FE CF >176         STA  KREG    ;zet in klok
8EBF: A9 09   >177         LDA  #$9     ;wacht >150 muSEC (=175 muSEC)
8EC1: 20 A8 FC >178         JSR  WAIT
>179 *
>180 *schrijf alle data in de klok
8EC4: A0 0D   >181         LDY  #$0C+1 ;maximaal KLOK commando nummer
>182 *
8EC6: 88      >183 VOLG_SCR DEY          ;verlaag commandonummer
8EC7: 30 9A   >184         BMI  K_TERUG ;klaar
8EC9: A9 86   >185         LDA  #<PLATAB ;bereid INDIRECT voor op
8ECB: 85 3A   >186         STA  INDIRECL ;inlezen tabelwaarde
8ECD: B1 3A   >187         LDA  (INDIRECT),Y ;plaats in RAM data
8ECF: AA      >188         TAX          ;zet in XREG
8ED0: A9 93   >189         LDA  #<STRIPTAB ;bereid INDIRECT voor op
8ED2: 85 3A   >190         STA  INDIRECL ;afstrippen bitten
8ED4: 98      >191         TYA          ;haal gevraagde klokcommando
8ED5: 09 10   >192         ORA  #HOLD_WR ;voeg juiste status bitten toe
8ED7: 8D FE CF >193         STA  KREG    ;zet in klokcommandoregister
8EDA: 8D FE CF >194         STA  KREG    ;voor timing
8EDD: BD 10 CF >195         LDA  KLOKDATA,X ;pak data uit RAM
8EE0: 31 3A   >196         AND  (INDIRECT),Y ;strip ongewenste bitten af
8EE2: C0 05   >197         CPY  #$5     ;is het tiental uren
8EE4: D0 02   >198         BNE  KSNORMAL ;nee
8EE6: 09 08   >199         ORA  #%00001000 ;ja zet 24 uur mode
8EE8: 09 10   >200 KSNORMAL ORA  #HOLD_WR ;zet juiste statusbitten weer
8EEA: 8D FD CF >201         STA  DREG    ;zet in klok
8EED: 8D FD CF >202         STA  DREG    ;voor timing

```

VIERLINGKAART BESTURINGSSOFTWARE

```

8EF0: 8D FD CF >203          STA DREG          ;voor timing (ruim genoeg)
8EF3: D0 D1          >204          BNE VOLG_SCR     ;altijd
>205          *
>206          *****
>207          *
>208          *Schakel naar andere BANK      *
>209          *
>210          *****
>211          *
>212          *Schakel andere bank in door het
>213          *gewenste banknummer naar adresgebied
>214          *$Cn00-$CnFE te schrijven, banknummer
>215          *staat reeds in ACCU
>216          *In adres INDIRECT staat reeds waar naar
>217          *toe gesprongen wordt bij binnenkomst
>218          *andere bank
>219          *
>220          *Y-register wordt gebruikt
>221          *
8EF5: A0 00          >222          SCHAKELLELDY #0
8EF7: 09 80          >223          ORA #SOFTBANK ;Zet softbankselectie bit
8EF9: 91 3A          >224          STA (INDIRECT),Y ;Schakel om
8EFB: 6C 3A 00      >225          JMP (INDIRECT) ;Spring weg bij binnenkomst
>226          *
8EFE: 1E            >227          DFB BNKLOK1 ;Nummer van deze BANK
8EFF: FF            >228          DFB $FF ;Adres Controle Register
>229          *

```

VIERLINGKAART BESTURINGSSOFTWARE

```

>1 *****
>2 *****
>3 ***                               ***
>4 ***   B A N K   N R   1 F   ***
>5 ***                               ***
>6 *****
>7 *****
>8 *                                 *
>9 *bank BNKKLOKP VERSIE 5.0/850607*
>10 *                                 *
>11 *****
>12 *
>13 *
>14 *vervolg van bank ENKKLOK en wel het PRODOS deel
>15 *PRODOS KLOKREGISTER INVULLING
>16 *
>17 *
>18 *Eigen EQU
>19 PRODATA =   $BF90           ;ProdOS klokregisters
>20 M_DECIM =   $0F             ;decimaal bitmask
>21 *
>22 *ga eerst de klok uitlezen
8F00: A9 00 >23 B_KLOKP LDA  #$0
8F02: 85 3A >24           STA  INDIRECL ;reset INDIRECT
8F04: AD 00 CF >25           LDA  RETADR  ;bewaars RETADR even
8F07: 48 >26           PHA
8F08: A9 20 >27           LDA  #<PRORTS
8F0A: 8D 00 CF >28           STA  RETADR  ;nieuwe erin
8F0D: AD 02 CF >29           LDA  RETBANK ;returnbank bewaren
8F10: 48 >30           PHA
8F11: A0 FE >31           LDY  #BANKNR
8F13: B1 3A >32           LDA  (INDIRECT),Y ;banknummer
8F15: 8D 02 CF >33           STA  RETBANK
8F18: A9 00 >34           LDA  #<B_KLOK1 ;bereid INDIRECT voor
8F1A: 85 3A >35           STA  INDIRECL
8F1C: A9 1E >36           LDA  #BNKKLOK1 ;volgende banknr erin
8F1E: D0 62 >37           BNE  TUSKLOKP ;altijd via tussen jumpje
>38 *****
>39 *terugkeer punt na klokinlezing
8F20: A9 00 >40 PRORTS LDA  #$0           ;reset INDIRECT
8F22: 85 3A >41           STA  INDIRECL
8F24: 68 >42           PLA           ;herstel RETBANK
8F25: 8D 02 CF >43           STA  RETBANK
8F28: 68 >44           PLA
8F29: 8D 00 CF >45           STA  RETADR  ;herstel RETADR
>46 *converteer ASCII naar decimaal getal
>47 *niet kijken wat er staat (GOOR)
8F2C: A2 11 >48           LDX  #KLOKLANG
8F2E: BD 10 CF >49 M_DECI LDA  KLOKDATA,X ;pak dataitem
8F31: 29 0F >50           AND  #M_DECIM ;ASCII eraf
8F33: 9D 10 CF >51           STA  KLOKDATA,X ;en weer bewaren
8F36: CA >52           DEX
8F37: 10 F5 >53           BPL  M_DECI
>54 *
>55 *ga de ProdOS klokregisters vullen

```

VIERLINGKAART BESTURINGSSOFTWARE

```

>56 *
8F39: 18 >57 CLC
8F3A: A9 00 >58 LDA #0
8F3C: AE 1D CF >59 LDX KLOKDATA+$D ;tiental DAG
8F3F: F0 05 >60 BEQ DAGKLAAR
8F41: 69 0A >61 DAGMEER ADC #10 ;verhoog met tien
8F43: CA >62 DEX
8F44: D0 FB >63 BNE DAGMEER ;gereed, nee
8F46: 6D 1E CF >64 DAGKLAAR ADC KLOKDATA+$E ;eenheid dag erbij
8F49: 8D 90 BF >65 STA PRODATA ;in klokregister
>66 *
>67 *MAAND bepalen en toevoegen
>68 *
8F4C: A9 00 >69 LDA #0
8F4E: AE 1B CF >70 LDX KLOKDATA+$B ;tiental maand
8F51: F0 02 >71 BEQ MNDKLAAR
8F53: 69 0A >72 ADC #10 ;tien erbij
8F55: 6D 1C CF >73 MNDKLAAR ADC KLOKDATA+$C ;eenheid maand erbij
8F58: 0A >74 ASL ;5 posities naar links
8F59: 0A >75 ASL ;=1 cijfer in CARRY
8F5A: 0A >76 ASL
8F5B: 0A >77 ASL
8F5C: 0A >78 ASL
8F5D: 0D 90 BF >79 ORA PRODATA ;dagen erbij
8F60: 8D 90 BF >80 STA PRODATA ;en tesamen bewaren
>81 *
>82 *jaren nu maar er staat nog CARRY van de maanden
>83 *NB de jaren worden 1 positie naar
>84 *links geshift berekend
>85 *
8F63: A9 00 >86 LDA #0
8F65: 69 00 >87 ADC #0
8F67: 18 >88 CLC ;gelijk maar 80 erbij
8F68: AE 19 CF >89 LDX KLOKDATA+$9 ;tiental jaren
8F6B: F0 05 >90 BEQ JRKLAAR
8F6D: 69 14 >91 JRMEER ADC #10*2 ;tien jaren erbij(1 pos geshift)
8F6F: CA >92 DEX
8F70: D0 FB >93 BNE JRMEER ;nogmeer
8F72: 8D 91 BF >94 JRKLAAR STA PRODATA+1
8F75: AD 1A CF >95 LDA KLOKDATA+$A ;eenheid jaren
8F78: 0A >96 ASL
8F79: 6D 91 BF >97 ADC PRODATA+1 ;bij vorige waarde
8F7C: 8D 91 BF >98 STA PRODATA+1 ;bewaren
>99 *
>100 *
8F7F: 18 >101 CLC
8F80: 90 02 >102 BCC MINBEP ;tussenjumpje
>103 *
8F82: D0 71 >104 TUSKLOKP BNE SCHAKELLF ;tussenjumpje
>105 *
>106 *minuten bepalen
>107 *
8F84: A9 00 >108 MINBEP LDA #0
8F86: AE 13 CF >109 LDX KLOKDATA+$3 ;tiental minuten
8F89: F0 05 >110 BEQ MINKLAAR ;geen tiental

```

VIERLINGKAART BESTURINGSSOFTWARE

```

8F8B: 69 0A >111 MINMEER ADC #10 ;tien erbij
8F8D: CA >112 DEX
8F8E: D0 FB >113 BNE MINMEER
8F90: 6D 14 CF >114 MINKLAAR ADC KLOKDATA+$4 ;eenheden minuten
8F93: 8D 92 BF >115 STA PRODATA+2 ;sla op
>116 *
>117 *uren bepalen
>118 *
8F96: A9 00 >119 LDA #$0
8F98: AE 10 CF >120 LDX KLOKDATA+$0 ;tiental uren
8F9B: F0 05 >121 BEQ UURKLAAR ;geen tiental
8F9D: 69 0A >122 UURMEER ADC #10 ;tien erbij
8F9F: CA >123 DEX
8FA0: D0 FB >124 BNE UURMEER
8FA2: 6D 11 CF >125 UURKLAAR ADC KLOKDATA+$1 ;eenheid uren
8FA5: 8D 93 BF >126 STA PRODATA+3 ;bewaars
>127 *
>128 *gereed nu weer terug naar aanroepende bank
8FA8: AD 00 CF >129 LDA RETADR
8FAB: 85 3A >130 STA INDIRECL
8FAD: AD 02 CF >131 LDA RETBANK ;volgend banknr erin
8FB0: D0 43 >132 BNE SCHAKELLF ;altijd
>133 *****
>134 *
>135 *
>136 *opvullen tot bankswitch op plaats
>137 *
8FB2: FF FF FF >138 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8FC2: FF FF FF >139 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8FD2: FF FF FF >140 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8FE2: FF FF FF >141 HEX FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8FF2: FF FF FF >142 HEX FFFFFFFF
>143 *
>144 *****
>145 * *
>146 *Schakel naar andere BANK *
>147 * *
>148 *****
>149 *
>150 *Schakel andere bank in door het
>151 *gewenste banknummer naar adresgebied
>152 *$Cn00-$CnFE te schrijven, banknummer
>153 *staat reeds in ACCU
>154 *In adres INDIRECT staat reeds waar naar
>155 *toe gesprongen wordt bij binnenkomst
>156 *andere bank
>157 *
>158 *Y-register wordt gebruikt
>159 *
8FF5: A0 00 >160 SCHAKELFLDY #$0
8FF7: 09 80 >161 ORA #SOFTBANK ;Zet softbankselectie bit
8FF9: 91 3A >162 STA (INDIRECT),Y ;Schakel om
8FFB: 6C 3A 00 >163 JMP (INDIRECT) ;Spring weg bij binnenkomst
>164 *
8FFE: 1F >165 DFB BNKKLOKP ;Nummer van deze BANK

```

VIERLINGKAART BESTURINGSSOFTWARE

8FFF: FF >166 DFB \$FF ;Adres Controle Register
 >167 *

--End assembly--

8192 bytes

Errors: 0

Klokhuis Vierlingkaart

Aanhangsel E: Beperkingen

=====

In dit aanhangsel wordt aangegeven, op welke punten het gebruik van de Vierlingkaart tot problemen kan leiden. Dat wil overigens niet zeggen dat iedereen daarmee te maken krijgt.

ACIA-snelheden

De zendsnelheid van de ACIA is altijd gelijk aan de ontvangstsnelheid. Dit heeft tot gevolg dat de Vierlingkaart niet gebruikt kan worden om van de Apple een Viditel-terminal te maken, althans niet met de huidige Besturingssoftware.

Omschakelen van functies

Het omschakelen van functies gebeurt door een waarde in het B-register te zetten. Dit is alleen mogelijk vanuit een programma of, indien BASIC actief is, vanaf het toetsenbord. Het is dus meestentijds niet mogelijk vanuit een kant-en-klaar programma zoals een tekstverwerker.

Instellen van een printerinterface

De beide printerinterfaces gaan ervan uit dat ze op de normale Applemanier worden gebruikt. Om dat te verduidelijken is het nodig, deze manier hier even uit te leggen.

Alle uitvoer uit een programma verloopt via het Monitor-programma. Als er bijv. een letter 'X' getoond moet worden, dan zet het programma dit in het A-register van de 6502-processor en roept een routine in de Monitor aan. Deze zet de letter dan gewoonlijk op het scherm. Is het daarentegen de bedoeling dat de letter op de printer komt, dan moet het programma dat eerst aan de Monitor

Klokhuis Vierlingkaart

kenbaar maken. In BASIC gaat dat met het PR#-commando. Het gevolg is dat de uitvoerroutine wordt omgeleid naar het besturingsprogramma in de aangegeven slot.

Nu is het zo dat een interfacekaart vaak hardware bevat die, voordat ze gebruikt kan worden, moet worden geïnitieerd. Dat is bijv. ook het geval met de VIA op de Vierlingkaart. Daarom zorgt het besturingsprogramma ervoor dat dit gebeurt zodra het eerste teken wordt uitgevoerd. Daarna wordt de uitvoeromleiding zo aangepast dat de verdere tekens zondermeer worden uitgevoerd.

Er blijken nu programma's te bestaan die zich niet aan deze werkwijze houden. Ze sturen hun uitvoer direct naar het besturingsprogramma, zonder de Monitor te verwittigen. Het gevolg is dat de hardware bij elk uitgevoerd teken opnieuw wordt geïnitieerd. Dit leidt in elk geval tot vertraging, maar soms zelfs tot verlies van uitvoer.

Een van de betrokken programma's is Appleworks.

Interne klok voor ProDOS

De Vierlingkaart is, zoals in hoofdstuk 6 beschreven, niet door ProDOS herkenbaar als klok. In plaats daarvan wordt er een routinetje mee-geleverd dat de Vierlingkaart opspoort en deze dan als klok aan ProDOS koppelt. Voor dat koppelen is een machinetaalprogrammaatje nodig. Bij een 'officiële' klok staat dat ergens in het ProDOS-programma zelf. Door een leemte in de ProDOS-documentatie (die door een gebrek aan kennis bij Apple Nederland niet gevuld blijkt te kunnen worden) is het niet mogelijk, dit zelfde voor de Vierlingklok te verwezenlijken. De koppelroutine is nu in pagina 3 gezet. Dit is echter een plaats waar veel BASIC-programma's kleine machinetaalroutines zetten. Dit kan dus problemen leveren. Om de invloed nog zoveel mogelijk te beperken, staat de routine zover mogelijk bovenin pagina 3, nl. ongeveer vanaf adres \$3B0.

Klokhuis Vierlingkaart

Aanhangsel F: Datasheets

=====

| | |
|-----------------------|---------|
| DATASHEET 6522 VIA | blad 2 |
| DATASHEET 6551 ACIA | blad 24 |
| DATASHEET 5832RS KLOK | blad 45 |



R6522 VERSATILE INTERFACE ADAPTER (VIA)

DESCRIPTION

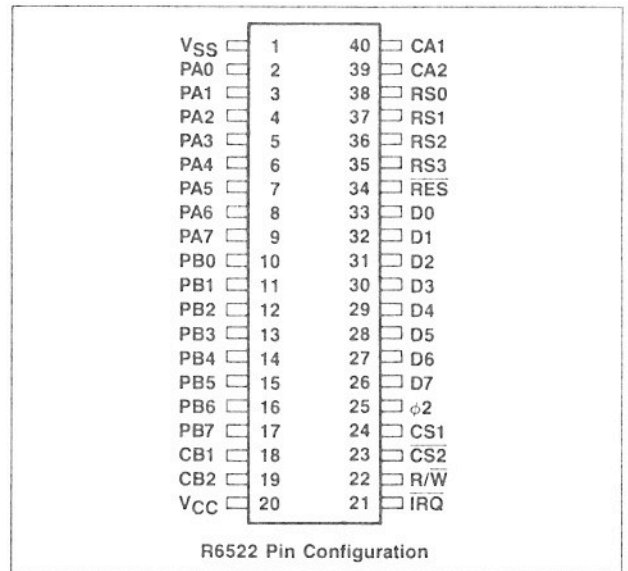
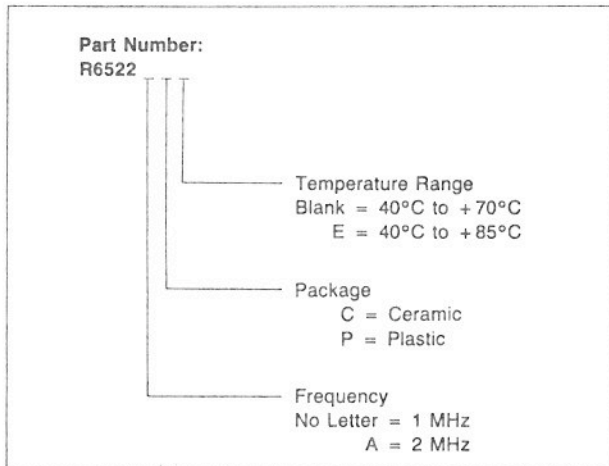
The R6522 Versatile Interface Adapter (VIA) is a very flexible I/O control device. In addition, this device contains a pair of very powerful 16-bit interval timers, a serial-to-parallel/parallel-to-serial shift register and input data latching on the peripheral ports. Expanded handshaking capability allows control of bidirectional data transfers between VIA's in multiple processor systems.

Control of peripheral devices is handled primarily through two 8-bit bidirectional 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.

FEATURES

- Two 8-bit bidirectional I/O ports
- Two 16-bit programmable timer/counters
- Serial data port
- 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
- Single +5V power supply

ORDERING INFORMATION



R6522

Versatile Interface Adapter (VIA)

INTERFACE SIGNALS

RESET (\overline{RES})

A low reset (\overline{RES}) input clears all R6522 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.

INPUT CLOCK (PHASE 2)

The input clock is the system $\phi 2$ clock and triggers all data transfers between processor bus and the R6522.

READ/WRITE (R/\overline{W})

The direction of the data transfers between the R6522 and the system processor is controlled by the R/\overline{W} line in conjunction with the CS1 and $\overline{CS2}$ inputs. When R/\overline{W} is low, (write operation) and the R6522 is selected, data is transferred from the processor bus into the selected R6522 register. When R/\overline{W} is high, (read operation) and the R6522 is selected, data is transferred from the selected R6522 register to the processor bus.

DATA BUS (D0-D7)

The eight bidirectional data bus lines transfer data between the R6522 and the system processor bus. During read cycles, the contents of the selected R6522 register are placed on the data bus lines. During write cycles, these lines are high-impedance inputs and data is transferred from the processor bus into the selected register. When the R6522 is not selected, the data bus lines are high-impedance.

CHIP SELECTS (CS1, $\overline{CS2}$)

The two chip select inputs are normally connected to processor address lines either directly or through decoding. The selected R6522 register is accessed when CS1 is high and $\overline{CS2}$ is low.

REGISTER SELECTS (RS0-RS3)

The coding of the four Register Select inputs select one of the 16 internal registers of the R6522, as shown in Table 1.

INTERRUPT REQUEST (\overline{IRQ})

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.

PERIPHERAL PORT A (PA0-PA7)

Port A consists of eight lines which can be individually programmed to act as inputs or outputs under control of Data Direction Register A. 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 2 illustrates the output circuit.

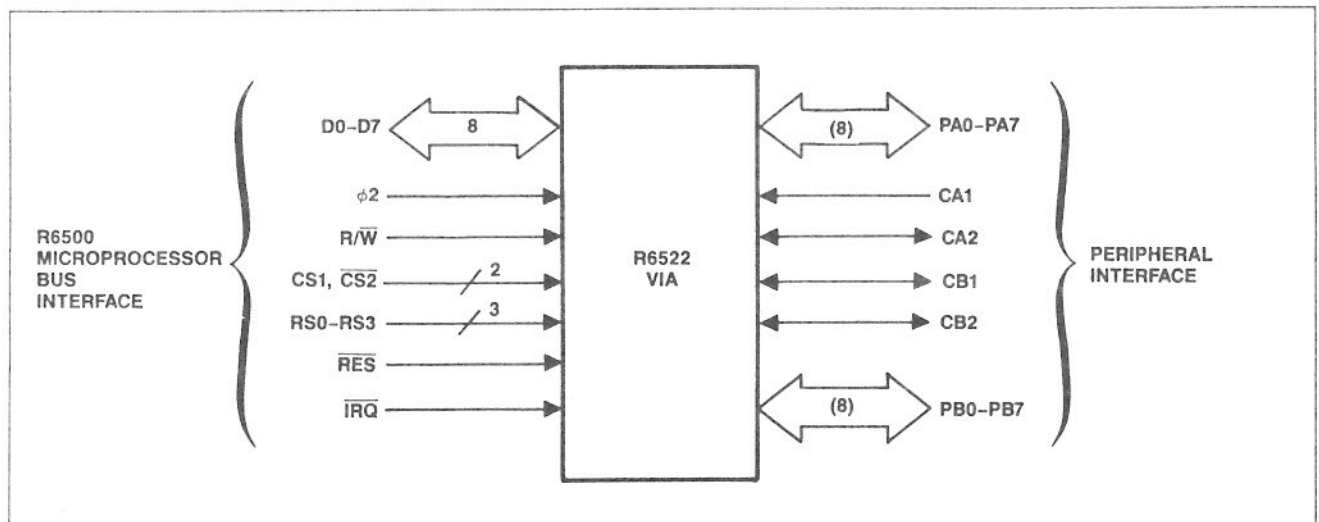


Figure 1. R6522 VIA Interface Signals

R6522

Versatile Interface Adapter (VIA)

PORT A CONTROL LINES (CA1, CA2)

The two Port 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 Port A 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.

PORT B (PB0-PB7)

Peripheral Port B consists of eight bidirectional lines which are controlled by an output register and a data direction register in much the same manner as the Port A. 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. Port 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 1.0 mA at 1.5 Vdc in the output mode to allow the outputs to directly drive Darlington transistor circuits. Figure 3 is the circuit schematic.

PORT B CONTROL LINES (CB1, CB2)

The Port 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. CB2 can also drive a Darlington transistor circuit; however, CB1 cannot.

Table 1. R6522 Register Addressing

| Register Number | RS Coding | | | | Register Desig. | Register/Description | |
|-----------------|-----------|-----|-----|-----|-----------------|-----------------------------|----------------------|
| | RS3 | RS2 | RS1 | RS0 | | Write (R/W = L) | Read (R/W = H) |
| 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 | Output Register B* | Input Register B* |

NOTE: *Same as Register 1 except no handshake.

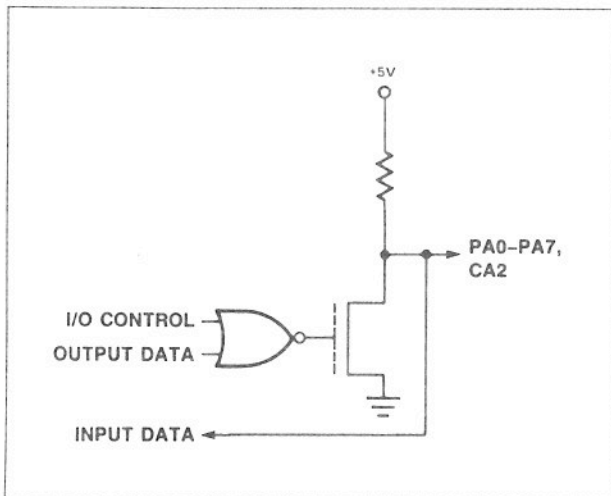


Figure 2. Port A Output Circuit

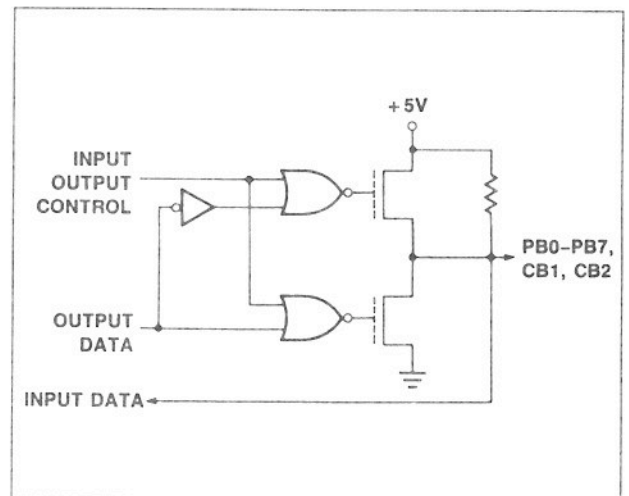


Figure 3. Port B Output Circuit

R6522

Versatile Interface Adapter (VIA)

FUNCTIONAL DESCRIPTION

The internal organization of the R6522 VIA is illustrated in Figure 4.

PORT A AND PORT B OPERATION

The R6522 VIA has two 8-bit bidirectional I/O ports (Port A and Port B) and each port has two associated control lines.

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 the Input Register (IRA, IRB). When the pin is programmed as an output, the voltage on the pin is controlled by the corresponding 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 5 through 8 illustrate the formats of the port registers. In addition, the input latching modes are selected by the Auxiliary Control Register (Figure 14).

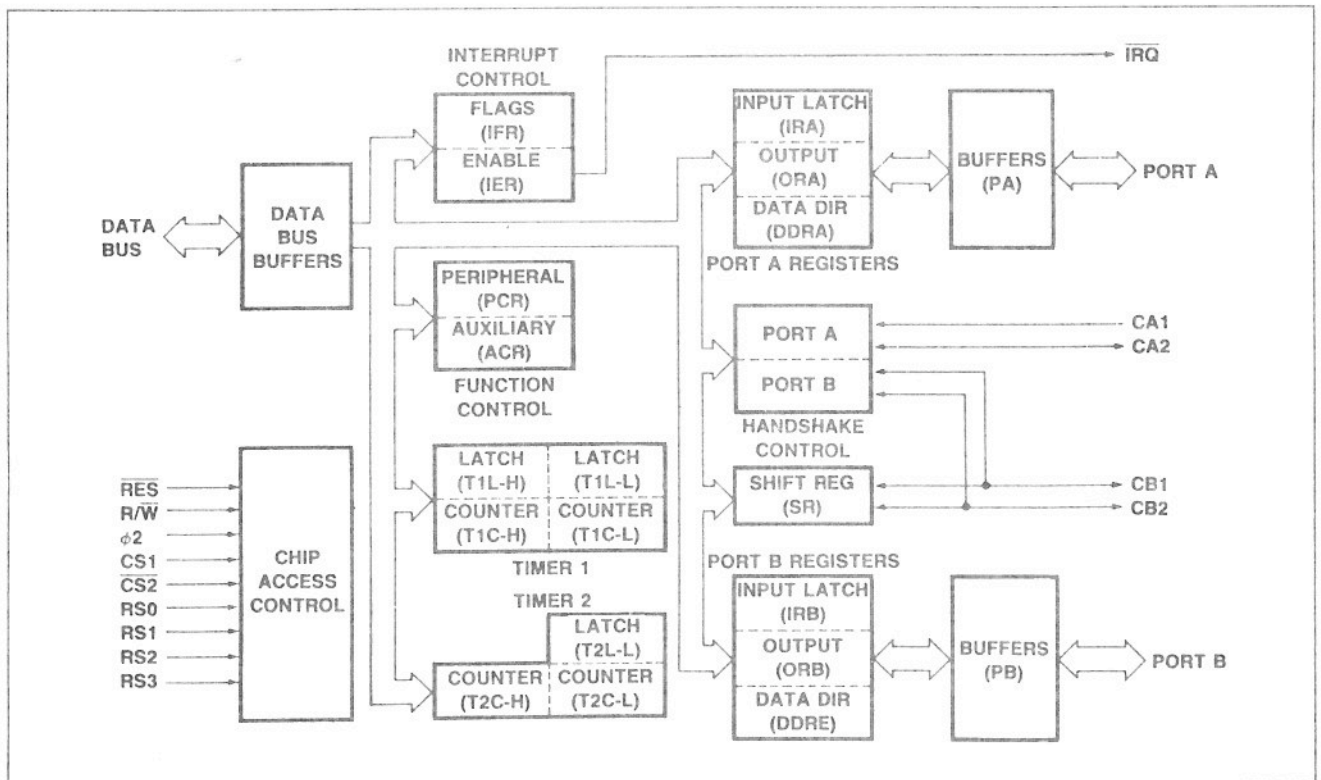


Figure 4. R6522 VIA Block Diagram

R6522

Versatile Interface Adapter (VIA)

HANDSHAKE CONTROL OF DATA TRANSFERS

The R6522 allows positive control of data transfers between the system processor and peripheral devices 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.

In the R6522, 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 9 which illustrates the normal Read Handshake sequence.

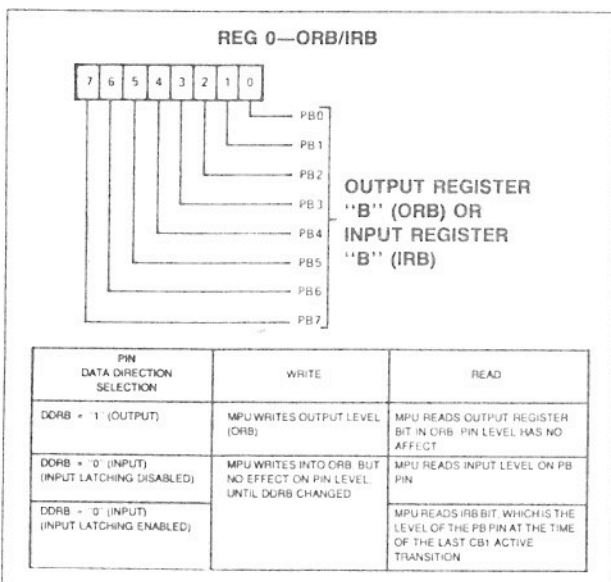


Figure 5. Output Register B (ORB), Input Register B (IRB)

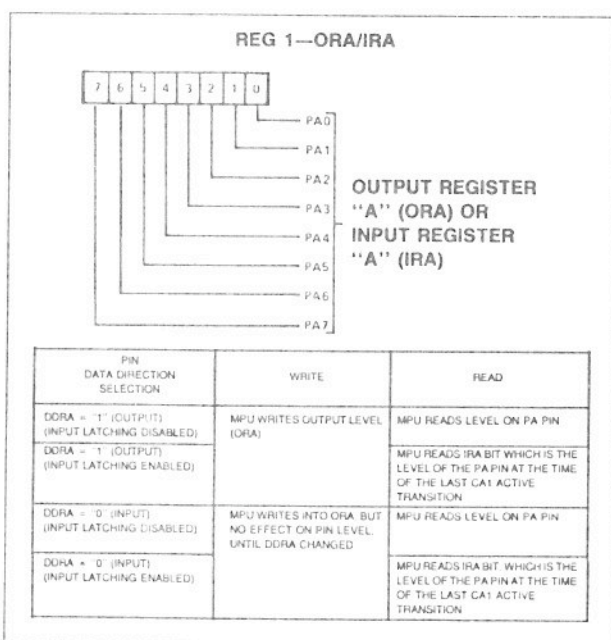


Figure 6. Output Register A (ORA), Input Register A (IRA)

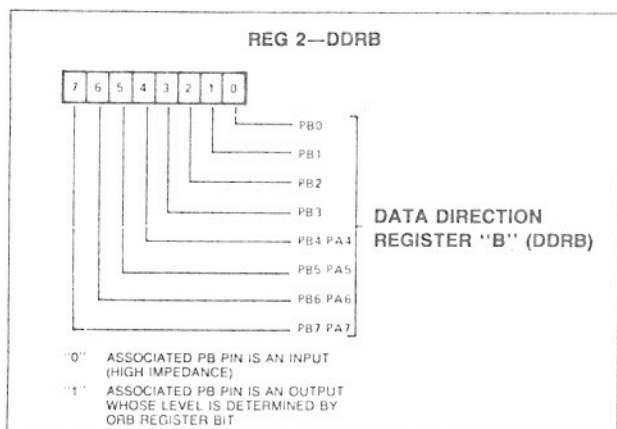


Figure 7. Data Direction Register B (DDRB)

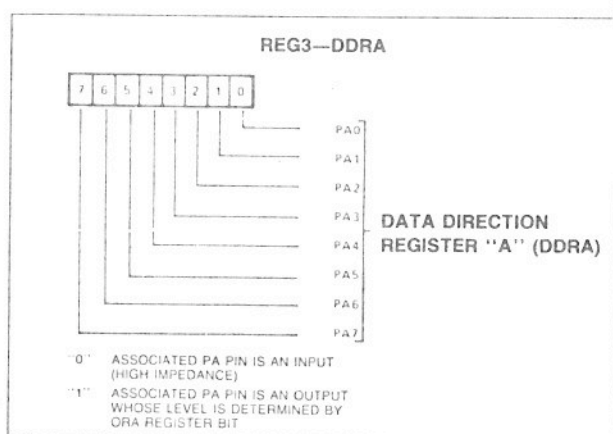


Figure 8. Data Direction Register A (DDRA)

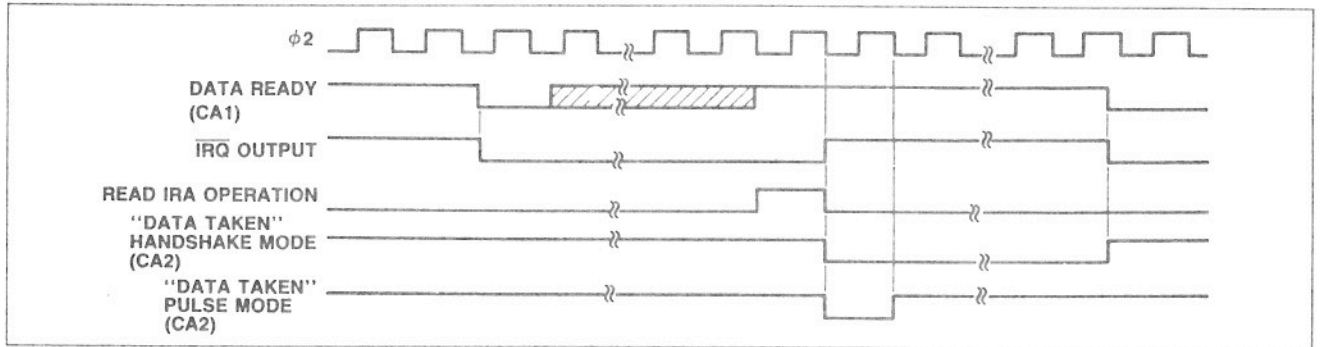


Figure 9. Read Handshake Timing (Port A, Only)

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 R6522 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 R6522. 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 clearing the "Data Ready" output. This sequence is shown in Figure 10.

Selection of operating modes for CA1, CA2, CB1, and CB2 is accomplished by the Peripheral Control Register (Figure 11).

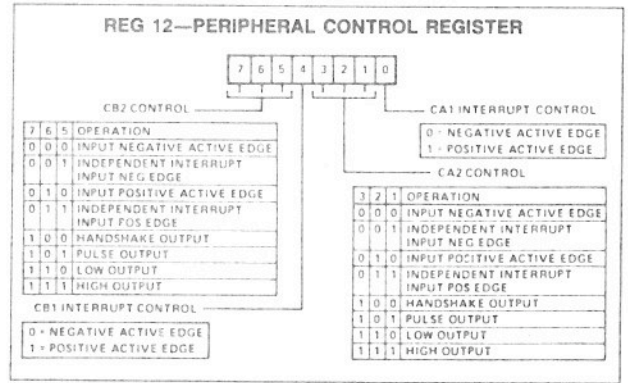


Figure 11. Peripheral Control Register (PCR)

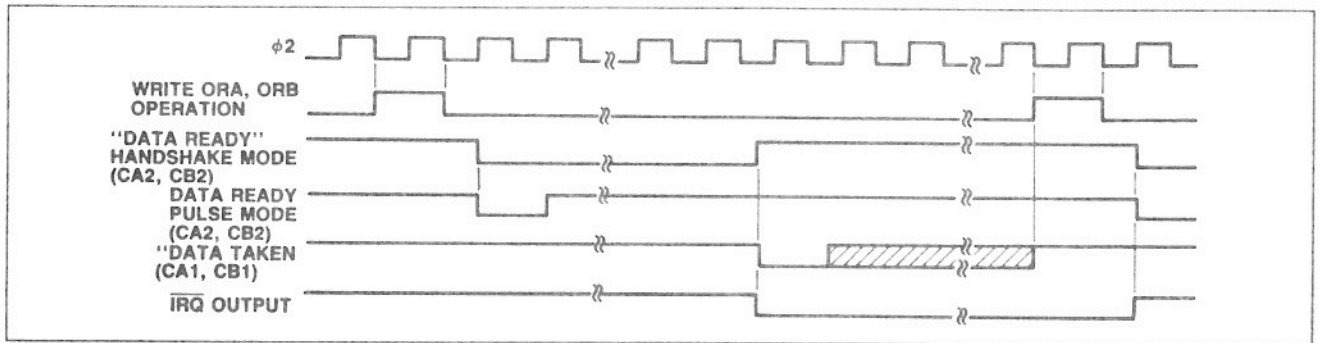


Figure 10. Write Handshake Timing

COUNTER/TIMERS

There are two independent 16-bit counter/timers (called Timer 1 and Timer 2) in the R6522. Each timer is controlled by writing bits into the Auxiliary Control Register (ACR) to select the mode of operation (Figure 14).

Timer 1 Operation

Interval Timer T1 consists of two 8-bit latches (Figure 12) and a 16-bit counter (Figure 13). The latches 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 is set, and IRQ goes low if the T1 interrupt is enabled. Timer 1 then disables any further interrupts, or automatically transfers the contents of

the latches into the counter and continues to decrement. In addition, the timer may be programmed to invert the output signal on a peripheral pin (PB7) each time it "times-out". Each of these modes is discussed separately below.

Note that 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 (T1L-L) when the processor writes into the high order counter (T1C-H). 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 latch.

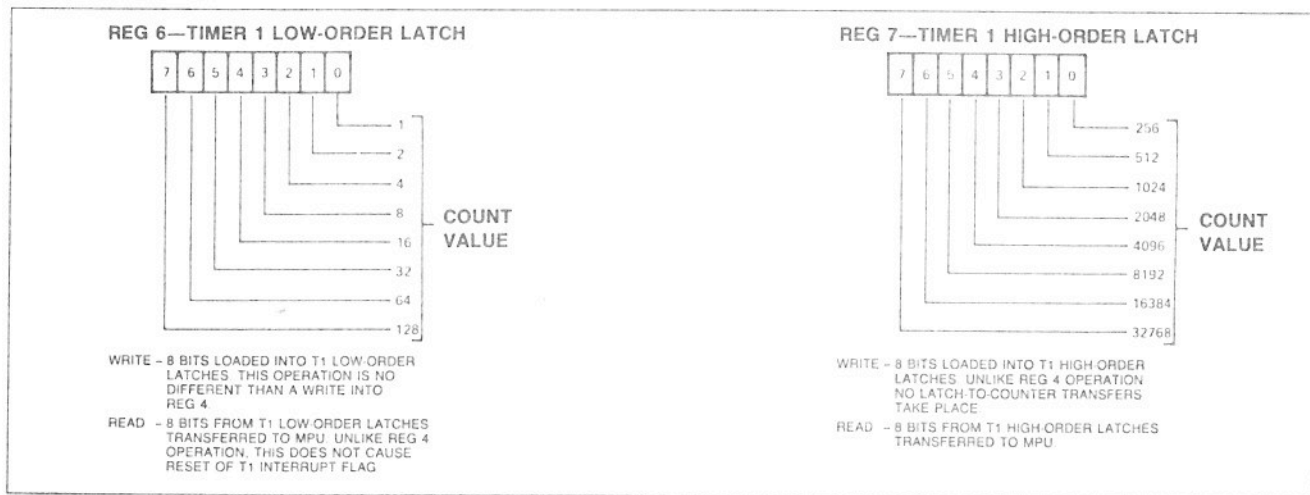


Figure 12. Timer 1 (T1) Latch Registers

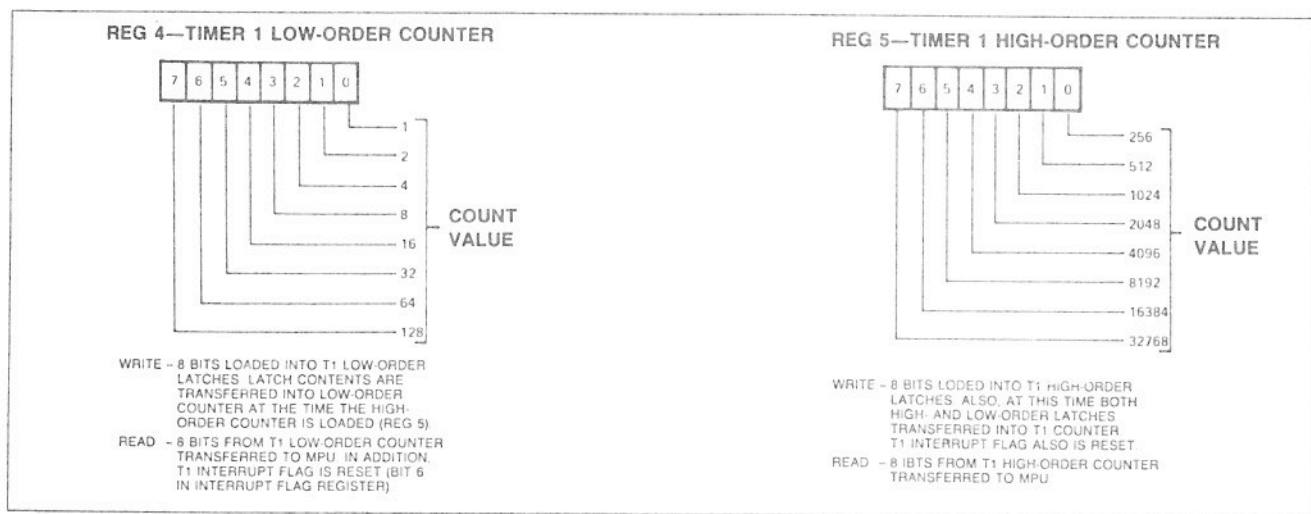


Figure 13. Timer 1 (T1) Counter Registers

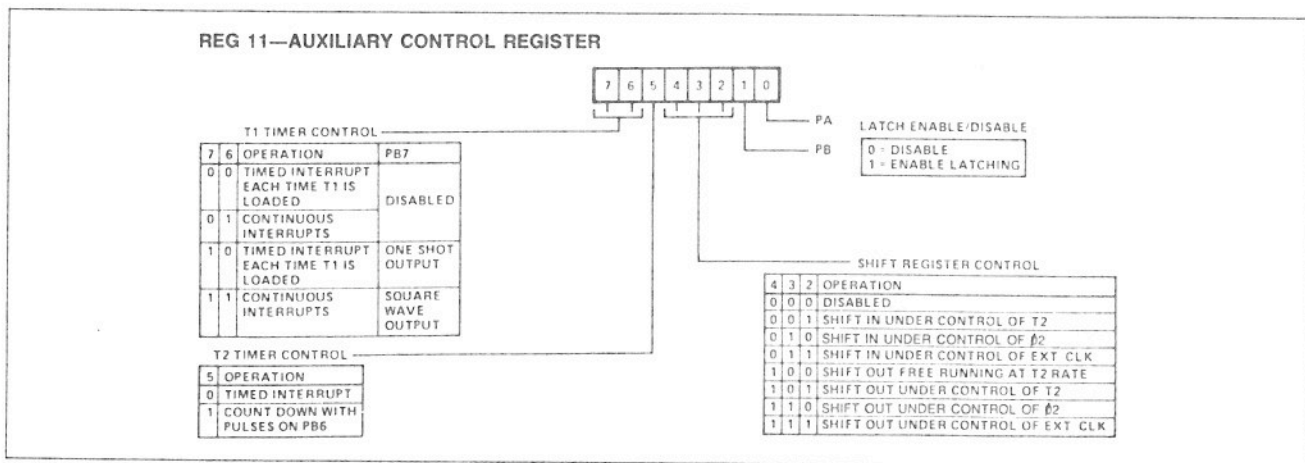


Figure 14. Auxiliary Control Register (ACR)

Timer 1 One-Shot Mode

The Timer 1 one-shot mode generates 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.

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 R6522 interval timer one-shot modes is shown in Figure 15.

In the one-shot mode, writing into the T1L-H 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 (T1C-H), 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 $\phi 2$ 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.

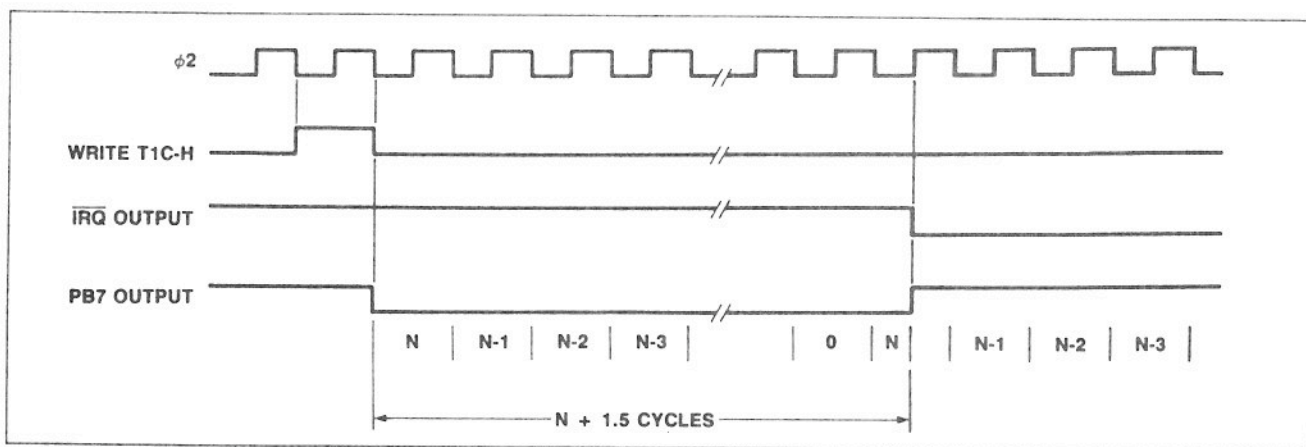


Figure 15. Timer 1 One-Shot Mode Timing

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 R6522 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.

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.

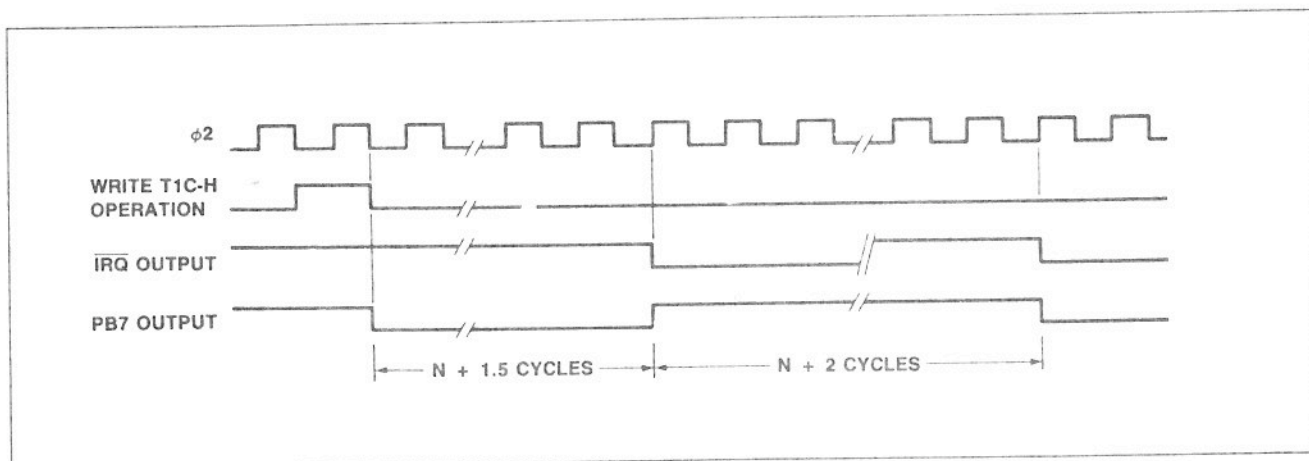


Figure 16. Timer 1 Free-Run Mode Timing

Timer 2 Operation

Timer 2 operates as an interval timer (in the "one-shot" mode only), or as a counter for counting negative pulses on the PB6 peripheral pin. A single control bit in the Auxiliary Control Register selects between these two modes. This timer is comprised of a "write-only" lower-order latch (T2L-L), a "read-only" low-order counter (T2C-L) and a read/write high order counter (T2C-H). The counter registers act as a 16-bit counter which decrements at $\phi 2$ rate. Figure 17 illustrates the T2 Latch/Counter Registers.

Timer 2 One-Shot Mode

As an interval timer, T2 operates in the "one-shot" mode similar to Time 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 is disabled after initial time-out so that it will not be set by the counter

decrementing again 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.

Timer 2 Pulse Counting Mode

In the pulse counting mode, T2 counts 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 is set when T2 counts down past zero. The counter will then continue to decrement with each pulse on PB6. However, it is necessary to rewrite T2C-H to allow the interrupt flag to set on a subsequent time-out. Timing for this mode is shown in Figure 19. The pulse must be low on the leading edge of $\phi 2$.

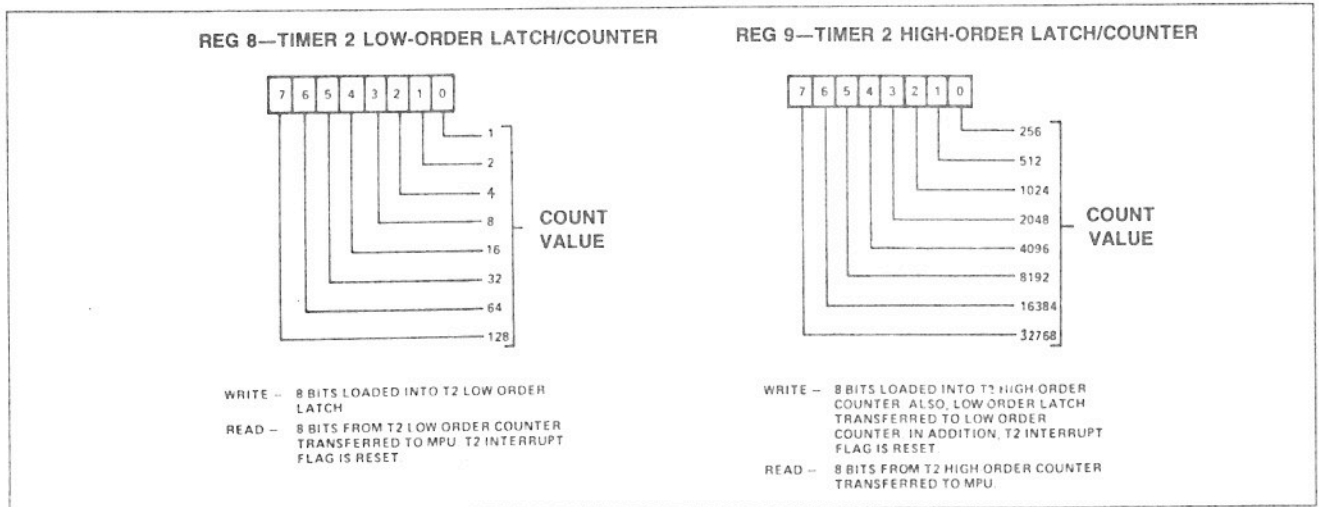


Figure 17. Timer 2 (T2) Latch/Counter Registers

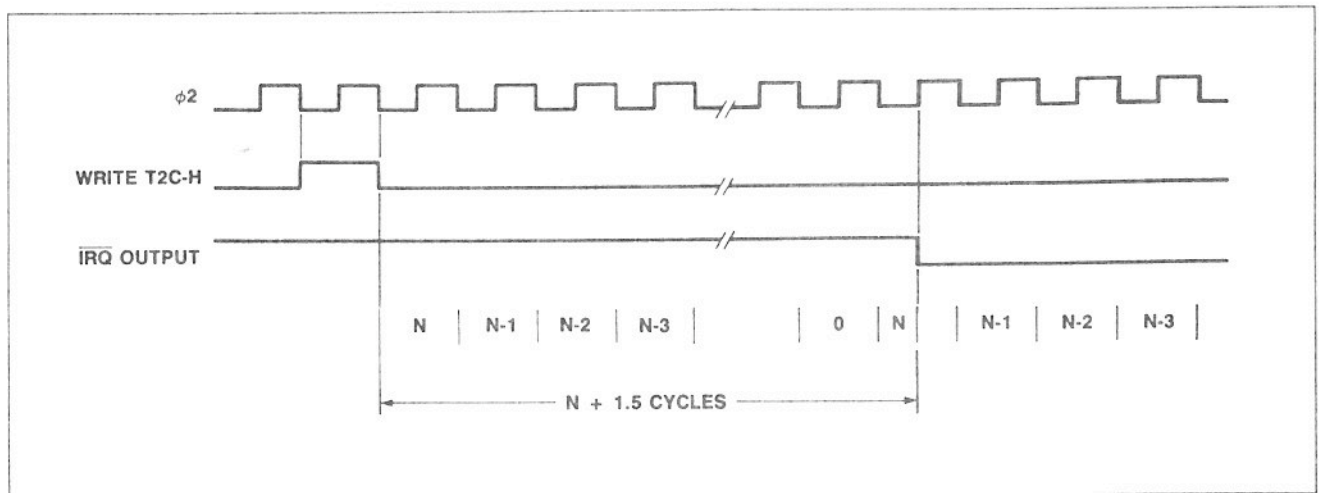


Figure 18. Timer 2 One-Shot Mode Timing

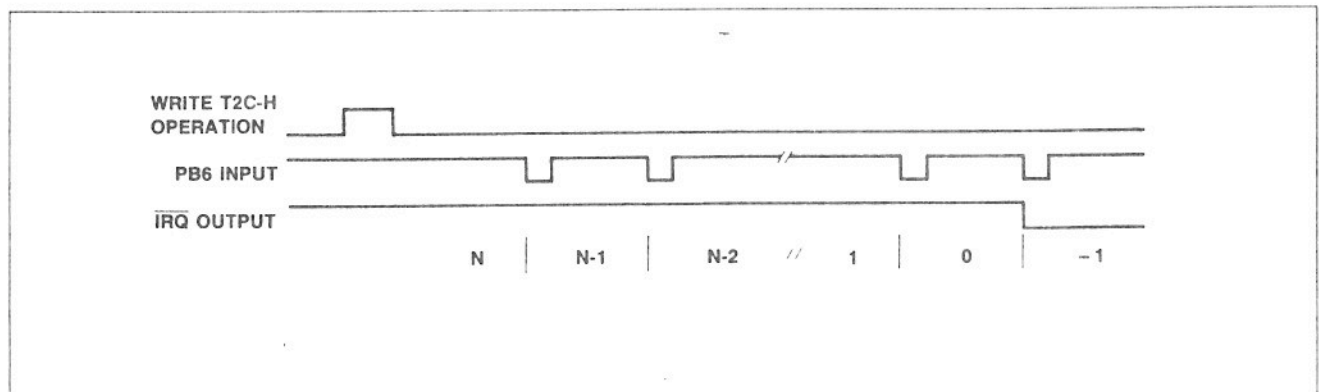


Figure 19. Timer 2 Pulse Counting Mode

R6522

Versatile Interface Adapter (VIA)

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 20 illustrates the configuration of the SR data bits and Figure 21 shows the SR control bits of the ACR.

SR Mode 0 — Disabled

Mode 0 disables the Shift Register. In this mode the microprocessor can write or read the SR and the SR will shift on each CB1 positive edge shifting in the value on CB2. In this mode the SR interrupt flag is disabled (held to a logic 0).

SR Mode 1 — Shift In Under Control of T2

In mode 1, the shifting rate is controlled by the low order 8 bits of T2 (Figure 22). 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 the read or write of the SR if the SR flag is set in the IFR. Otherwise the first shift will occur at the next time-out of T2 after a read or write of the SR. 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 set and \overline{IRQ} will go low.

SR Mode 2 — Shift In Under ϕ_2 Control

In mode 2, the shift rate is a direct function of the system clock frequency (Figure 23). 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.

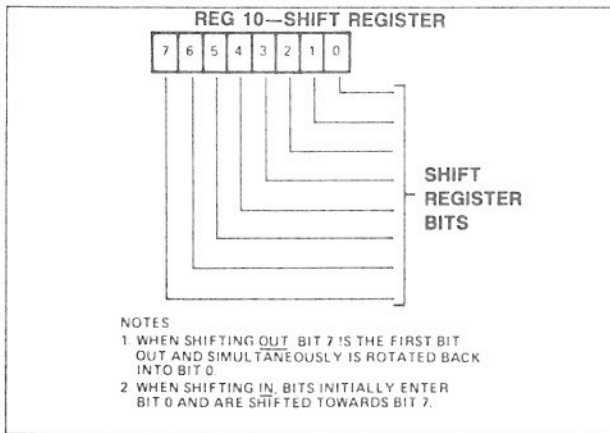


Figure 20. Shift Registers

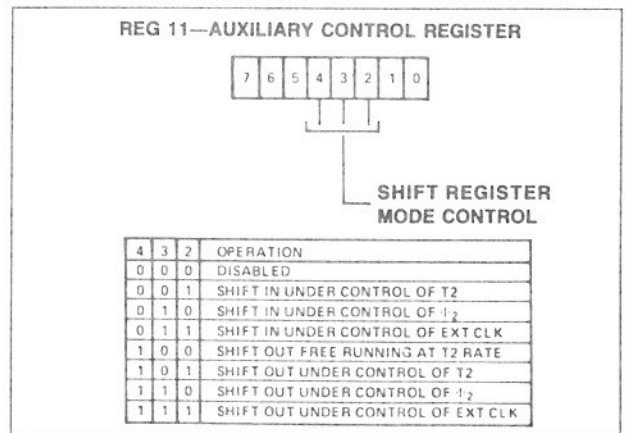


Figure 21. Shift Register Modes

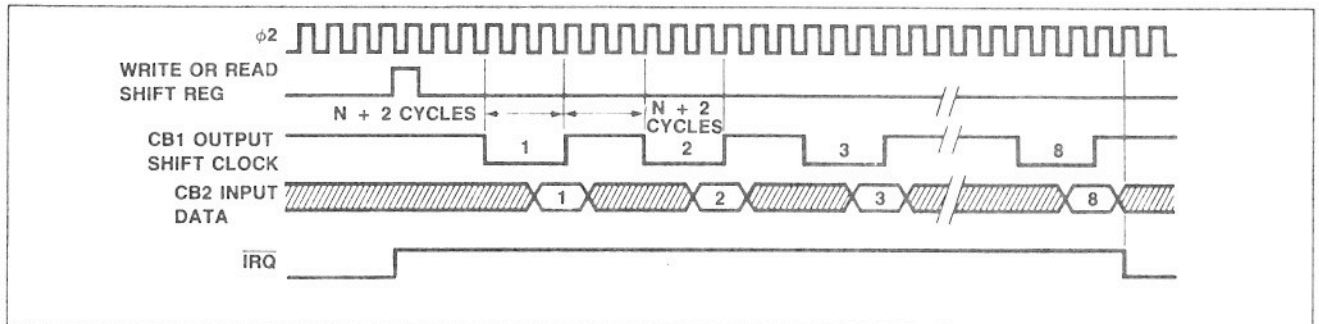


Figure 22. SR Mode 1 — Shift In Under T2 Control

SR Mode 3 — Shift In Under CB1 Control

In mode 3, external pin CB1 becomes an input (Figure 24). 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.

SR Mode 4 — Shift Out Under T2 Control (Free-Run)

Mode 4 is very similar to mode 5 in which the shifting rate is set by

T2. However, in mode 4 the SR Counter does not stop the shifting operation (Figure 25). 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.

SR Mode 5 — Shift Out Under T2 Control

In mode 5, the shift rate is controlled by T2 (as in mode 4). The shifting operation is triggered by the read or write of the SR if the SR flag is set in the IFR (Figure 26). Otherwise the first shift will occur at the next time-out of T2 after a read or write of the SR. 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.

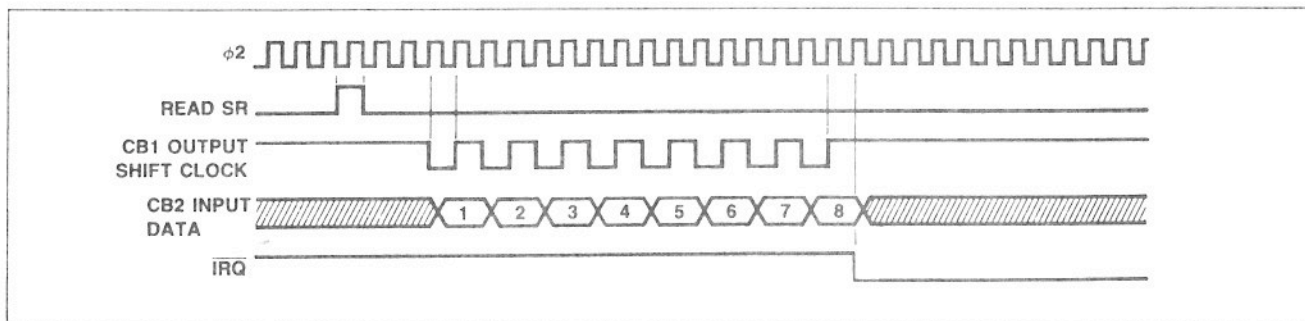


Figure 23. SR Mode 2 — Shift In Center $\phi 2$ Control

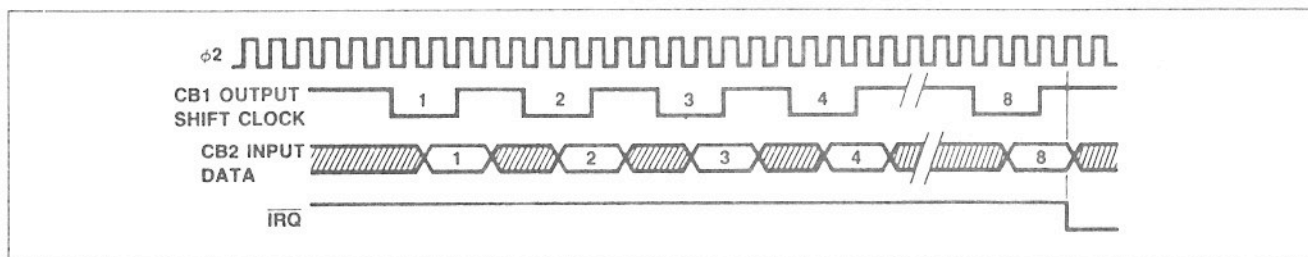


Figure 24. SR Mode 3 — Shift In Under CB1 Control

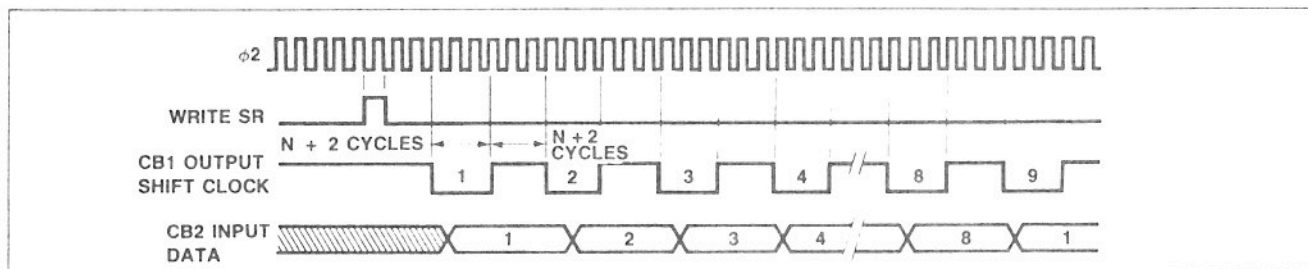


Figure 25. SR Mode 4 — Shift Our Under T2 Control (Free-Run)

SR Mode 6 — Shift Out Under $\phi 2$ Control

In mode 6, the shift rate is controlled by the $\phi 2$ system clock (Figure 27).

SR Mode 7 — Shift Out Under CB1 Control

In mode 7, shifting is controlled by pulses applied to the CB1 pin by an external device (Figure 28). 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.

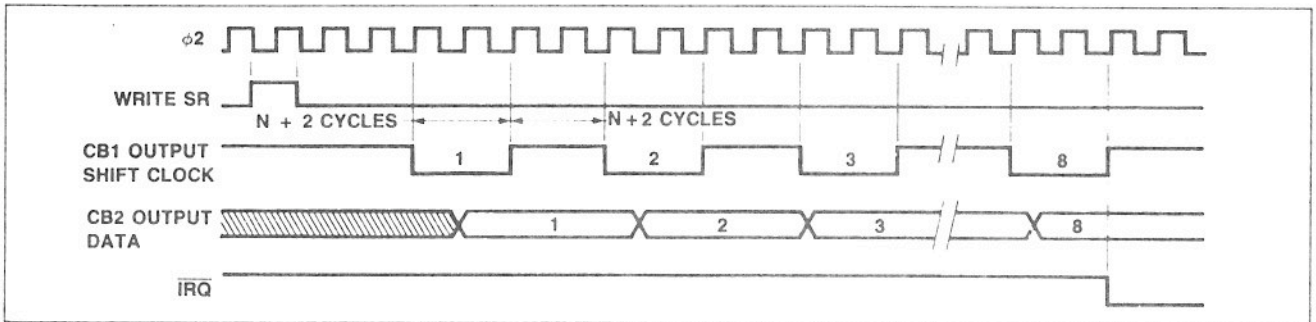


Figure 26. SR Mode 5 — Shift Out Under T2 Control

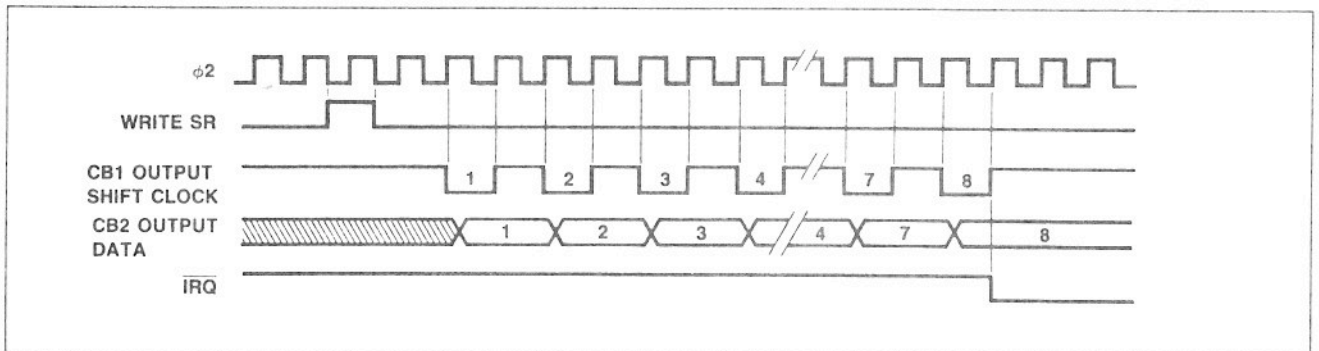


Figure 27. SR Mode 6 — Shift Out Under $\phi 2$ Control

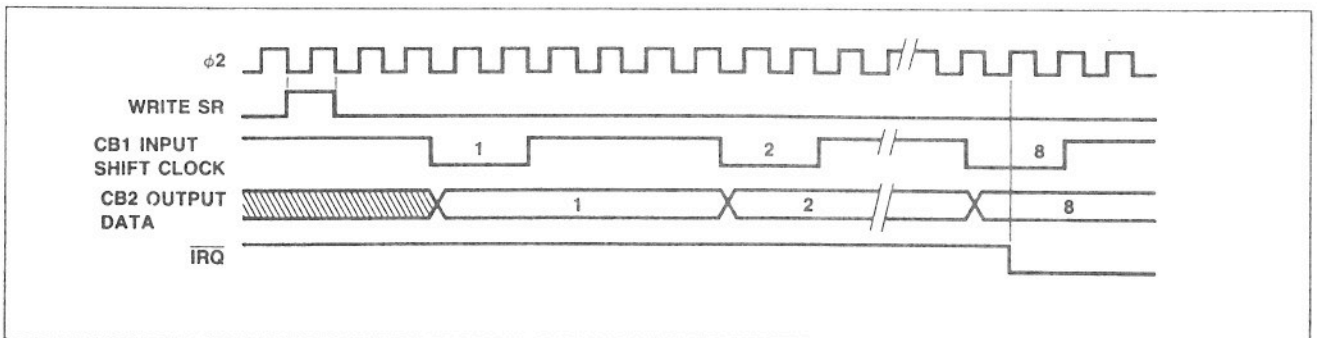


Figure 28. SR Mode 7 — Shift Out Under CB1 Control

R6522

Versatile Interface Adapter (VIA)

Interrupt Operation

Controlling interrupts within the R6522 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 in the Interrupt Flag Register (IFR) by conditions detected within the R6522 or on inputs to the R6522. 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.

Associated with each interrupt flag is an interrupt enable bit in the Interrupt Enable Register (IER). 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 "wired-OR'ed" with other devices in the system to interrupt the processor.

Interrupt Flag Register (IFR)

In the R6522, all the interrupt flags are contained in one register, i.e., the IFR (Figure 29). 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.

The Interrupt Flag Register (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 \overline{IRQ} output. This bit corresponds to the logic function: $\overline{IRQ} = IFR6 \times IER6 + IFR5 \times IER5 + IFR4 \times IER4 + IFR3 \times IER3 + IFR2 \times IER2 + IFR1 \times IER1 + IFR0 \times IER0$.

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.

Interrupt Enable Register (IER)

For each interrupt flag in IFR, there is a corresponding bit in the Interrupt Enable Register (IER) (Figure 30). Individual bits in the IER can be set or cleared to facilitate controlling individual interrupts without affecting others. This is accomplished by writing to the (IER) after bit 7 set or cleared to, in turn, set or clear selected enable bits. 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.

Selected bits in the IER can be set by writing to the IER with bit 7 in the data word set to a 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 contents of this register can be read at any time. Bit 7 will be read as a logic 1, however.

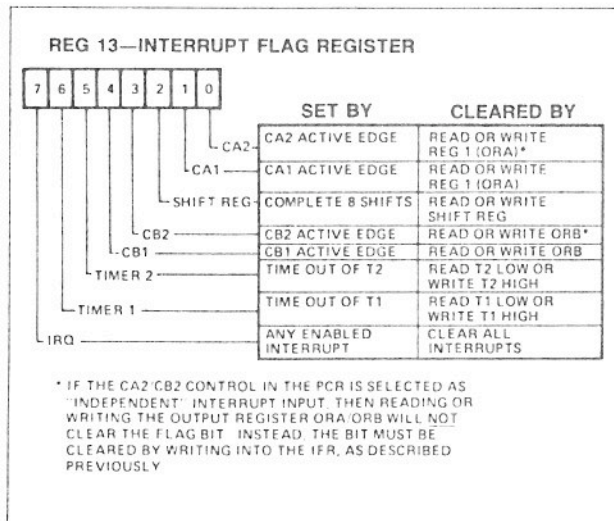


Figure 29. Interrupt Flag Register (IFR)

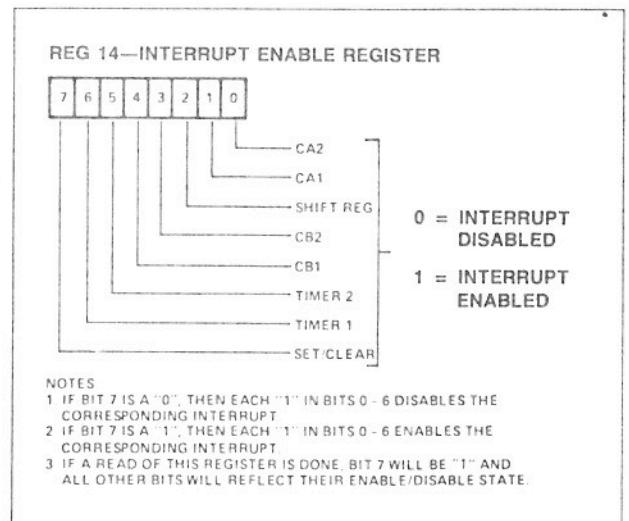


Figure 30. Interrupt Enable Register (IER)

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 | μ S | — |
| t_{CA2} | Delay Time, Clock Negative Transition to CA2 Negative Transition (read handshake or pulse mode) | — | 1.0 | μ S | 31a, 31b |
| t_{RS1} | Delay Time, Clock Negative Transition to CA2 Positive Transition (pulse mode) | — | 1.0 | μ S | 31a |
| t_{RS2} | Delay Time, CA1 Active Transition to CA2 Positive Transition (handshake mode) | — | 2.0 | μ S | 31b |
| t_{WHS} | Delay Time, Clock Positive Transition to CA2 or CB2 Negative Transition (write handshake) | 0.05 | 1.0 | μ S | 31c, 31d |
| t_{DS} | Delay Time, Peripheral Data Valid to CB2 Negative Transition | 0.20 | 1.5 | μ S | 31c, 31d |
| t_{RS3} | Delay Time, Clock Positive Transition to CA2 or CB2 Positive Transition (pulse mode) | — | 1.0 | μ S | 31c |
| t_{RS4} | Delay Time, CA1 or CB1 Active Transition to CA2 or CB2 Positive Transition (handshake mode) | — | 2.0 | μ S | 31d |
| t_{21} | Delay Time Required from CA2 Output to CA1 Active Transition (handshake mode) | 400 | — | ns | 31d |
| t_{IL} | Setup Time, Peripheral Data Valid to CA1 or CB1 Active Transition (input latching) | 300 | — | ns | 31e |
| t_{AL} | CA1, CB1 Setup Prior to Transition to Arm Latch | 300 | — | ns | 31e |
| t_{PDH} | Peripheral Data Hold After CA1, CB1 Transition | 150 | — | ns | 31e |
| t_{SR1} | Shift-Out Delay Time — Time from ϕ_2 Falling Edge to CB2 Data Out | — | 300 | ns | 31f |
| t_{SR2} | Shift-In Setup Time — Time from CB2 Data In to ϕ_2 Rising Edge | 300 | — | ns | 31g |
| t_{SR3} | External Shift Clock (CB1) Setup Time Relative to ϕ_2 Trailing Edge | 100 | T_{CY} | ns | 31g |
| t_{IPW} | Pulse Width — PB6 Input Pulse | $2 \times T_{CY}$ | — | | 31i |
| t_{ICW} | Pulse Width — CB1 Input Clock | $2 \times T_{CY}$ | — | | 31h |
| t_{IPS} | Pulse Spacing — PB6 Input Pulse | $2 \times T_{CY}$ | — | | 31i |
| t_{ICS} | Pulse Spacing — CB1 Input Pulse | $2 \times T_{CY}$ | — | | 31h |

PERIPHERAL INTERFACE WAVEFORMS

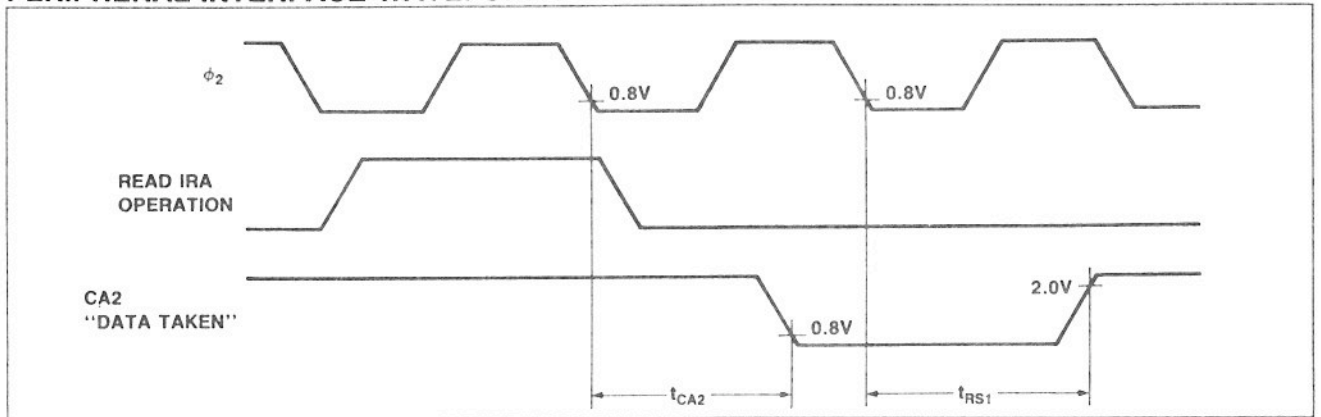


Figure 31a. CA2 Timing for Read Handshake, Pulse Mode

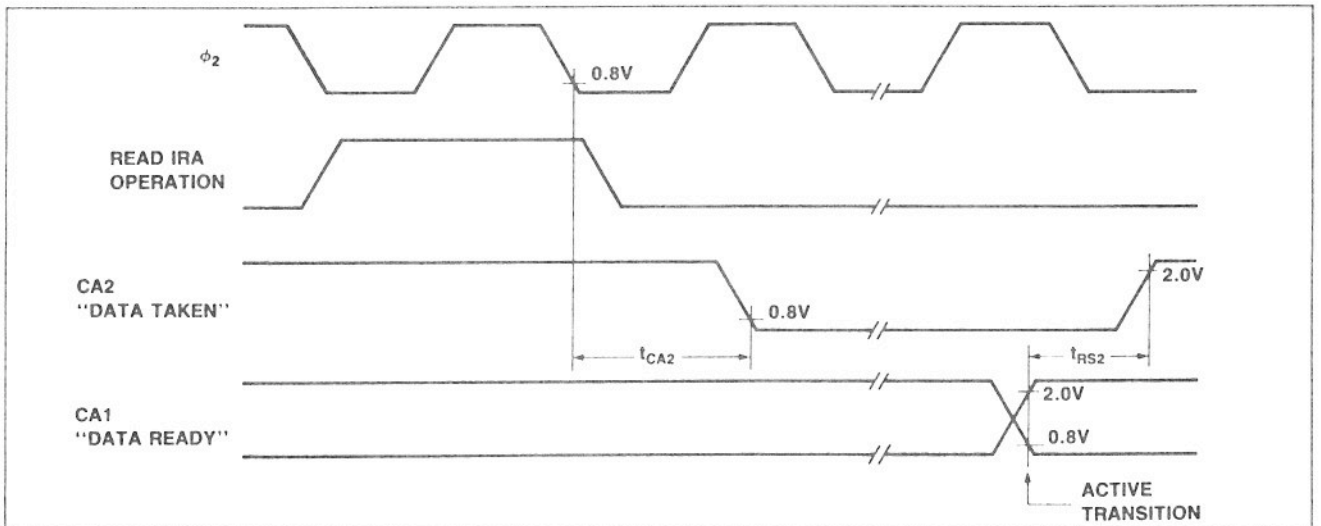


Figure 31b. CA2 Timing for Read Handshake, Handshake Mode

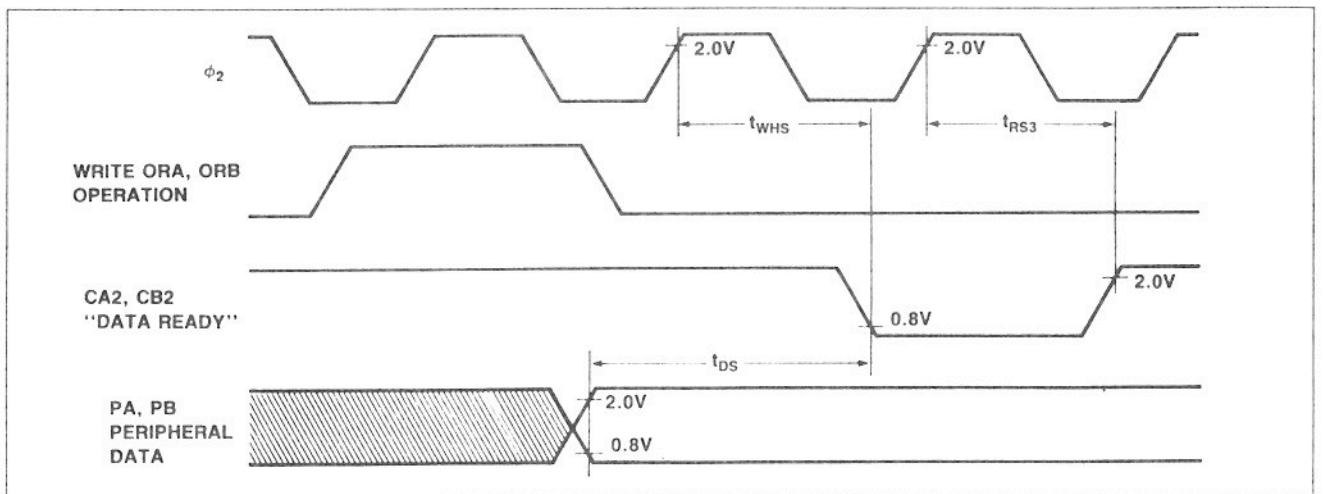


Figure 31c. CA2, CB2 Timing for Write Handshake, Pulse Mode

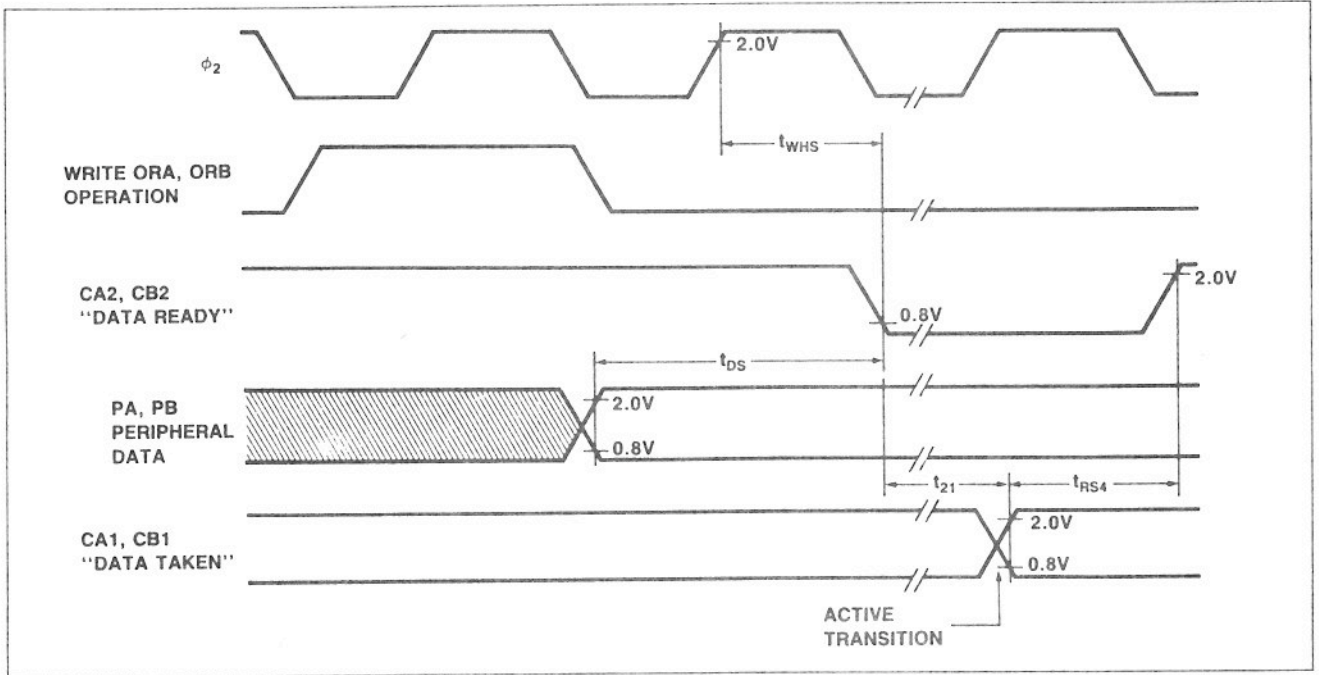


Figure 31d. CA2, CB2 Timing for Write Handshake, Handshake Mode

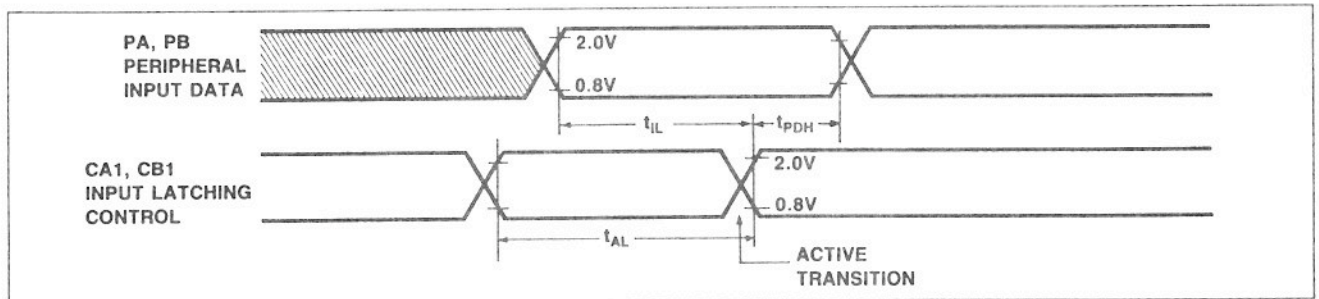


Figure 31e. Peripheral Data Input Latching Timing

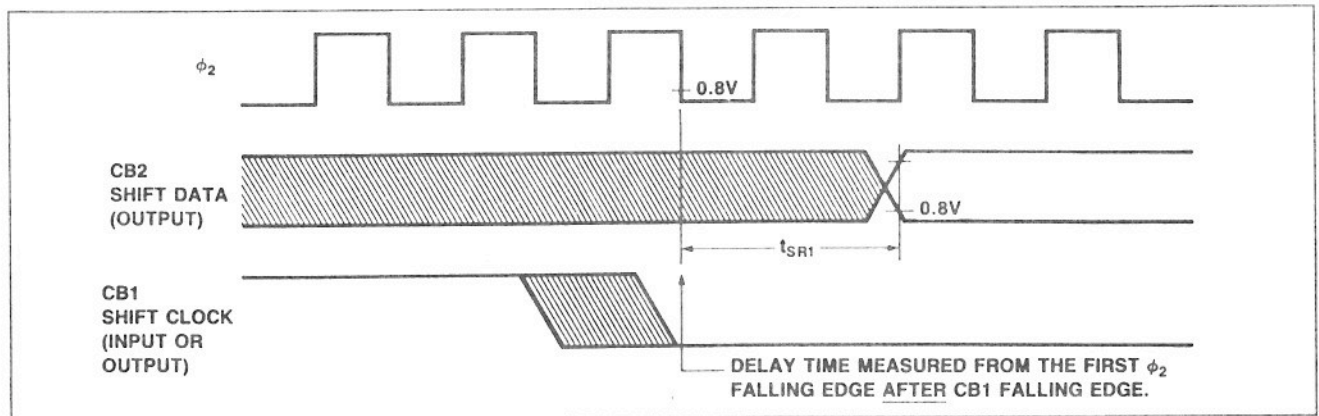


Figure 31f. Timing for Shift Out with Internal or External Shift Clocking

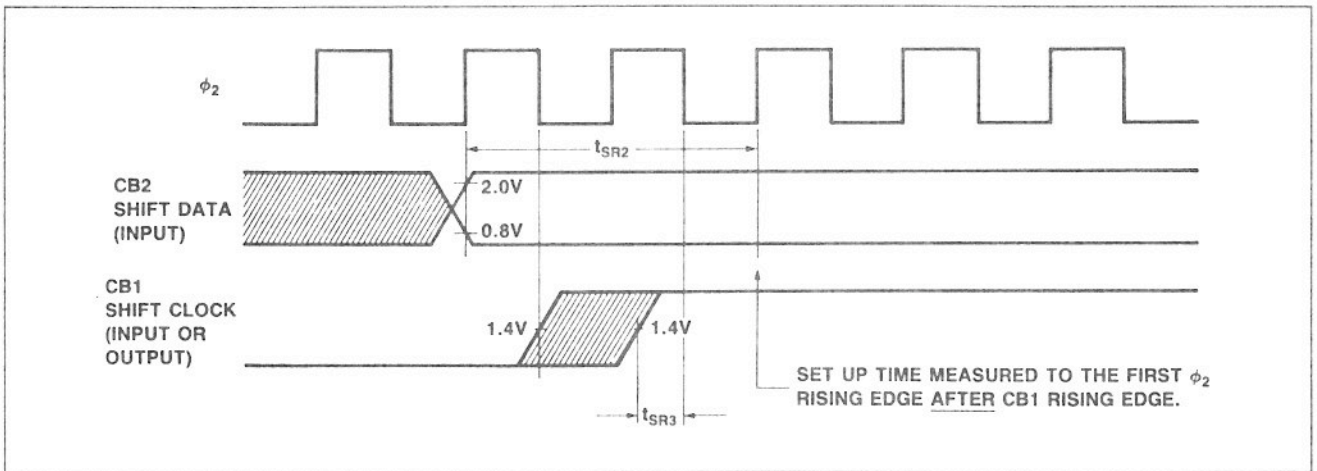


Figure 31g. Timing for Shift in with Internal or External Shift Clocking

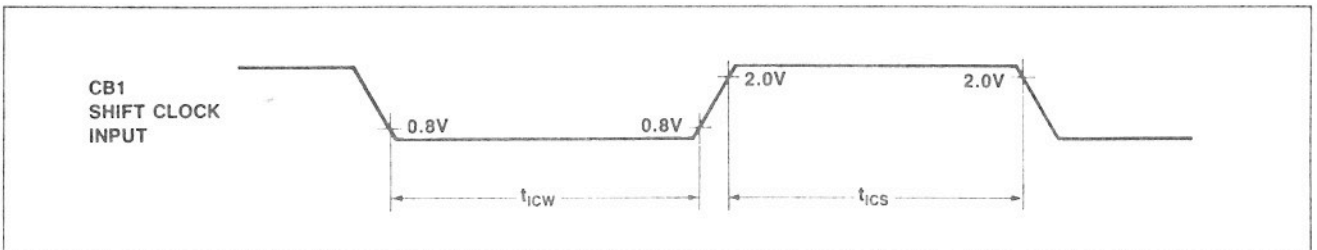


Figure 31h. External Shift Clock Timing

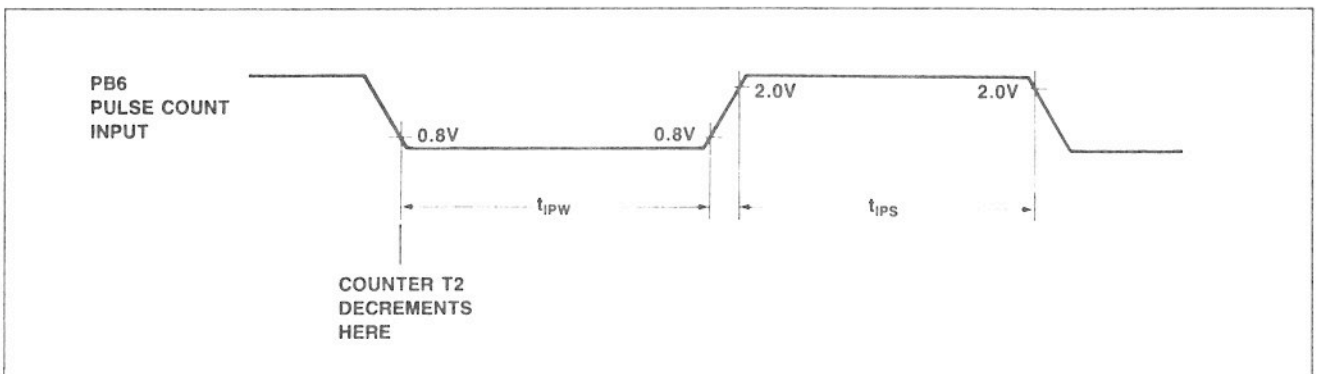


Figure 31i. Pulse Count Input Timing

R6522**Versatile Interface Adapter (VIA)****BUS TIMING CHARACTERISTICS**

| Parameter | Symbol | R6522 (1 MHz) | | R6522A (2 MHz) | | Unit |
|-----------|--------|---------------|------|----------------|------|------|
| | | Min. | Max. | Min. | Max. | |

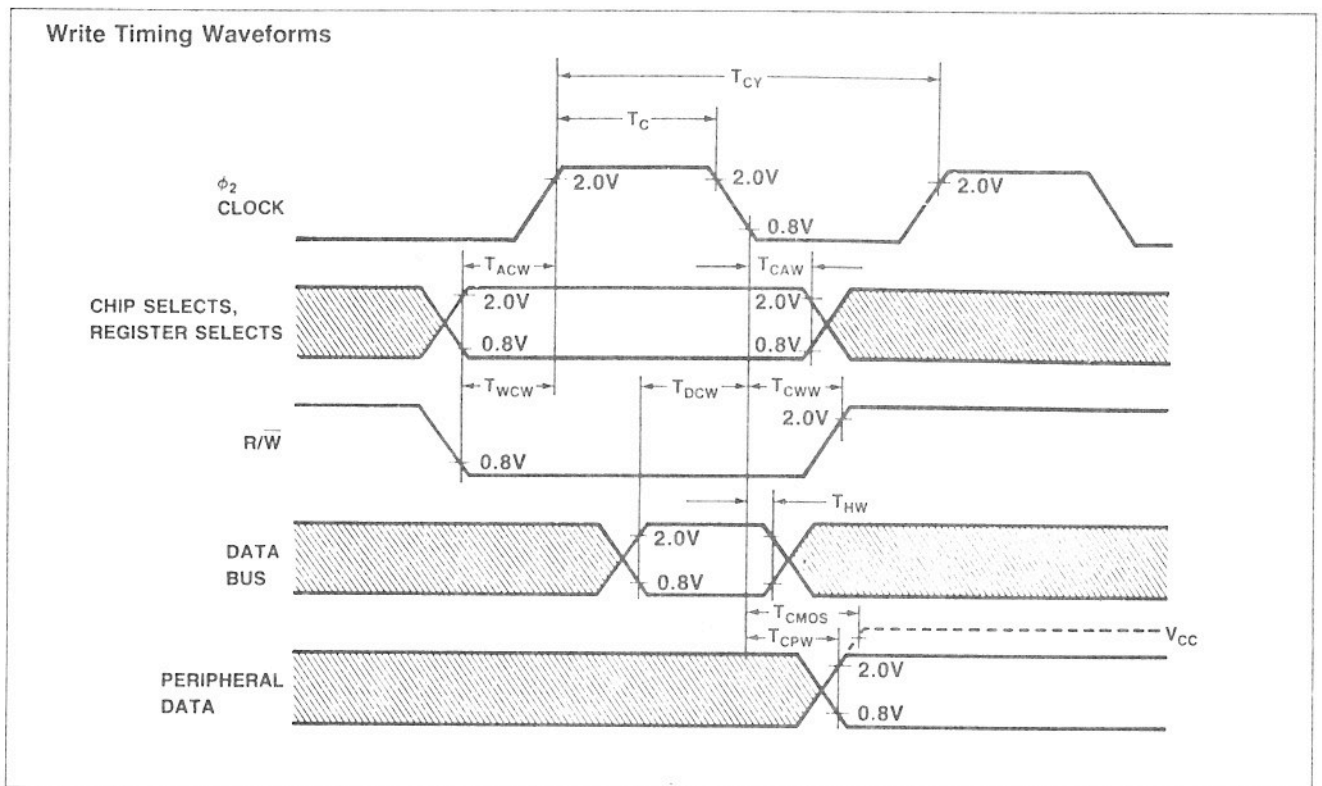
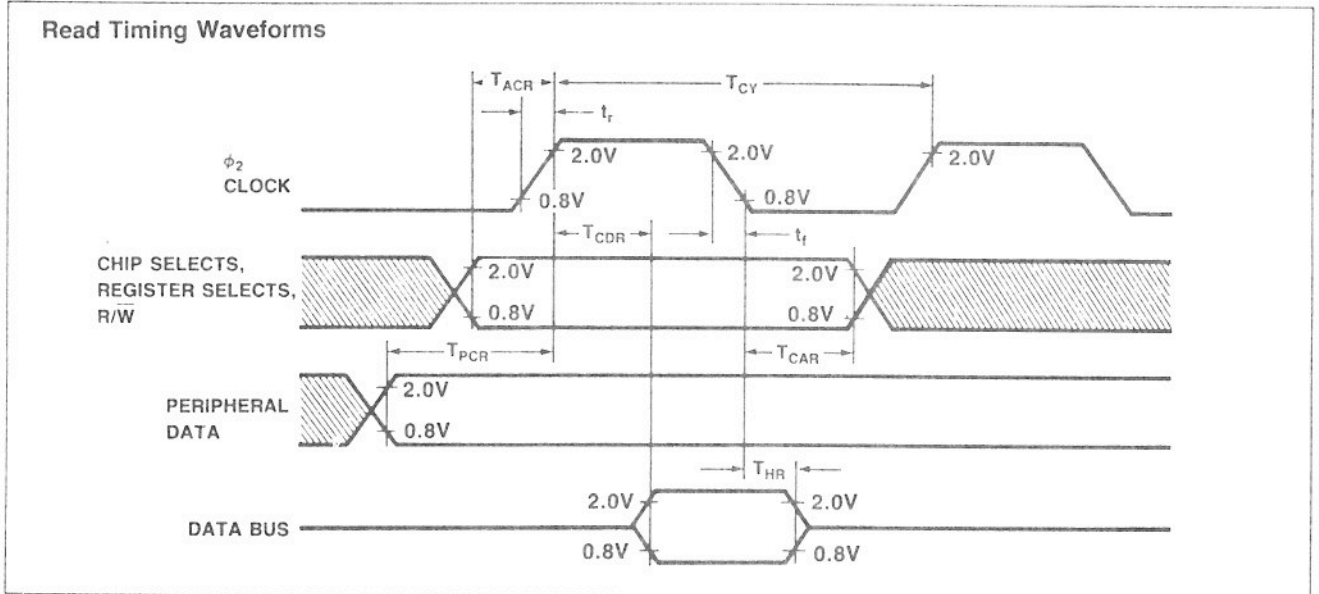
READ TIMING

| | | | | | | |
|-----------------------------|-----------|-----|-----|-----|-----|---------|
| Cycle Time | T_{CY} | 1 | 10 | 0.5 | 10 | μ s |
| Address Set-Up Time | T_{ACR} | 180 | — | 90 | — | ns |
| Address Hold Time | T_{CAR} | 0 | — | 0 | — | ns |
| Peripheral Data Set-Up Time | T_{PCR} | 300 | — | 150 | — | ns |
| Data Bus Delay Time | T_{CDR} | — | 365 | — | 190 | ns |
| Data Bus Hold Time | T_{HR} | 10 | — | 10 | — | ns |

WRITE TIMING

| | | | | | | |
|---|------------|-----|-----|------|-----|---------|
| Cycle Time | T_{CY} | 1 | 10 | 0.50 | 10 | μ s |
| ϕ 2 Pulse width | T_C | 470 | — | 235 | — | ns |
| Address Set-Up Time | T_{ACW} | 180 | — | 90 | — | ns |
| Address Hold Time | T_{CAW} | 0 | — | 0 | — | ns |
| R/\bar{W} Set-Up Time | T_{WCW} | 180 | — | 90 | — | ns |
| R/\bar{W} Hold Time | T_{CWW} | 0 | — | 0 | — | ns |
| Data Bus Set-Up Time | T_{DCW} | 200 | — | 90 | — | ns |
| Data Bus Hold Time | T_{HW} | 10 | — | 10 | — | ns |
| Peripheral Data Delay Time | T_{CPW} | — | 1.0 | — | 0.5 | μ s |
| Peripheral Data Delay Time to CMOS Levels | T_{CMOS} | — | 2.0 | — | 1.0 | μ s |
| Note: t_R and $t_F = 10$ to 30 ns. | | | | | | |

BUS TIMING WAVEFORMS



R6522

Versatile Interface Adapter (VIA)

ABSOLUTE MAXIMUM RATINGS*

| Parameter | 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 Commercial Industrial | T_A | 0 to +70 -40 to +85 | °C °C |
| Storage Temperature | T_{STG} | -55 to +150 | °C |

*NOTE: 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 other sections of this document is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

OPERATING CONDITIONS

| Parameter | Symbol | Value |
|---------------------------------|----------|-------------|
| Supply Voltage | V_{CC} | 5V \pm 5% |
| Temperature Range Commercial | T_A | 0°C to 70°C |

DC CHARACTERISTICS

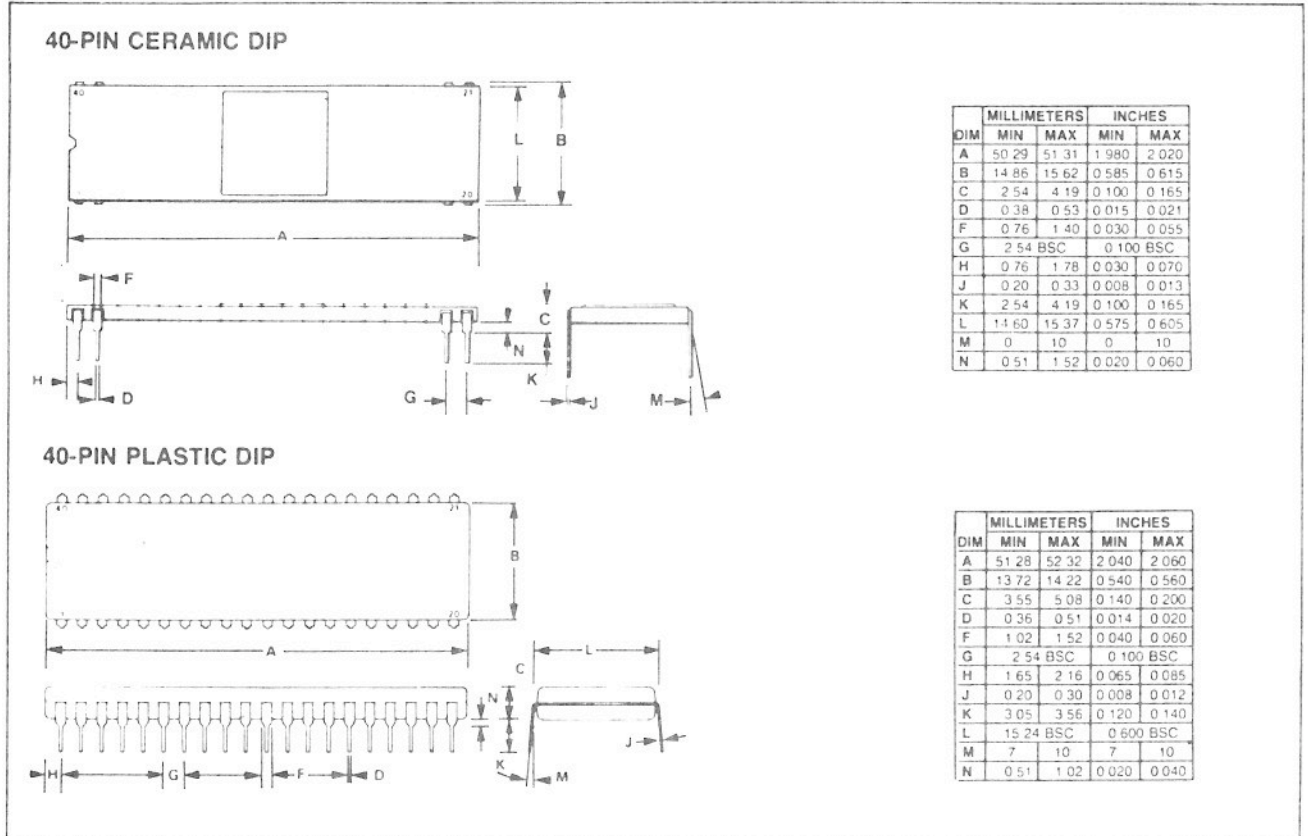
($V_{CC} = 5.0$ Vdc \pm 5%, $V_{SS} = 0$, $T_A = T_L$ to T_H , unless otherwise noted)

| Parameter | Symbol | Min. | Typ. ³ | Max. | Unit | Test Conditions |
|--|-----------|--------------|-------------------|---------------|----------------|---|
| Input High Voltage | V_{IH} | 2.4 | — | V_{CC} | V | |
| Input Low Voltage | V_{IL} | -0.3 | — | 0.4 | V | |
| Input Leakage Current R/W, RES, RS0, RS1, RS2, RS3, CS1, CS2, CA1, \emptyset 2 | I_{IN} | — | ± 1 | ± 2.5 | μ A | $V_{IN} = 0V$ to 5.25V $V_{CC} = 0V$ |
| Input Leakage Current for Three-State Off D0-D07 | I_{TSI} | — | ± 2 | ± 10 | μ A | $V_{IN} = 0.4V$ to 2.4V $V_{CC} = 5.25V$ |
| Input High Current PA0-PA7, CA2, PB0-PB7, CB1, CBS | I_{IH} | -100 | -200 | — | μ A | $V_{IN} = 2.4V$ $V_{CC} = 5.25V$ |
| Input Low Current PA0-PA7, CA2, PB0-PB7, CB1, CB2 | I_{IL} | — | -0.9 | -1.8 | mA | $V_{IL} = 0.4V$ $V_{CC} = 5.25V$ |
| Output High Voltage All outputs PB0-PB7, CB2 (Darlington Drive) | V_{OH} | 2.4 1.5 | — — | — — | V V | $V_{CC} = 4.75V$ $I_{LOAD} = -100 \mu A$ $I_{LOAD} = -1.0 mA$ |
| Output Low Voltage | V_{OL} | — | — | 0.4 | V | $V_{CC} = 4.75V$ $I_{LOAD} = 1.6 mA$ |
| Output High Current (Sourcing) Logic PB0-PB7, CB2 (Darlington Drive) | I_{OH} | -100 -1.0 | -1000 -2.5 | — -10 | μ A mA | $V_{OH} = 2.4V$ $V_{OH} = 1.5V$ |
| Output Low Current (Sinking) | I_{OL} | 1.6 | — | — | mA | $V_{OL} = 0.4V$ |
| Output Leakage Current (Off State) IRQ | I_{OFF} | — | 4 | ± 10 | μ A | $V_{OH} = 2.4V$ $V_{CC} = 5.25V$ |
| Power Dissipation | P_D | — | 450 | 700 | mW | |
| Input Capacitance R/W, RES, RS0, RS1, RS2, RS3, CS1, CS2, D0-D7, PA0-PA7, CA1, CA2, PB0-PB7 CB1, CB2 \emptyset 2 Input | C_{IN} | — | — | 7 10 20 | pF pF pF | $V_{CC} = 5.0V$ $V_{IN} = 0V$ $f = 1 MHz$ $T_A = 25^\circ C$ |
| Output Capacitance | C_{OUT} | — | — | 10 | pF | |

Notes:

- All units are direct current (DC) except for capacitance.
- Negative sign indicates outward current flow, positive indicates inward flow.
- Typical values shown for $V_{CC} = 5.0V$ and $T_A = 25^\circ C$.

PACKAGE DIMENSIONS



Information furnished by Rockwell International Corporation is believed to be accurate and reliable. However, no responsibility is assumed by Rockwell International for its use, nor any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Rockwell International other than for circuitry embodied in a Rockwell product. Rockwell International reserves the right to change circuitry at any time without notice. This specification is subject to change without notice.

©Rockwell International Corporation 1984
All Rights Reserved

Printed in U.S.A.

SEMICONDUCTOR PRODUCTS DIVISION REGIONAL ROCKWELL SALES OFFICES

| | | | |
|---|---|--|---|
| <p>HOME OFFICE Semiconductor Products Division Rockwell International 4311 Jamboree Road P.O. Box C, MS 501-300 Newport Beach, California 92658-8902 (714) 833-4700 TWX: 910 591-1698</p> <p>UNITED STATES Semiconductor Products Division Rockwell International 1842 Reynolds Irvine, California 92714 (714) 833-4655 TWX: 910 595-2518</p> <p>Semiconductor Products Division Rockwell International 3375 Scott Blvd., Suite 410 Santa Clara, California 95051 (408) 980-1900</p> <p>Semiconductor Products Division Rockwell International 921 Bowser Road Richardson, Texas 75080 (214) 996-6500 TLX: 73-307</p> | <p>Semiconductor Products Division Rockwell International 10700 West Higgins Rd., Suite 102 Rosemont, Illinois 60018 (312) 297-8862 TWX: 910 233-0179 (RI MED ROSM)</p> <p>Semiconductor Products Division Rockwell International 5001B Greentree Executive Campus, Rt. 73 Marlton, New Jersey 08053 (609) 596-0090 TWX: 710 940-1377</p> <p>FAR EAST Semiconductor Products Division Rockwell International Overseas Corp. Itohpia Hiraakawa-cho Bldg 7-6, 2-chome, Hiraakawa-cho Chiyoda-ku, Tokyo 102, Japan (03) 265-8806 TLX: J22198</p> <p>Rockwell Collins International Tai Sang Commercial Bldg., 11th Floor 24-34 Hennessy Rd. Hong Kong (5) 274-321 TLX: 74071 HK</p> | <p>EUROPE Semiconductor Products Division Rockwell International GmbH Fraunhoferstrasse 11 D-8033 Munchen-Martinsried West Germany (089) 857-6016 TLX: 0521/2650 rimd d</p> <p>Semiconductor Products Division Rockwell International Heathrow House, Bath Rd. Cranford, Hounslow, Middlesex, England (01) 759-2366 TLX: 851-25463</p> <p>Semiconductor Products Rockwell Collins Italiana S.P.A. Via Boccaccio, 23 20123 Milano, Italy (02) 498.74.79 TLX: 316562 RCIMIL 1</p> | <p>YOUR LOCAL REPRESENTATIVE</p> |
|---|---|--|---|



R6551 ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTER (ACIA)

DESCRIPTION

The Rockwell R6551 Asynchronous Communications Interface Adapter (ACIA) provides an easily implemented, program controlled interface between 8-bit microprocessor-based systems and serial communication data sets and modems.

The ACIA has an internal baud rate generator. This feature eliminates the need for multiple component support circuits, a crystal being the only other part required. The Transmitter baud rate can be selected under program control to be either 1 of 15 different rates from 50 to 19,200 baud, or at $1/16$ times an external clock rate. The Receiver baud rate may be selected under program control to be either the Transmitter rate, or at $1/16$ times the external clock rate. The ACIA has programmable word lengths of 5, 6, 7, or 8 bits; even, odd, or no parity; 1, $1\frac{1}{2}$, or 2 stop bits.

The ACIA is designed for maximum programmed control from the microprocessor (MPU), to simplify hardware implementation. Three separate registers permit the MPU to easily select the R6551's operating modes and data checking parameters and determine operational status.

The Command Register controls parity, receiver echo mode, transmitter interrupt control, the state of the \overline{RTS} line, receiver interrupt control, and the state of the \overline{DTR} line.

The Control Register controls the number of stop bits, word length, receiver clock source, and baud rate.

The Status Register indicates the states of the \overline{IRQ} , \overline{DSR} , and \overline{DCD} lines, Transmitter and Receiver Data Registers, and Overrun, Framing, and Parity Error conditions.

The Transmitter and Receiver Data Registers are used for temporary data storage by the ACIA Transmit and Receiver circuits.

ORDERING INFORMATION

| | |
|---------------------------------------|--|
| Part No.: R6551 | |
| Temperature Range (T_L to T_H): | |
| Blank = 0°C to +70°C | |
| E = -40°C to +85°C | |
| Frequency Range: | |
| 1 = 1 MHz | |
| 2 = 2 MHz | |
| Package: | |
| C = Ceramic | |
| P = Plastic | |

FEATURES

- Compatible with 8-bit microprocessors
- Full duplex operation with buffered receiver and transmitter
- Data set/modem control functions
- Internal baud rate generator with 15 programmable baud rates (50 to 19,200)
- Program-selectable internally or externally controlled receiver rate
- Programmable word lengths, number of stop bits, and parity bit generation and detection
- Programmable interrupt control
- Program reset
- Program-selectable serial echo mode
- Two chip selects
- 2 or 1 MHz operation
- 5.0 Vdc \pm 5% supply requirements
- 28-pin plastic or ceramic DIP
- Full TTL compatibility
- Compatible with R6500, R6500/* and R65C00 microprocessors

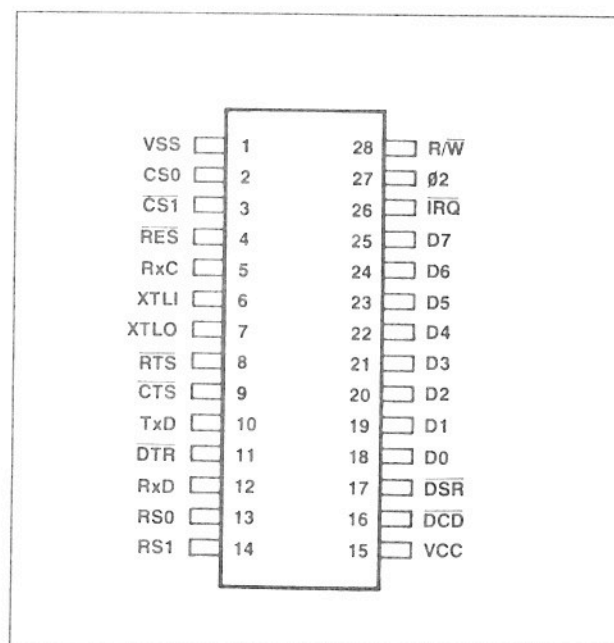


Figure 1. R6551 ACIA Pin Configuration

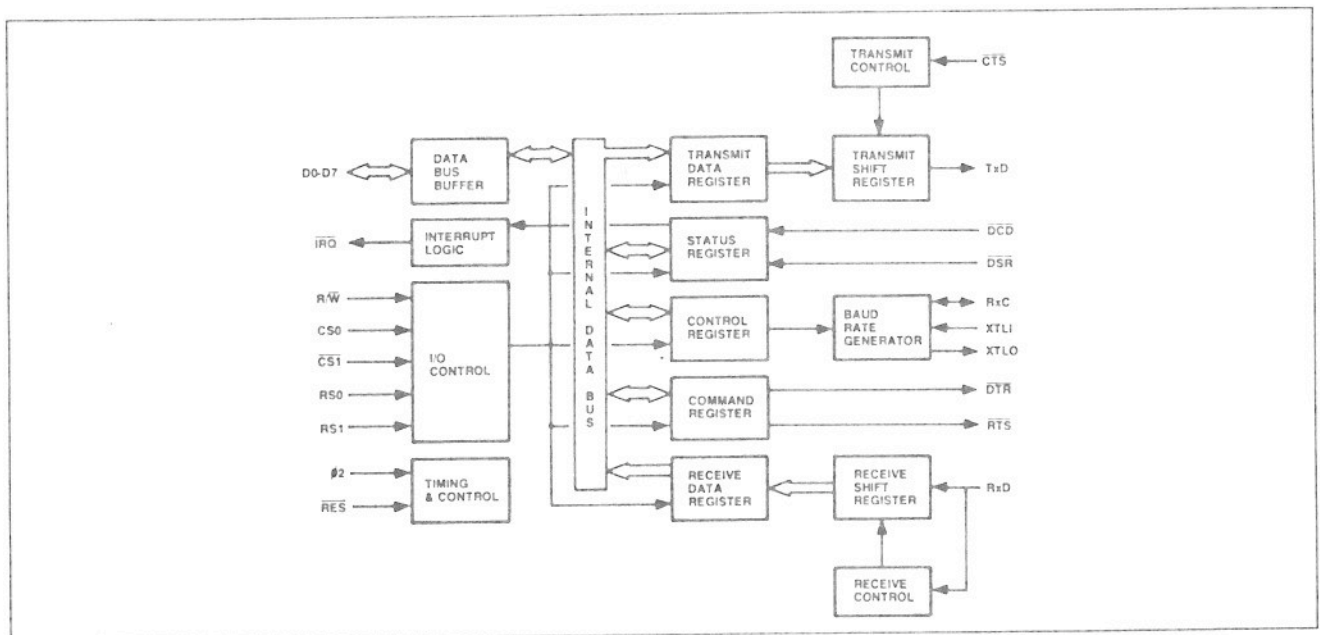


Figure 2. ACIA Internal Organization

FUNCTIONAL DESCRIPTION

A block diagram of the ACIA is presented in Figure 2 followed by a description of each functional element of the device.

DATA BUS BUFFERS

The Data Bus Buffer interfaces the system data lines to the internal data bus. The Data Bus Buffer is bi-directional. When the $\overline{R/W}$ line is high and the chip is selected, the Data Bus Buffer passes the data from the system data lines to the ACIA internal data bus. When the $\overline{R/W}$ line is low and the chip is selected, the Data Bus Buffer writes the data from the internal data bus to the system data bus.

INTERRUPT LOGIC

The Interrupt Logic will cause the \overline{IRQ} line to the microprocessor to go low when conditions are met that require the attention of the microprocessor. The conditions which can cause an interrupt will set bit 7 and the appropriate bit of bits 3 through 6 in the Status Register, if enabled. Bits 5 and 6 correspond to the Data Carrier Detect (\overline{DCD}) logic and the Data Set Ready (\overline{DSR}) logic. Bits 3 and 4 correspond to the Receiver Data Register full and the Transmitter Data Register empty conditions. These conditions can cause an interrupt request if enabled by the Command Register.

I/O CONTROL

The I/O Control Logic controls the selection of internal registers in preparation for a data transfer on the internal data bus and the direction of the transfer to or from the register.

The registers are selected by the Receiver Select ($RS1$, $RS0$) and Read/Write ($\overline{R/W}$) lines as described later in Table 1.

TIMING AND CONTROL

The Timing and Control logic controls the timing of data transfers on the internal data bus and the registers, the Data Bus Buffer, and the microprocessor data bus, and the hardware reset features.

Timing is controlled by the system $\phi 2$ clock input. The chip will perform data transfers to or from the microcomputer data bus during the $\phi 2$ high period when selected.

All registers will be initialized by the Timing and Control Logic when the Reset (\overline{RES}) line goes low. See the individual register description for the state of the registers following a hardware reset.

TRANSMITTER AND RECEIVER DATA REGISTERS

These registers are used as temporary data storage for the ACIA Transmit and Receive Circuits. Both the Transmitter and Receiver are selected by a Register Select 0 ($RS0$) and Register Select 1 ($RS1$) low condition. The Read/Write ($\overline{R/W}$) line determines which actually uses the internal data bus; the Transmitter Data Register is write only and the Receiver Data Register is read only.

Bit 0 is the first bit to be transmitted from the Transmitter Data Register (least significant bit first). The higher order bits follow in order. Unused bits in this register are "don't care".

The Receiver Data Register holds the first received data bit in bit 0 (least significant bit first). Unused high-order bits are "0". Parity bits are not contained in the Receiver Data Register. They are stripped off after being used for parity checking.

R6551

Asynchronous Communications Interface Adapter (ACIA)

STATUS REGISTER

The Status Register indicates the state of interrupt conditions and other non-interrupt status lines. The interrupt conditions are the Data Set Ready, Data Carrier Detect, Transmitter Data Register Empty and Receiver Data Register Full as reported in bits 6 through 3, respectively. If any of these bits are set, the Interrupt (IRQ) indicator (bit 7) is also set. Overrun, Framing Error, and Parity Error are also reported (bits 2 through 0 respectively).

| | | | | | | | |
|-----|------------------|------------------|------|------|------|----|----|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| IRQ | \overline{DSR} | \overline{DCD} | TDRE | RDRE | OVRN | FE | PE |

- Bit 7** **Interrupt (IRQ)**
 0 No interrupt
 1 Interrupt has occurred
- Bit 6** **Data Set Ready (\overline{DSR})**
 0 \overline{DSR} low (ready)
 1 \overline{DSR} high (not ready)
- Bit 5** **Data Carrier Detect (\overline{DCD})**
 0 \overline{DCD} low (detected)
 1 \overline{DCD} high (not detected)
- Bit 4** **Transmitter Data Register Empty**
 0 Not empty
 1 Empty
- Bit 3** **Receiver Data Register Full**
 0 Not full
 1 Full
- Bit 2** **Overrun***
 0 No overrun
 1 Overrun has occurred
- Bit 1** **Framing Error***
 0 No framing error
 1 Framing error detected
- Bit 0** **Parity Error***
 0 No parity error
 1 Parity error detected

*No interrupt occurs for these conditions

Reset Initialization

| | | | | | | | | |
|---|---|---|---|---|---|---|---|----------------|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| 0 | — | — | 1 | 0 | 0 | 0 | 0 | Hardware reset |
| — | — | — | — | — | 0 | — | — | Program reset |

Parity Error (Bit 0), Framing Error (Bit 1), and Overrun (2)

None of these bits causes a processor interrupt to occur, but they are normally checked at the time the Receiver Data Register is read so that the validity of the data can be verified. These bits are self clearing (i.e., they are automatically cleared after a read of the Receiver Data Register).

Receiver Data Register Full (Bit 3)

This bit goes to a 1 when the ACIA transfers data from the Receiver Shift Register to the Receiver Data Register, and goes to a 0 (is cleared) when the processor reads the Receiver Data Register.

Transmitter Data Register Empty (Bit 4)

This bit goes to a 1 when the ACIA transfers data from the Transmitter Data Register to the Transmitter Shift Register, and goes to a 0 (is cleared) when the processor writes new data onto the Transmitter Data Register.

Data Carrier Detect (Bit 5) and Data Set Ready (Bit 6)

These bits reflect the levels of the \overline{DCD} and \overline{DSR} inputs to the ACIA. A 0 indicates a low level (true condition) and a 1 indicates a high level (false). Whenever either of these inputs change state, an immediate processor interrupt (IRQ) occurs, unless bit 1 of the Command Register (IRD) is set to a 1 to disable \overline{IRQ} . When the interrupt occurs, the status bits indicate the levels of the inputs immediately after the change of state occurred. Subsequent level changes will not affect the status bits until the Status Register is interrogated by the processor. At that time, another interrupt will immediately occur and the status bits reflect the new input levels. These bits are not automatically cleared (or reset) by an internal operation.

Interrupt (Bit 7)

This bit goes to a 1 whenever an interrupt condition occurs and goes to a 0 (is cleared) when the Status Register is read.

R6551 Asynchronous Communications Interface Adapter (ACIA)

CONTROL REGISTER

The Control Register selects the desired baud rate, frequency source, word length, and the number of stop bits.

| | | | | | | | |
|-----|-----|-----|-----|------|------|------|------|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SBN | WL | | RCS | SBR | | | |
| | WL1 | WL0 | | SBR3 | SBR2 | SBR1 | SBR0 |

Bit 7 Stop Bit Number (SBN)
 0 1 Stop bit
 1 2 Stop bits
 1 1½ Stop bits
 For WL = 5 and no parity
 1 1 Stop bit
 For WL = 8 and parity

Bits 6-5 Word Length (WL)

| 6 | 5 | No. Bits |
|---|---|----------|
| 0 | 0 | 8 |
| 0 | 1 | 7 |
| 1 | 0 | 6 |
| 1 | 1 | 5 |

Bit 4 Receiver Clock Source (RCS)
 0 External receiver clock
 1 Baud rate

Bits 3-0 Selected Baud Rate (SBR)

| 3 | 2 | 1 | 0 | Baud |
|---|---|---|---|--------------------|
| 0 | 0 | 0 | 0 | 16x External Clock |
| 0 | 0 | 0 | 1 | 50 |
| 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 |

Reset Initialization

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| - | - | - | - | - | - | - | - |

Hardware reset (\overline{RES})
 Program reset

Selected Baud Rate (Bits 0, 1, 2, 3)

These bits select the Transmitter baud rate, which can be at 1/16 an external clock rate or one of 15 other rates controlled by the internal baud rate generator.

If the Receiver clock uses the same baud rate as the transmitter, then RxC becomes an output and can be used to slave other circuits to the ACIA. Figure 3 shows the Transmitter and Receiver layout.

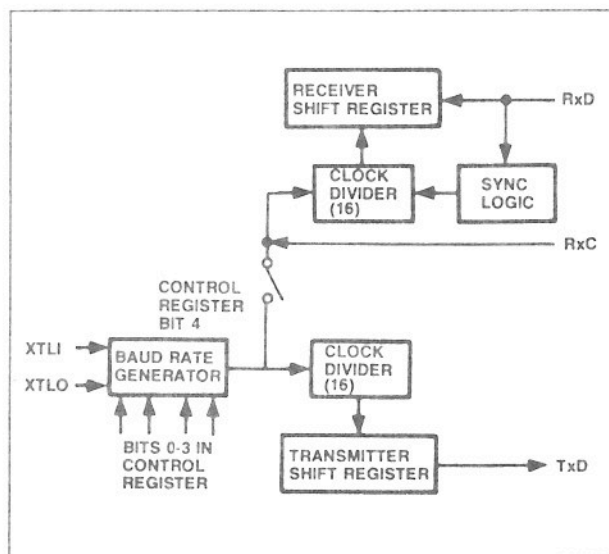


Figure 3. Transmitter/Receiver Clock Circuits

Receiver Clock Source (Bit 4)

This bit controls the clock source to the Receiver. A 0 causes the Receiver to operate at a baud rate of 1/16 an external clock. A 1 causes the Receiver to operate at the same baud rate as is selected for the transmitter.

Word Length (Bits 5, 6)

These bits determine the word length to be used (5, 6, 7 or 8 bits).

Stop Bit Number (Bit 7)

This bit determines the number of stop bits used. A 0 always indicates one stop bit. A 1 indicates 1½ stop bits if the word length is 5 with no parity selected, 1 stop bit if the word length is 8 with parity selected, or 2 stop bits in all other configurations.

R6551

Asynchronous Communications Interface Adapter (ACIA)

COMMAND REGISTER

The Command Register controls specific modes and functions.

| | | | | | | | |
|------|------|-----|-----|------|------|-----|-----|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PMC | | PME | REM | TIC | | IRD | DTR |
| PMC1 | PMC0 | | | TIC1 | TIC0 | | |

Bits 7-6 Parity Mode Control (PMC)

| | | |
|---|---|---|
| 7 | 6 | |
| 0 | 0 | Odd parity transmitted/received |
| 0 | 1 | Even parity transmitted/received |
| 1 | 0 | Mark parity bit transmitted Parity check disabled |
| 1 | 1 | Space parity bit transmitted Parity check disabled |

Bit 5 Parity Mode Enabled (PME)

| | |
|---|--|
| 0 | Parity mode disabled No parity bit generated Parity check disabled |
| 1 | Parity mode enabled |

Bit 4 Receiver Echo Mode (REM)

| | |
|---|--|
| 0 | Receiver normal mode |
| 1 | Receiver echo mode bits 2 and 3 Must be zero for receiver echo mode, RTS will be low. |

Bits 3-2 Transmitter Interrupt Control (TIC)

| | | |
|---|---|---|
| 3 | 2 | |
| 0 | 0 | $\overline{\text{RTS}}$ = High, transmit interrupt disabled |
| 0 | 1 | $\overline{\text{RTS}}$ = Low, transmit interrupt enabled |
| 1 | 0 | $\overline{\text{RTS}}$ = Low, transmit interrupt disabled |
| 1 | 1 | $\overline{\text{RTS}}$ = Low, transmit interrupt disabled transmit break on TxD |

Bit 1 Interrupt Request Disabled (IRD)

| | |
|---|----------------------------------|
| 0 | $\overline{\text{IRQ}}$ enabled |
| 1 | $\overline{\text{IRQ}}$ disabled |

Bit 0 Data Terminal Ready (DTR)

| | |
|---|------------------------------------|
| 0 | Data terminal not ready (DTR high) |
| 1 | Data terminal ready (DTR low) |

Data Terminal Ready (Bit 0)

This bit enables all selected interrupts and controls the state of the Data Terminal Ready ($\overline{\text{DTR}}$) line. A 0 indicates the microcomputer system is not ready by setting the $\overline{\text{DTR}}$ line high. A 1 indicates the microcomputer system is ready by setting the $\overline{\text{DTR}}$ line low.

Receiver Interrupt Control (Bit 1)

This bit disables the Receiver from generating an interrupt when set to a 1. The Receiver interrupt is enabled when this bit is set to a 0 and Bit 0 is set to a 1.

Transmitter Interrupt Control (Bits 2, 3)

These bits control the state of the Ready to Send ($\overline{\text{RTS}}$) line and the Transmitter interrupt.

Receiver Echo Mode (Bit 4)

A 1 enables the Receiver Echo Mode and a 0 enables the Receiver Echo Mode. When bit 4 is a 1, bits 2 and 3 must be 0. In the Receiver Echo Mode, the Transmitter returns each transmission received by the Receiver delayed by one-half bit time.

Parity Mode Enable (Bit 5)

This bit enables parity bit generation and checking. A 0 disables parity bit generation by the Transmitter and parity bit checking by the Receiver. A 1 bit enables generation and checking of parity bits.

Parity Mode Control (Bits 6, 7)

These bits determine the type of parity generated by the Transmitter, (even, odd, mark or space) and the type of parity check done by the Receiver (even, odd, or no check).

Reset Initialization

| | | | | | | | | |
|---|---|---|---|---|---|---|---|--|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Hardware reset ($\overline{\text{RES}}$) |
| - | - | - | 0 | 0 | 0 | 0 | 0 | Program reset |

R6551

Asynchronous Communications Interface Adapter (ACIA)

INTERFACE SIGNALS

Figure 4 shows the ACIA interface signals associated with the microprocessor and the modem.

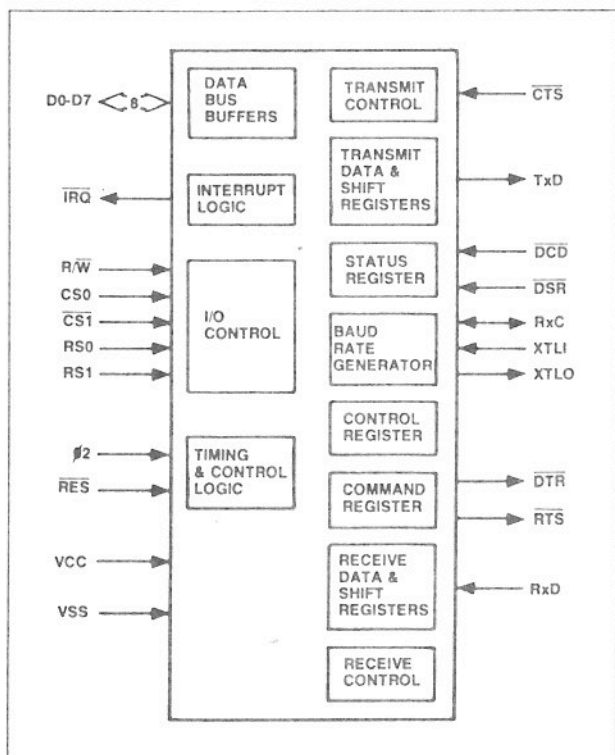


Figure 4. ACIA Interface Diagram

MICROPROCESSOR INTERFACE

Reset (\overline{RES})

During system initialization a low on the \overline{RES} input causes a hardware reset to occur. Upon reset, the Command Register and the Control Register are cleared (all bits set to 0). The Status Register is cleared with the exception of the indications of Data Set Ready and Data Carrier Detect, which are externally controlled by the DSR and DCD lines, and the transmitter Empty bit, which is set. \overline{RES} must be held low for one $\phi 2$ clock cycle for a reset to occur.

Input Clock ($\phi 2$)

The input clock is the system $\phi 2$ clock and clocks all data transfers between the system microprocessor and the ACIA.

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

The R/\overline{W} input, generated by the microprocessor controls the direction of data transfers. A high on the R/\overline{W} pin allows the processor to read the data supplied by the ACIA, a low allows a write to the ACIA.

Interrupt Request (\overline{IRQ})

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

Data Bus (D0-D7)

The eight data line (D0-D7) pins transfer data between the processor and the ACIA. These lines are bi-directional and are normally high-impedance except during Read cycles when the ACIA is selected.

Chip Selects ($CS0$, $\overline{CS1}$)

The two chip select inputs are normally connected to the processor address lines either directly or through decoders. The ACIA is selected when $CS0$ is high and $\overline{CS1}$ is low. When the ACIA is selected, the internal registers are addressed in accordance with the register select lines ($RS0$, $RS1$).

Register Selects ($RS0$, $RS1$)

The two register select lines are normally connected to the processor address lines to allow the processor to select the various ACIA internal registers. Table 1 shows the internal register select coding.

Table 1. ACIA Register Selection

| RS1 | RS0 | Register Operation | |
|-----|-----|---|-----------------------------|
| | | R/W = Low | R/W = High |
| L | L | Write Transmit Data Register | Read Receiver Data Register |
| L | H | Programmed Reset (Data is "Don't Care") | Read Status Register |
| H | L | Write Command Register | Read Command Register |
| H | H | Write Control Register | Read Control Register |

Only the Command and Control registers can both be read and written. The programmed Reset operation does not cause any data transfer, but is used to clear bits 4 through 0 in the Command register and bit 2 in the Status Register. The Control Register is unchanged by a programmed Reset. It should be noted that the programmed Reset is slightly different from the hardware Reset (\overline{RES}); refer to the register description.

R6551

Asynchronous Communications Interface Adapter (ACIA)

ACIA/MODEM INTERFACE

Crystal Pins (XTLI, XTLO)

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

Transmit Data (TxD)

The TxD output line transfers serial nonreturn-to-zero (NRZ) data to the modem. The least significant bit (LSB) of the Transmit Data Register is the first data bit transmitted and the rate of data transmission is determined by the baud rate selected or under control of an external clock. This selection is made by programming the Control Register.

Receive Data (RxD)

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

Receive Clock (RxC)

The RxC is a bi-directional pin which is 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.

Request to Send (RTS)

The RTS output pin controls the modem from the processor. The state of the RTS pin is determined by the contents of the Command Register.

Clear to Send (CTS)

The CTS input pin controls the transmitter operation. The enable state is with CTS low. The transmitter is automatically disabled if CTS is high.

Data Terminal Ready (DTR)

This output pin indicates the status of the ACIA to the modem. A low on DTR indicates the ACIA is enabled, a high indicates it is disabled. The processor controls this pin via bit 0 of the Command Register.

Data Set Ready (DSR)

The DSR input pin indicates to the ACIA the status of the modem. A low indicates the "ready" state and a high, "not-ready."

Data Carrier Detect (DCD)

The DCD input pin indicates to the ACIA 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.

TRANSMITTER AND RECEIVER OPERATION

Continuous Data Transmit

In the normal operating mode, the interrupt request output (IRQ) signals when the ACIA is ready to accept the next data word to be transmitted. This interrupt occurs at the beginning of the Start Bit. When the processor reads the Status Register of the ACIA, the interrupt is cleared.

The processor must then identify that the Transmit Data Register is ready to be loaded and must then load it with the next data word. This must occur before the end of the Stop Bit, otherwise a continuous "MARK" will be transmitted. Figure 5 shows the continuous Data Transmit timing relationship.

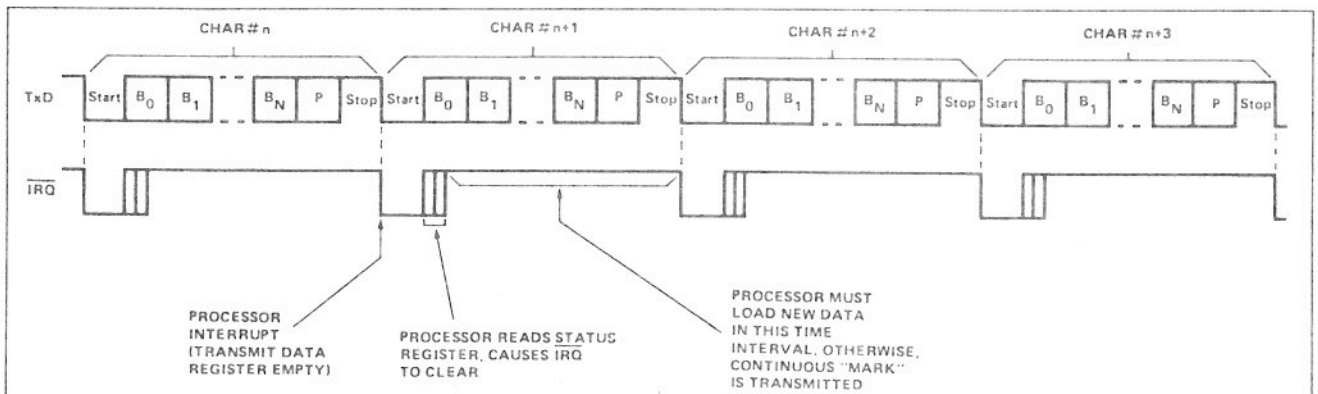


Figure 5. Continuous Data Transmit

R6551

Asynchronous Communications Interface Adapter (ACIA)

Continuous Data Receive

Similar to the Continuous Data Transmit case, the normal operation of this mode is to assert $\overline{\text{IRQ}}$ when the ACIA has received a full data word. This occurs at about $9/16$ point through the Stop Bit. The processor must read the Status Register and

read the data word before the next interrupt, otherwise the Overrun condition occurs. Figure 6 shows the continuous Data Receive Timing Relationship.

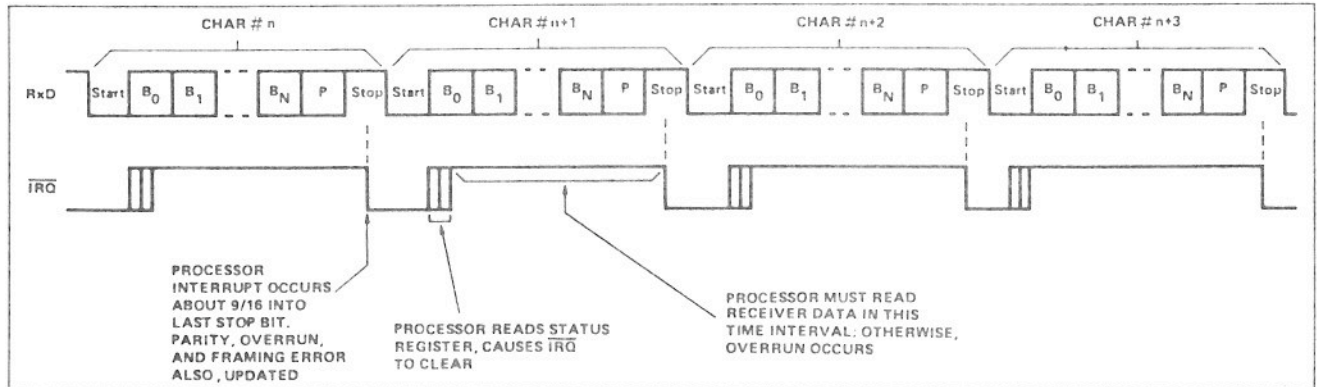


Figure 6. Continuous Data Receive

Transmit Data Register Not Loaded by Processor

If the processor is unable to load the Transmit Data Register in the allocated time, then the TxD line goes to the "MARK" condition until the data is loaded. $\overline{\text{IRQ}}$ interrupts continue to occur at the same rate as previously, except no data is transmitted.

When the processor finally loads new data, a Start Bit immediately occurs, the data word transmission is started, and another interrupt is initiated, signaling for the next data word. Figure 7 shows the timing relationship for this mode of operation.

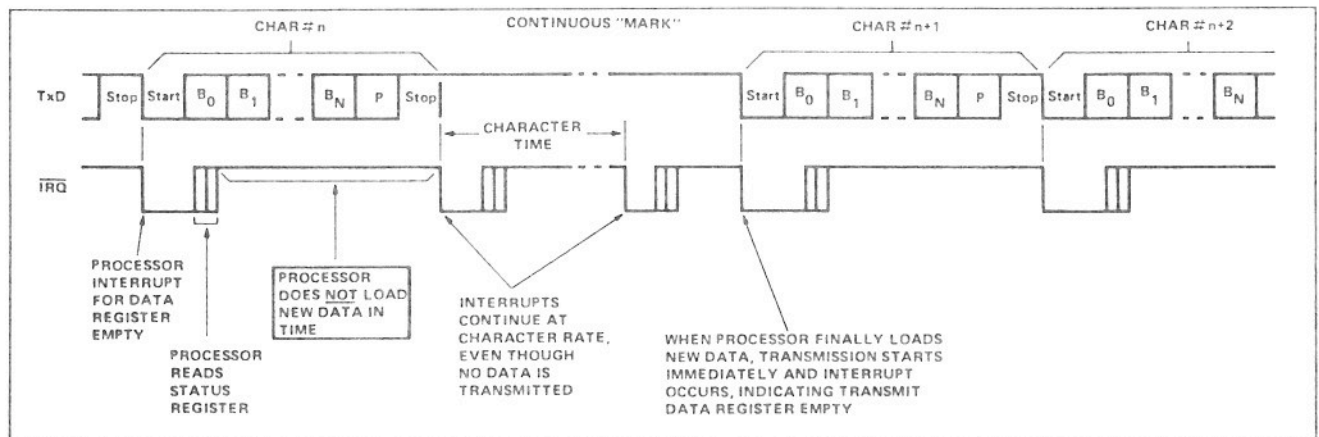


Figure 7. Transmit Data Register Not Loaded by Processor

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 20 illustrates the configuration of the SR data bits and Figure 21 shows the SR control bits of the ACR.

SR Mode 0 — Disabled

Mode 0 disables the Shift Register. In this mode the microprocessor can write or read the SR and the SR will shift on each CB1 positive edge shifting in the value on CB2. In this mode the SR interrupt flag is disabled (held to a logic 0).

SR Mode 1 — Shift In Under Control of T2

In mode 1, the shifting rate is controlled by the low order 8 bits of T2 (Figure 22). 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 the read or write of the SR if the SR flag is set in the IFR. Otherwise the first shift will occur at the next time-out of T2 after a read or write of the SR. 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 set and IRQ will go low.

SR Mode 2 — Shift In Under ϕ_2 Control

In mode 2, the shift rate is a direct function of the system clock frequency (Figure 23). 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.

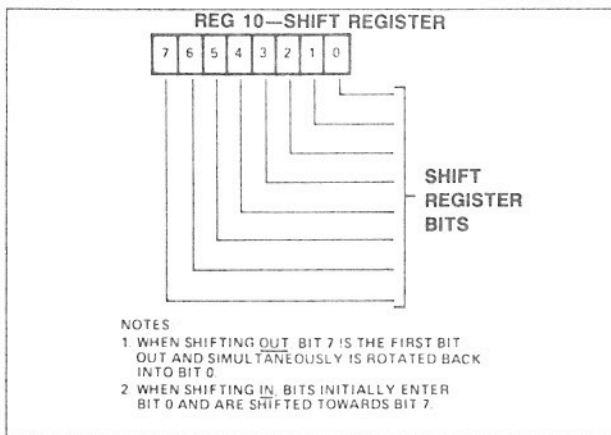


Figure 20. Shift Registers

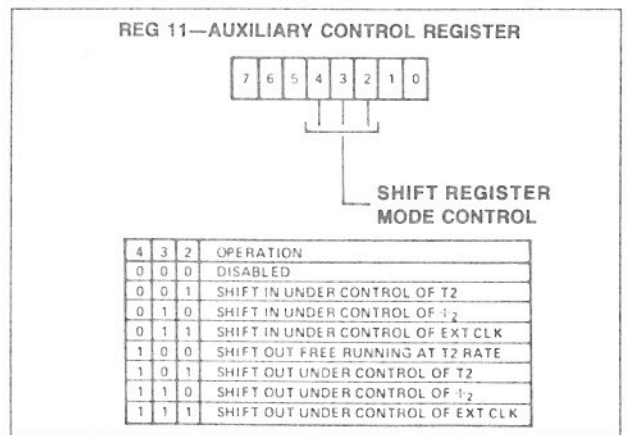


Figure 21. Shift Register Modes

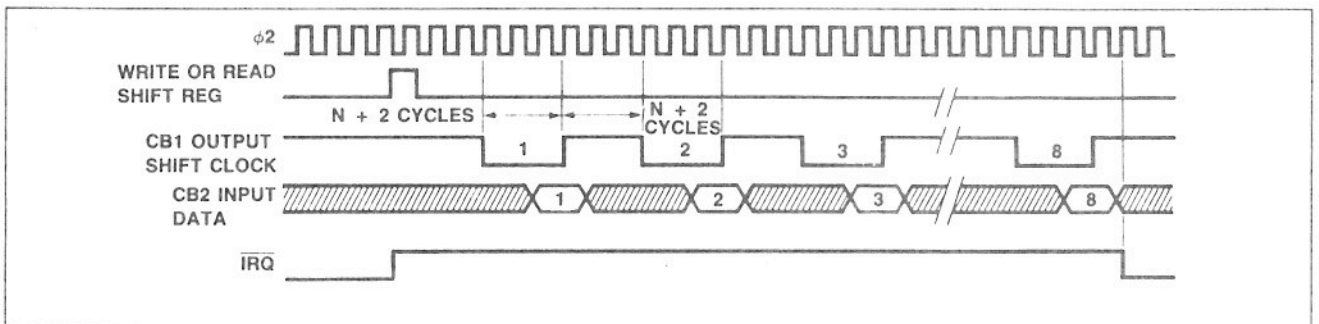


Figure 22. SR Mode 1 — Shift In Under T2 Control

SR Mode 3 — Shift in Under CB1 Control

In mode 3, external pin CB1 becomes an input (Figure 24). 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.

SR Mode 4 — Shift Out Under T2 Control (Free-Run)
Mode 4 is very similar to mode 5 in which the shifting rate is set by

T2. However, in mode 4 the SR Counter does not stop the shifting operation (Figure 25). 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.

SR Mode 5 — Shift Out Under T2 Control

In mode 5, the shift rate is controlled by T2 (as in mode 4). The shifting operation is triggered by the read or write of the SR if the SR flag is set in the IFR (Figure 26). Otherwise the first shift will occur at the next time-out of T2 after a read or write of the SR. 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.

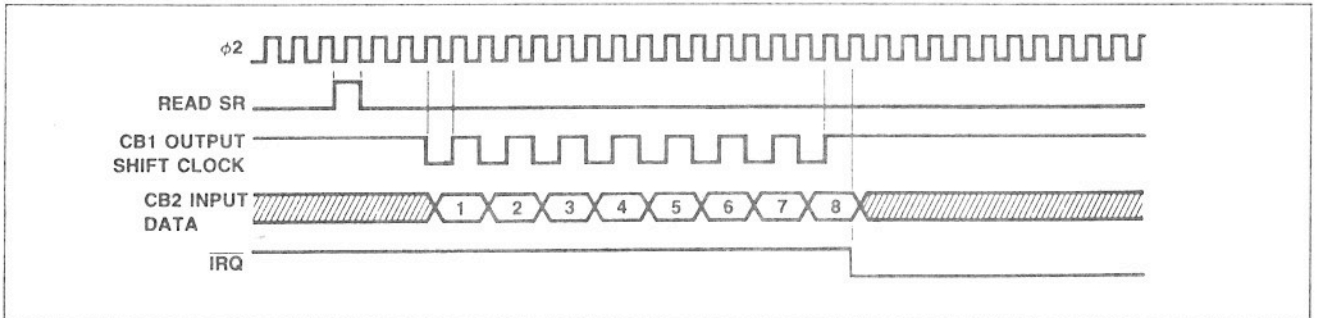


Figure 23. SR Mode 2 — Shift In Center ϕ_2 Control

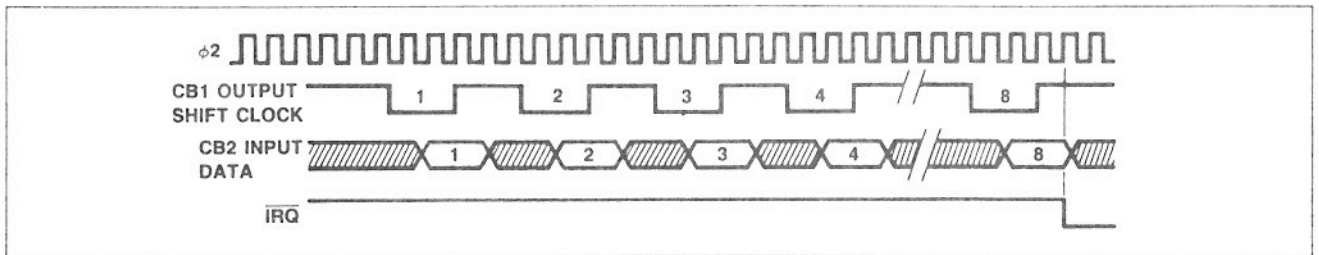


Figure 24. SR Mode 3 — Shift In Under CB1 Control

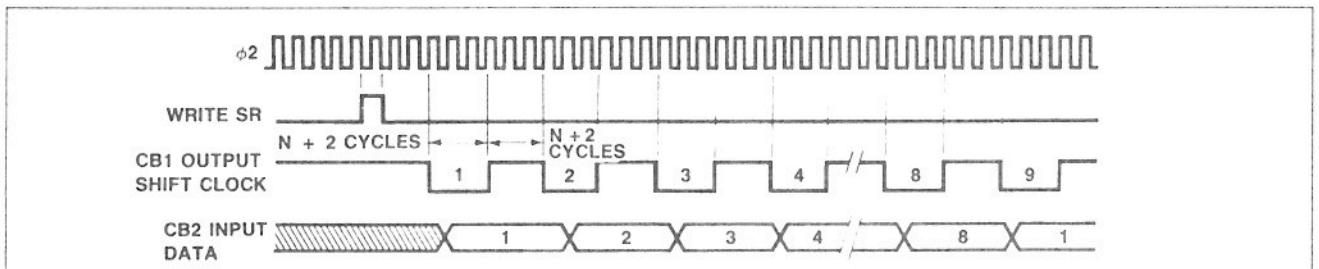


Figure 25. SR Mode 4 — Shift Our Under T2 Control (Free-Run)

SR Mode 6 — Shift Out Under $\phi 2$ Control

In mode 6, the shift rate is controlled by the $\phi 2$ system clock (Figure 27).

SR Mode 7 — Shift Out Under CB1 Control

In mode 7, shifting is controlled by pulses applied to the CB1 pin by an external device (Figure 28). 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.

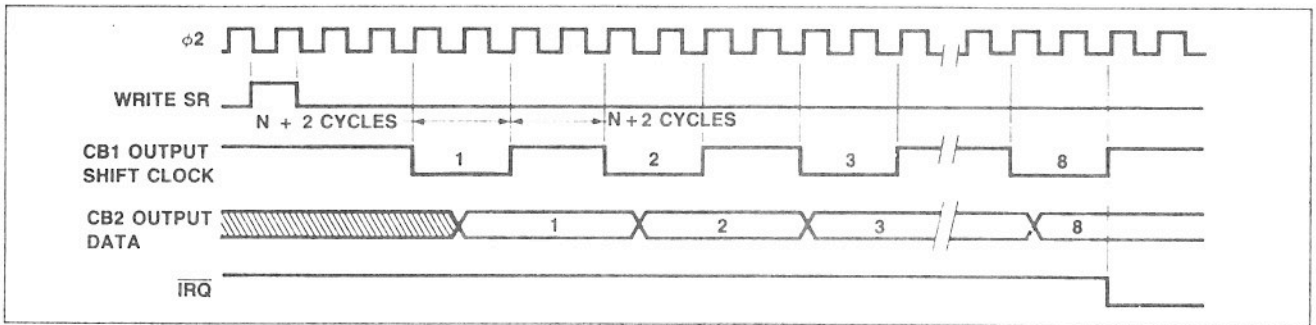


Figure 26. SR Mode 5 — Shift Out Under T2 Control

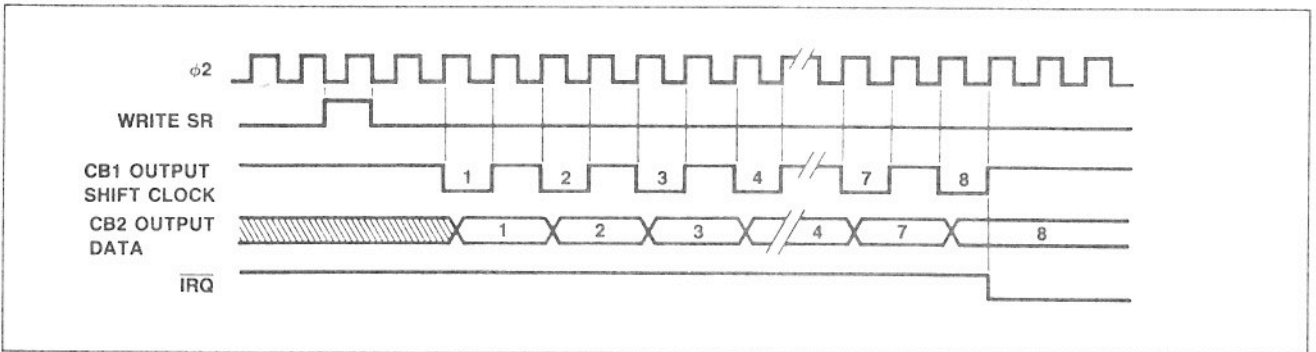


Figure 27. SR Mode 6 — Shift Out Under $\phi 2$ Control

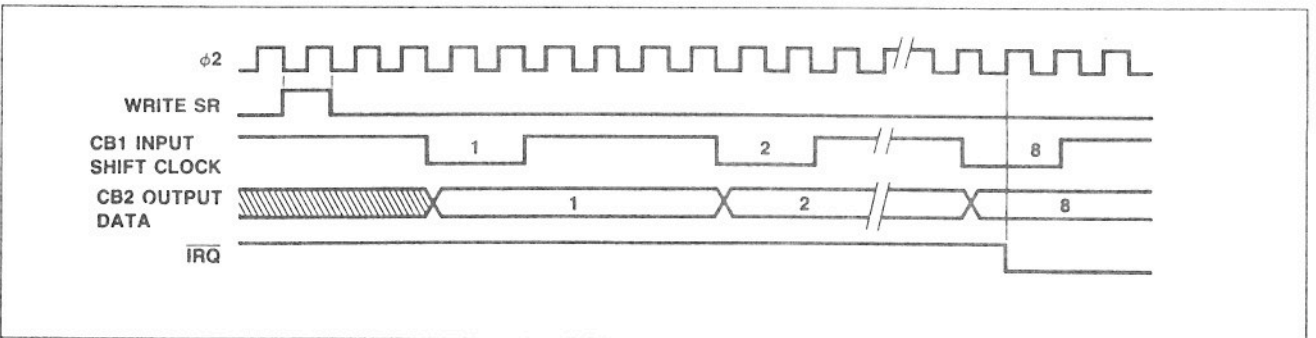


Figure 28. SR Mode 7 — Shift Out Under CB1 Control

R6522

Versatile Interface Adapter (VIA)

Interrupt Operation

Controlling interrupts within the R6522 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 in the Interrupt Flag Register (IFR) by conditions detected within the R6522 or on inputs to the R6522. 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.

Associated with each interrupt flag is an interrupt enable bit in the Interrupt Enable Register (IER). 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 (IRQ) will go low. IRQ is an "open-collector" output which can be "wired-OR"ed with other devices in the system to interrupt the processor.

Interrupt Flag Register (IFR)

In the R6522, all the interrupt flags are contained in one register, i.e., the IFR (Figure 29). 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.

The Interrupt Flag Register (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 $\overline{\text{IRQ}}$ output. This bit corresponds to the logic function: $\overline{\text{IRQ}} = \text{IFR6} \times \text{IER6} + \text{IFR5} \times \text{IER5} + \text{IFR4} \times \text{IER4} + \text{IFR3} \times \text{IER3} + \text{IFR2} \times \text{IER2} + \text{IFR1} \times \text{IER1} + \text{IFR0} \times \text{IER0}$.

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.

Interrupt Enable Register (IER)

For each interrupt flag in IFR, there is a corresponding bit in the Interrupt Enable Register (IER) (Figure 30). Individual bits in the IER can be set or cleared to facilitate controlling individual interrupts without affecting others. This is accomplished by writing to the (IER) after bit 7 set or cleared to, in turn, set or clear selected enable bits. 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.

Selected bits in the IER can be set by writing to the IER with bit 7 in the data word set to a 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 contents of this register can be read at any time. Bit 7 will be read as a logic 1, however.

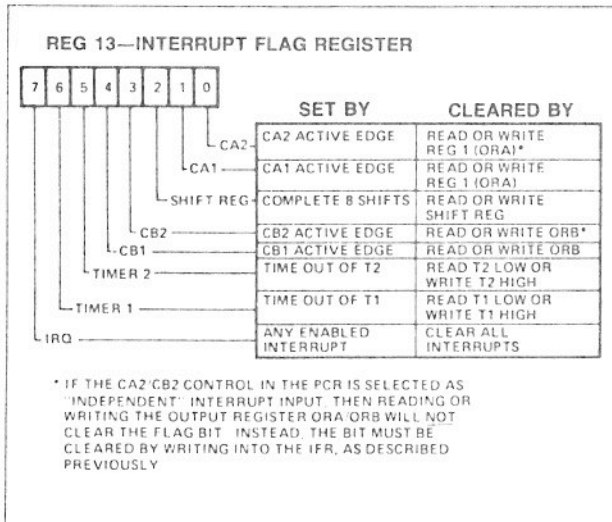


Figure 29. Interrupt Flag Register (IFR)

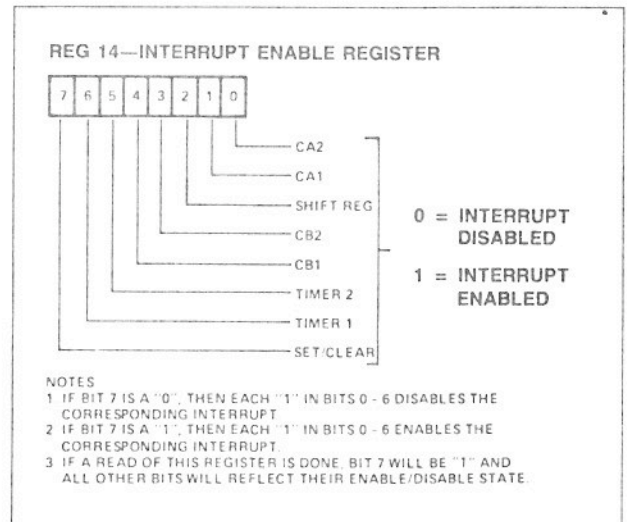


Figure 30. Interrupt Enable Register (IER)

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 | 31a, 31b |
| t_{RS1} | Delay Time, Clock Negative Transition to CA2 Positive Transition (pulse mode) | — | 1.0 | μs | 31a |
| t_{RS2} | Delay Time, CA1 Active Transition to CA2 Positive Transition (handshake mode) | — | 2.0 | μs | 31b |
| t_{WHS} | Delay Time, Clock Positive Transition to CA2 or CB2 Negative Transition (write handshake) | 0.05 | 1.0 | μs | 31c, 31d |
| t_{DS} | Delay Time, Peripheral Data Valid to CB2 Negative Transition | 0.20 | 1.5 | μs | 31c, 31d |
| t_{RS3} | Delay Time, Clock Positive Transition to CA2 or CB2 Positive Transition (pulse mode) | — | 1.0 | μs | 31c |
| t_{RS4} | Delay Time, CA1 or CB1 Active Transition to CA2 or CB2 Positive Transition (handshake mode) | — | 2.0 | μs | 31d |
| t_{21} | Delay Time Required from CA2 Output to CA1 Active Transition (handshake mode) | 400 | — | ns | 31d |
| t_{IL} | Setup Time, Peripheral Data Valid to CA1 or CB1 Active Transition (input latching) | 300 | — | ns | 31e |
| t_{AL} | CA1, CB1 Setup Prior to Transition to Arm Latch | 300 | — | ns | 31e |
| t_{PDH} | Peripheral Data Hold After CA1, CB1 Transition | 150 | — | ns | 31e |
| t_{SR1} | Shift-Out Delay Time — Time from ϕ_2 Falling Edge to CB2 Data Out | — | 300 | ns | 31f |
| t_{SR2} | Shift-In Setup Time — Time from CB2 Data In to ϕ_2 Rising Edge | 300 | — | ns | 31g |
| t_{SR3} | External Shift Clock (CB1) Setup Time Relative to ϕ_2 Trailing Edge | 100 | T_{CY} | ns | 31g |
| t_{IPW} | Pulse Width — PB6 Input Pulse | $2 \times T_{CY}$ | — | | 31i |
| t_{ICW} | Pulse Width — CB1 Input Clock | $2 \times T_{CY}$ | — | | 31h |
| t_{IPS} | Pulse Spacing — PB6 Input Pulse | $2 \times T_{CY}$ | — | | 31i |
| t_{ICS} | Pulse Spacing — CB1 Input Pulse | $2 \times T_{CY}$ | — | | 31h |

PERIPHERAL INTERFACE WAVEFORMS

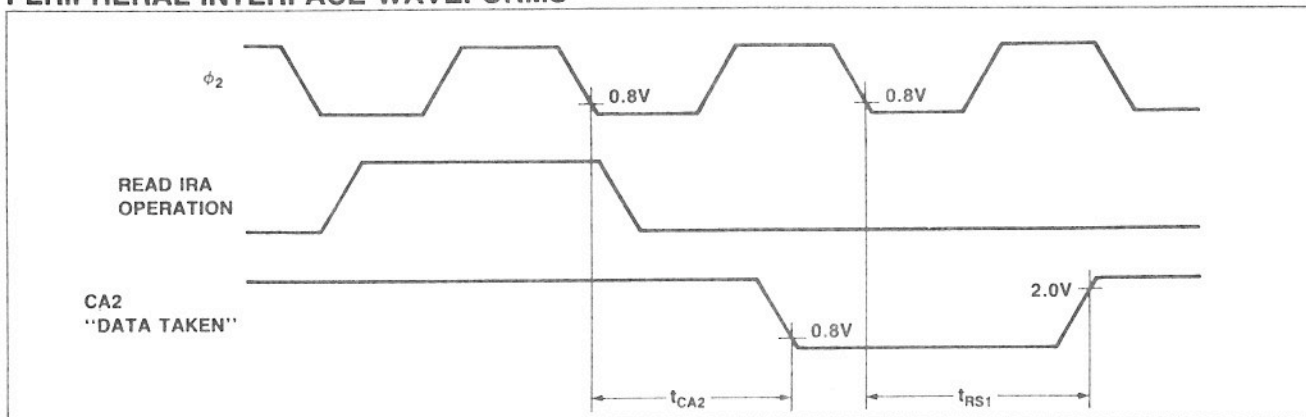


Figure 31a. CA2 Timing for Read Handshake, Pulse Mode

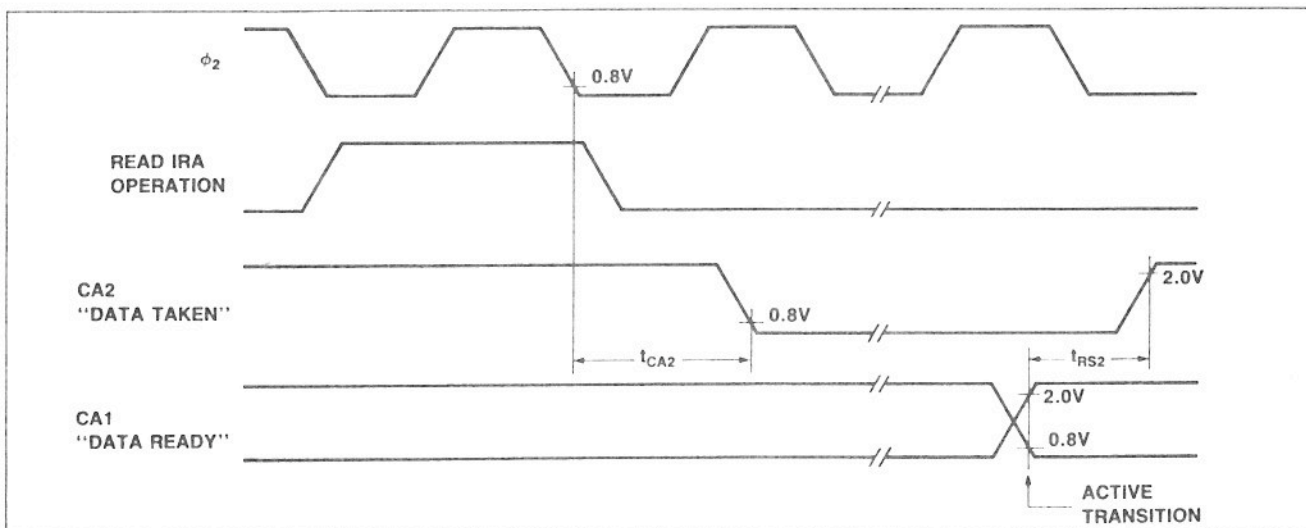


Figure 31b. CA2 Timing for Read Handshake, Handshake Mode

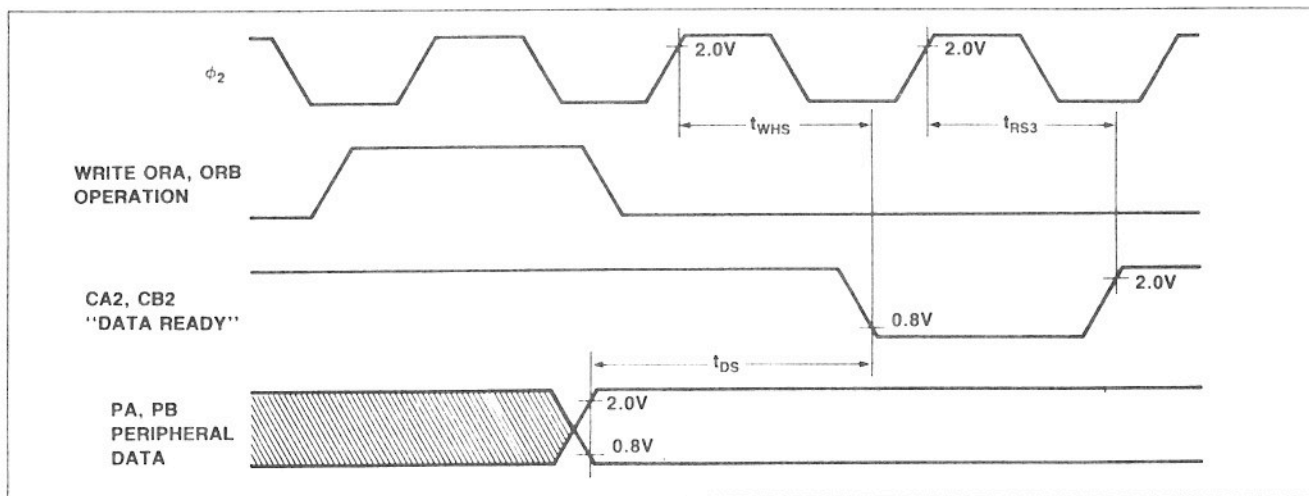


Figure 31c. CA2, CB2 Timing for Write Handshake, Pulse Mode

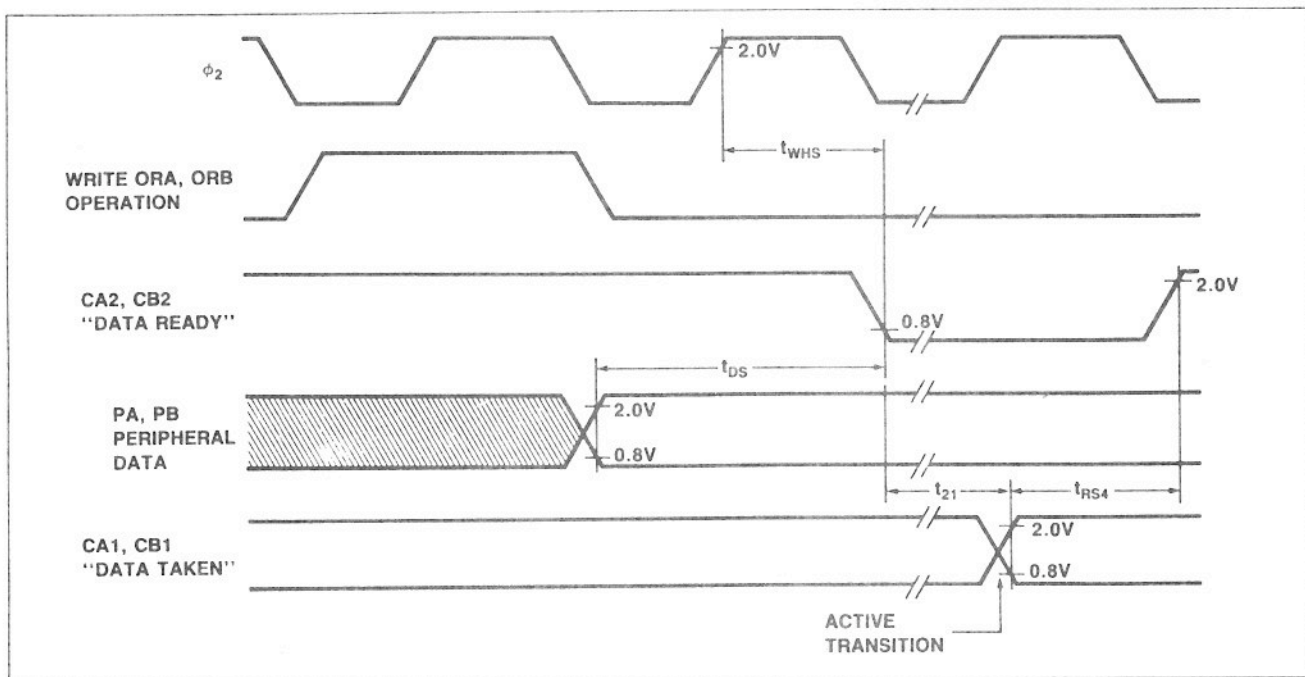


Figure 31d. CA2, CB2 Timing for Write Handshake, Handshake Mode

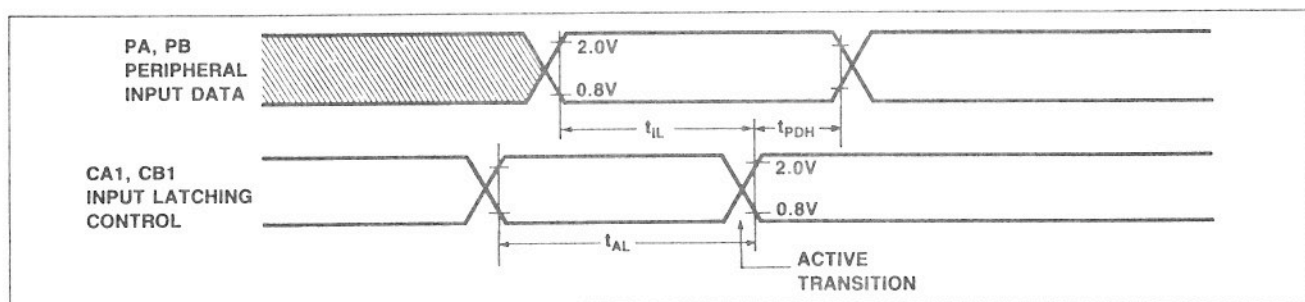


Figure 31e. Peripheral Data Input Latching Timing

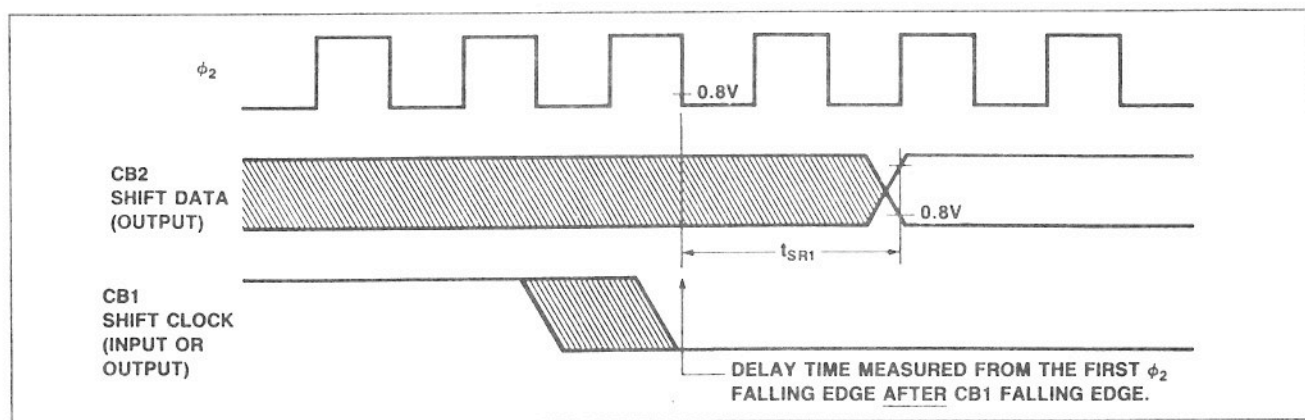


Figure 31f. Timing for Shift Out with Internal or External Shift Clocking

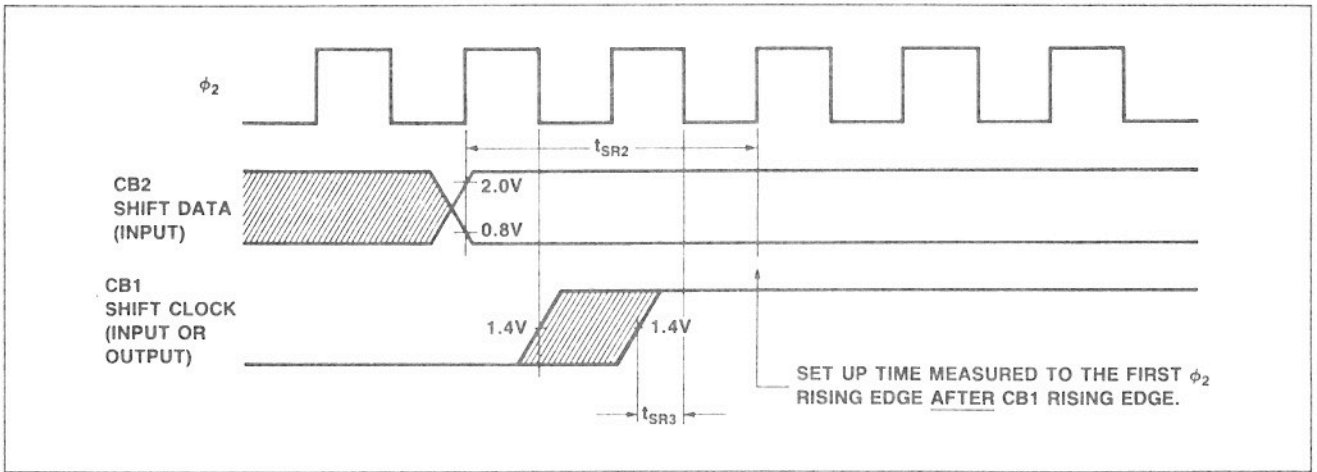


Figure 31g. Timing for Shift In with Internal or External Shift Clocking

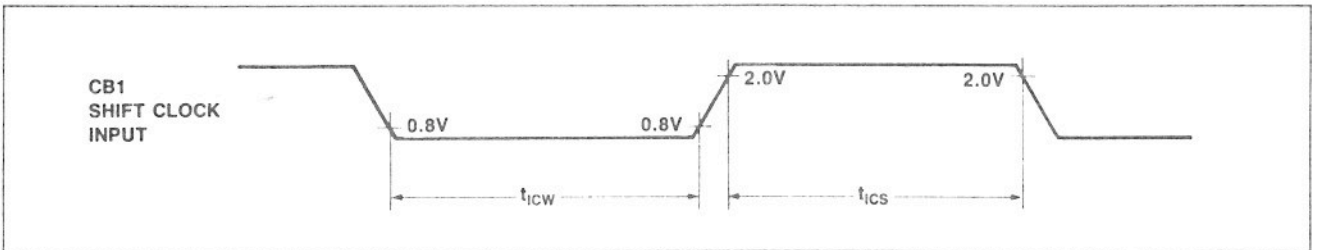


Figure 31h. External Shift Clock Timing

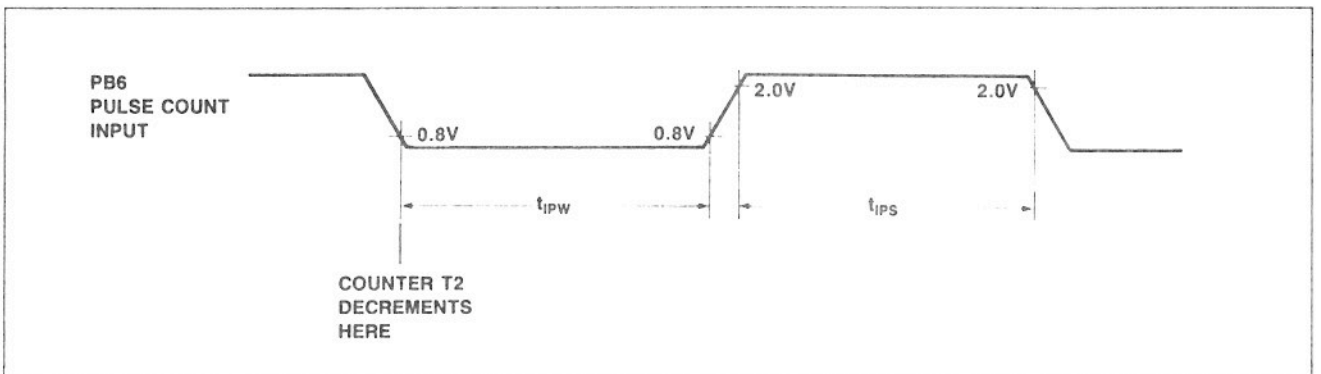


Figure 31i. Pulse Count Input Timing

Klokhuis Vierlingkaart

R6522
Versatile Interface Adapter (VIA)
BUS TIMING CHARACTERISTICS

| Parameter | Symbol | R6522 (1 MHz) | | R6522A (2 MHz) | | Unit |
|-----------|--------|---------------|------|----------------|------|------|
| | | Min. | Max. | Min. | Max. | |

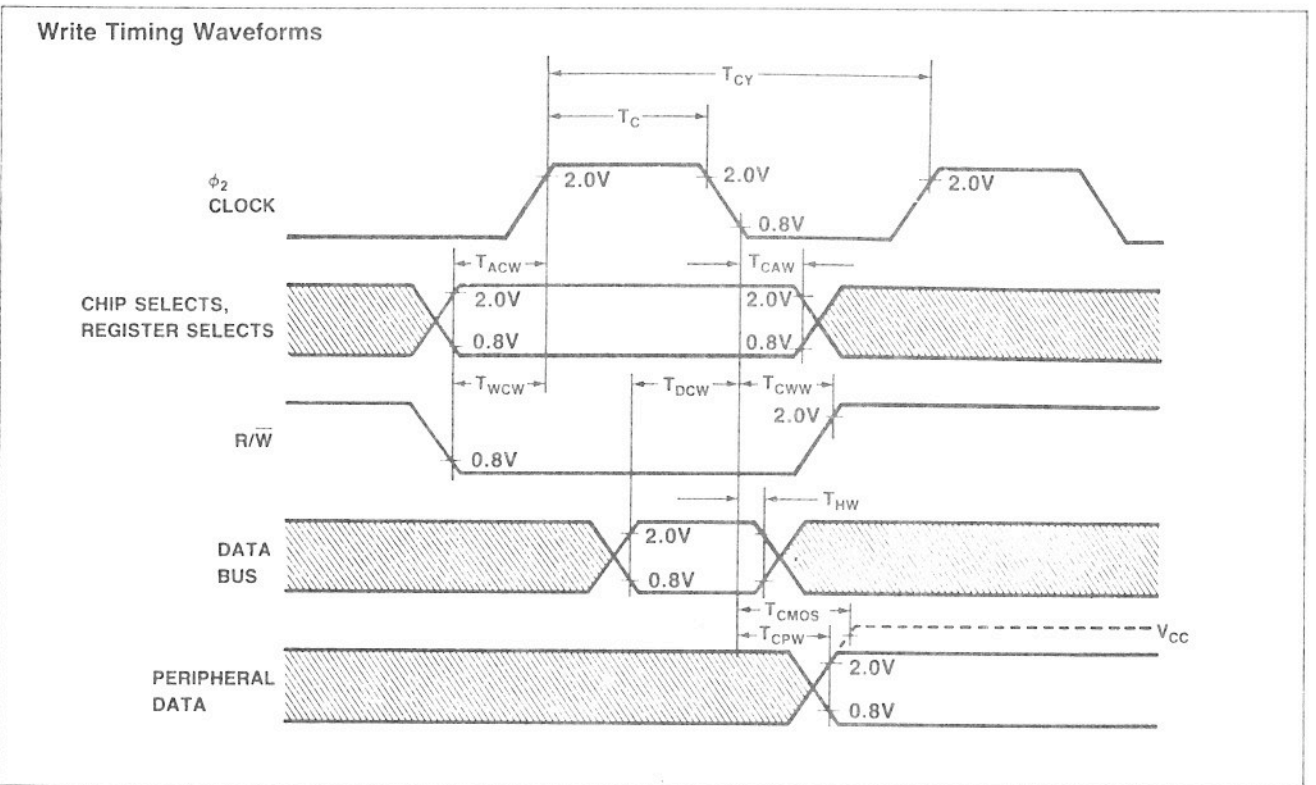
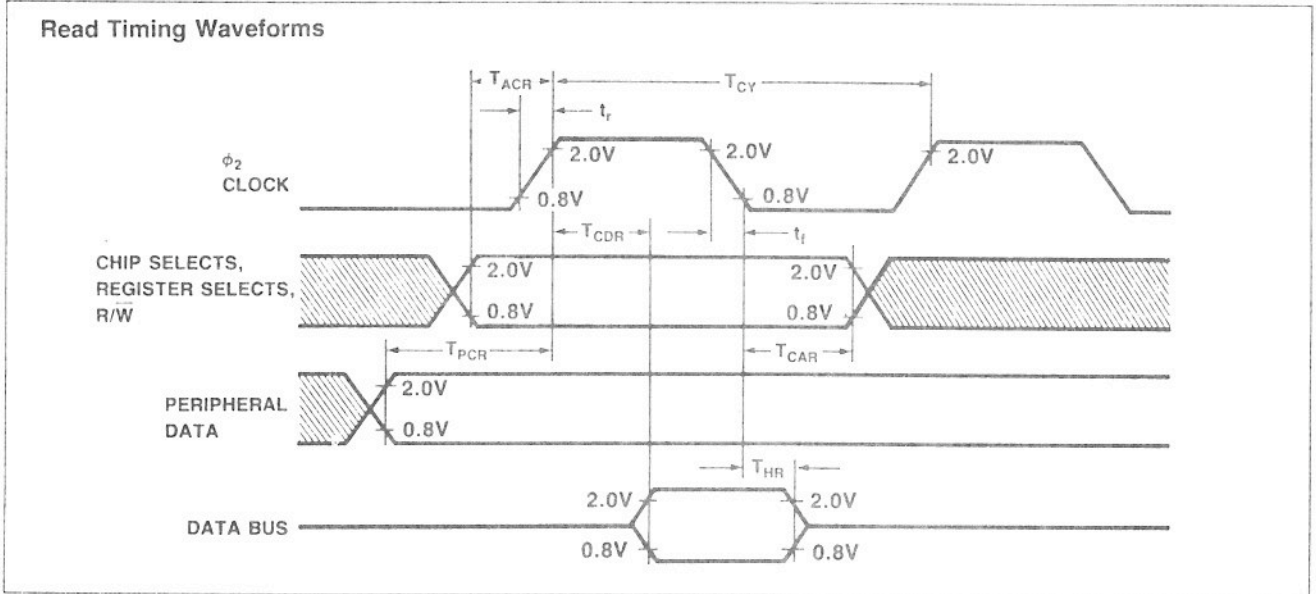
READ TIMING

| | | | | | | |
|-----------------------------|-----------|-----|-----|-----|-----|---------|
| Cycle Time | T_{CY} | 1 | 10 | 0.5 | 10 | μ S |
| Address Set-Up Time | T_{ACR} | 180 | — | 90 | — | ns |
| Address Hold Time | T_{CAR} | 0 | — | 0 | — | ns |
| Peripheral Data Set-Up Time | T_{PCR} | 300 | — | 150 | — | ns |
| Data Bus Delay Time | T_{CDR} | — | 365 | — | 190 | ns |
| Data Bus Hold Time | T_{HR} | 10 | — | 10 | — | ns |

WRITE TIMING

| | | | | | | |
|---|------------|-----|-----|------|-----|---------|
| Cycle Time | T_{CY} | 1 | 10 | 0.50 | 10 | μ S |
| ϕ 2 Pulse width | T_C | 470 | — | 235 | — | ns |
| Address Set-Up Time | T_{ACW} | 180 | — | 90 | — | ns |
| Address Hold Time | T_{CAW} | 0 | — | 0 | — | ns |
| R/W Set-Up Time | T_{WCW} | 180 | — | 90 | — | ns |
| R/W Hold Time | T_{CWW} | 0 | — | 0 | — | ns |
| Data Bus Set-Up Time | T_{DCW} | 200 | — | 90 | — | ns |
| Data Bus Hold Time | T_{HW} | 10 | — | 10 | — | ns |
| Peripheral Data Delay Time | T_{CPW} | — | 1.0 | — | 0.5 | μ S |
| Peripheral Data Delay Time to CMOS Levels | T_{CMOS} | — | 2.0 | — | 1.0 | μ S |
| Note: t_R and t_F = 10 to 30 ns. | | | | | | |

BUS TIMING WAVEFORMS



R6522

Versatile Interface Adapter (VIA)

ABSOLUTE MAXIMUM RATINGS*

| Parameter | 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 Commercial Industrial | T _A | 0 to +70 -40 to +85 | °C °C |
| Storage Temperature | T _{STG} | -55 to +150 | °C |

*NOTE: 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 other sections of this document is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

OPERATING CONDITIONS

| Parameter | Symbol | Value |
|---------------------------------|-----------------|-------------|
| Supply Voltage | V _{CC} | 5V ± 5% |
| Temperature Range Commercial | T _A | 0°C to 70°C |

DC CHARACTERISTICS

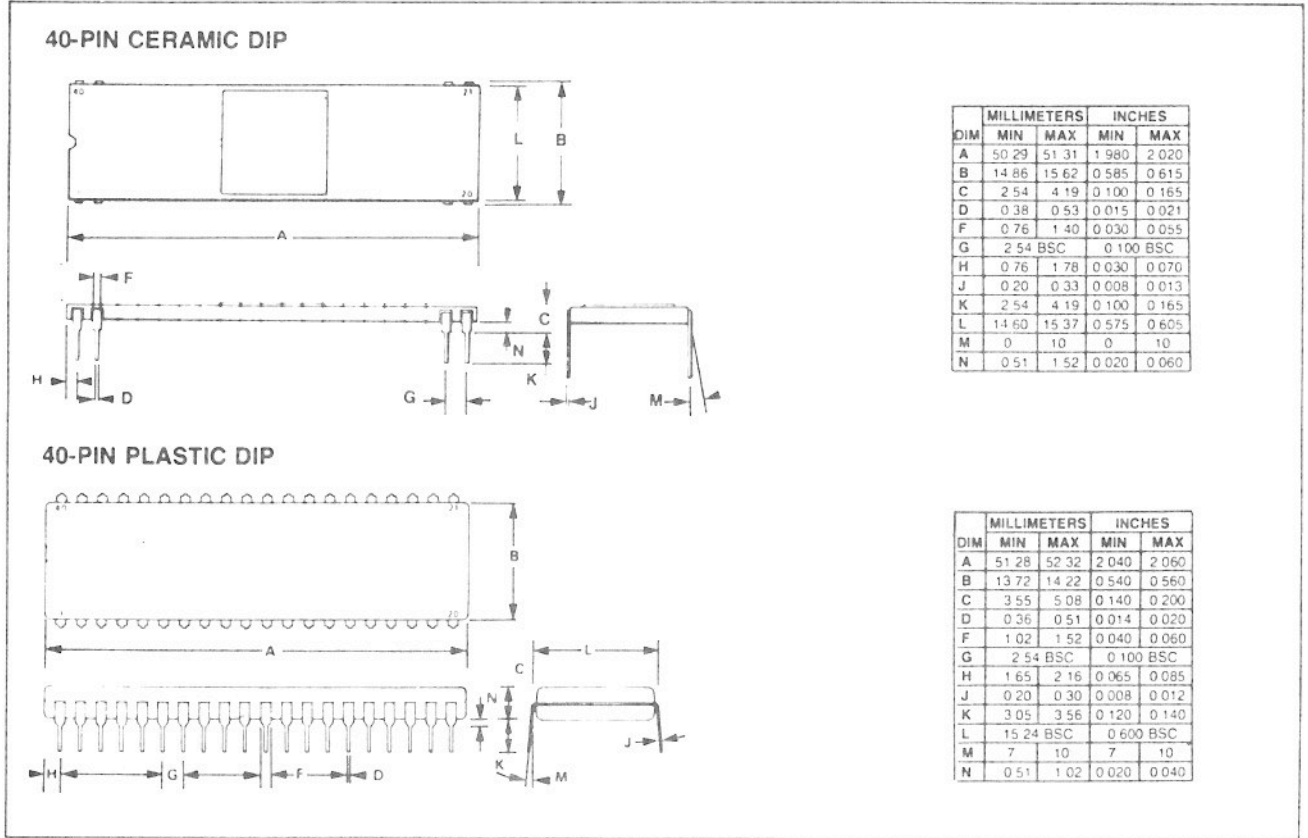
(V_{CC} = 5.0 Vdc ± 5%, V_{SS} = 0, T_A = T_L to T_H, unless otherwise noted)

| Parameter | Symbol | Min. | Typ. ³ | Max. | Unit | Test Conditions |
|---|------------------|--------------|-------------------|-----------------|----------------|---|
| Input High Voltage | V _{IH} | 2.4 | — | V _{CC} | V | |
| Input Low Voltage | V _{IL} | -0.3 | — | 0.4 | V | |
| Input Leakage Current R/W, RES, RS0, RS1, RS2, RS3, CS1, CS2, CA1, Ø2 | I _{IN} | — | ± 1 | ± 2.5 | µA | V _{IN} = 0V to 5.25V V _{CC} = 0V |
| Input Leakage Current for Three-State Off D0-D07 | I _{TSI} | — | ± 2 | ± 10 | µA | V _{IN} = 0.4V to 2.4V V _{CC} = 5.25V |
| Input High Current PA0-PA7, CA2, PB0-PB7, CB1, CBS | I _{IH} | -100 | -200 | — | µA | V _{IN} = 2.4V V _{CC} = 5.25V |
| Input Low Current PA0-PA7, CA2, PB0-PB7, CB1, CB2 | I _{IL} | — | -0.9 | -1.8 | mA | V _{IL} = 0.4V V _{CC} = 5.25V |
| Output High Voltage All outputs PB0-PB7, CB2 (Darlington Drive) | V _{OH} | 2.4 1.5 | — — | — — | V V | V _{CC} = 4.75V I _{LOAD} = -100 µA I _{LOAD} = -1.0 mA |
| Output Low Voltage | V _{OL} | — | — | 0.4 | V | V _{CC} = 4.75V I _{LOAD} = 1.6 mA |
| Output High Current (Sourcing) Logic PB0-PB7, CB2 (Darlington Drive) | I _{OH} | -100 -1.0 | -1000 -2.5 | — -10 | µA mA | V _{OH} = 2.4V V _{OH} = 1.5V |
| Output Low Current (Sinking) | I _{OL} | 1.6 | — | — | mA | V _{OL} = 0.4V |
| Output Leakage Current (Off State) IRQ | I _{OFF} | — | 4 | ± 10 | µA | V _{OH} = 2.4V V _{CC} = 5.25V |
| Power Dissipation | P _D | — | 450 | 700 | mW | |
| Input Capacitance R/W, RES, RS0, RS1, RS2, RS3, CS1, CS2, D0-D7, PA0-PA7, CA1, CA2, PB0-PB7 CB1, CB2 Ø2 Input | C _{IN} | — | — | 7 10 20 | pF pF pF | V _{CC} = 5.0V V _{IN} = 0V f = 1 MHz T _A = 25°C |
| Output Capacitance | C _{OUT} | — | — | 10 | pF | |

Notes:

1. All units are direct current (DC) except for capacitance.
2. Negative sign indicates outward current flow, positive indicates inward flow.
3. Typical values shown for V_{CC} = 5.0V and T_A = 25°C.

PACKAGE DIMENSIONS



Information furnished by Rockwell International Corporation is believed to be accurate and reliable. However, no responsibility is assumed by Rockwell International for its use, nor any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Rockwell International other than for circuitry embodied in a Rockwell product. Rockwell International reserves the right to change circuitry at any time without notice. This specification is subject to change without notice.

©Rockwell International Corporation 1984
All Rights Reserved

Printed in U.S.A

SEMICONDUCTOR PRODUCTS DIVISION REGIONAL ROCKWELL SALES OFFICES

| | | | |
|--|--|--|---|
| <p>HOME OFFICE Semiconductor Products Division Rockwell International 4311 Jamboree Road P O Box C, MS 501-300 Newport Beach, California 92658-8902 (714) 833-4700 TWX: 910 591-1698</p> <p>UNITED STATES Semiconductor Products Division Rockwell International 1842 Reynolds Irvine, California 92714 (714) 833-4655 TWX: 910 595-2518</p> <p>Semiconductor Products Division Rockwell International 3375 Scott Blvd., Suite 410 Santa Clara, California 95051 (408) 980-1900</p> <p>Semiconductor Products Division Rockwell International 921 Bowser Road Richardson, Texas 75080 (214) 996-6500 TLX: 73-307</p> | <p>Semiconductor Products Division Rockwell International 10700 West Higgins Rd., Suite 102 Rosemont, Illinois 60018 (312) 297-8862 TWX: 910 233-0179 (RI MED ROSM)</p> <p>Semiconductor Products Division Rockwell International 5001B Greentree Executive Campus, Rt. 73 Marlton, New Jersey 08053 (609) 596-0090 TWX: 710 940-1377</p> <p>FAR EAST Semiconductor Products Division Rockwell International Overseas Corp. Itohpa Hiraakawa-cho Bldg 7-6, 2-chome, Hiraakawa-cho Chiyoda-ku, Tokyo 102, Japan (03) 265-8806 TLX: J22198</p> <p>Rockwell Collins International Tai Sang Commercial Bldg., 11th Floor 24-34 Hennessy Rd. Hong Kong (5) 274-321 TLX: 74071 HK</p> | <p>EUROPE Semiconductor Products Division Rockwell International GmbH Fraunhoferstrasse 11 D-8033 Munchen-Martinsried West Germany (089) 857-6016 TLX: 0521/2650 rmd d</p> <p>Semiconductor Products Division Rockwell International Heathrow House, Bath Rd. Cranford, Hounslow, Middlesex, England (01) 759-2366 TLX: 851-25463</p> <p>Semiconductor Products Rockwell Collins Italiana S P A Via Boccaccio, 23 20123 Milano, Italy (02) 498 74 79 TLX: 316582 RCIMIL 1</p> | <p>YOUR LOCAL REPRESENTATIVE</p> |
|--|--|--|---|



R6551 ASYNCHRONOUS COMMUNICATIONS INTERFACE ADAPTER (ACIA)

DESCRIPTION

The Rockwell R6551 Asynchronous Communications Interface Adapter (ACIA) provides an easily implemented, program controlled interface between 8-bit microprocessor-based systems and serial communication data sets and modems.

The ACIA has an internal baud rate generator. This feature eliminates the need for multiple component support circuits, a crystal being the only other part required. The Transmitter baud rate can be selected under program control to be either 1 of 15 different rates from 50 to 19,200 baud, or at $1/16$ times an external clock rate. The Receiver baud rate may be selected under program control to be either the Transmitter rate, or at $1/16$ times the external clock rate. The ACIA has programmable word lengths of 5, 6, 7, or 8 bits; even, odd, or no parity; 1, $1\frac{1}{2}$, or 2 stop bits.

The ACIA is designed for maximum programmed control from the microprocessor (MPU), to simplify hardware implementation. Three separate registers permit the MPU to easily select the R6551's operating modes and data checking parameters and determine operational status.

The Command Register controls parity, receiver echo mode, transmitter interrupt control, the state of the \overline{RTS} line, receiver interrupt control, and the state of the \overline{DTR} line.

The Control Register controls the number of stop bits, word length, receiver clock source, and baud rate.

The Status Register indicates the states of the \overline{IRQ} , \overline{DSR} , and \overline{DCD} lines, Transmitter and Receiver Data Registers, and Overrun, Framing, and Parity Error conditions.

The Transmitter and Receiver Data Registers are used for temporary data storage by the ACIA Transmit and Receiver circuits.

ORDERING INFORMATION

| | |
|---------------------------------------|--|
| Part No.: R6551 | |
| Temperature Range (T_L to T_H): | |
| Blank = 0°C to +70°C | |
| E = -40°C to +85°C | |
| Frequency Range: | |
| 1 = 1 MHz | |
| 2 = 2 MHz | |
| Package: | |
| C = Ceramic | |
| P = Plastic | |

FEATURES

- Compatible with 8-bit microprocessors
- Full duplex operation with buffered receiver and transmitter
- Data set/modem control functions
- Internal baud rate generator with 15 programmable baud rates (50 to 19,200)
- Program-selectable internally or externally controlled receiver rate
- Programmable word lengths, number of stop bits, and parity bit generation and detection
- Programmable interrupt control
- Program reset
- Program-selectable serial echo mode
- Two chip selects
- 2 or 1 MHz operation
- 5.0 Vdc \pm 5% supply requirements
- 28-pin plastic or ceramic DIP
- Full TTL compatibility
- Compatible with R6500, R6500/* and R65C00 microprocessors

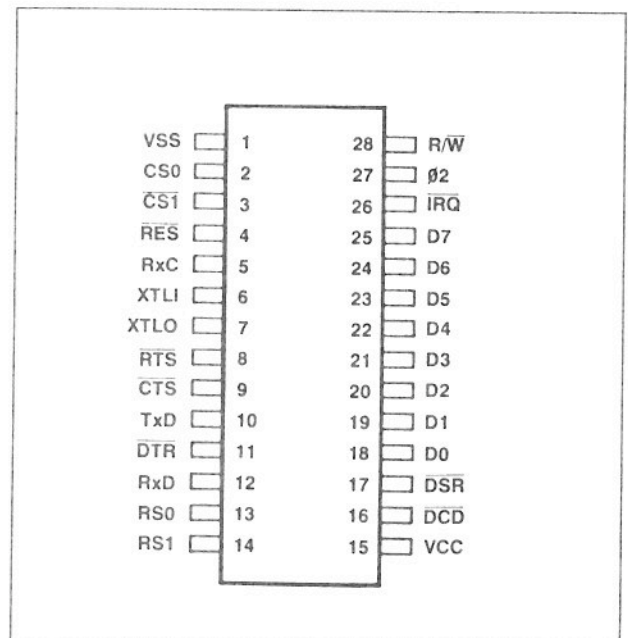


Figure 1. R6551 ACIA Pin Configuration

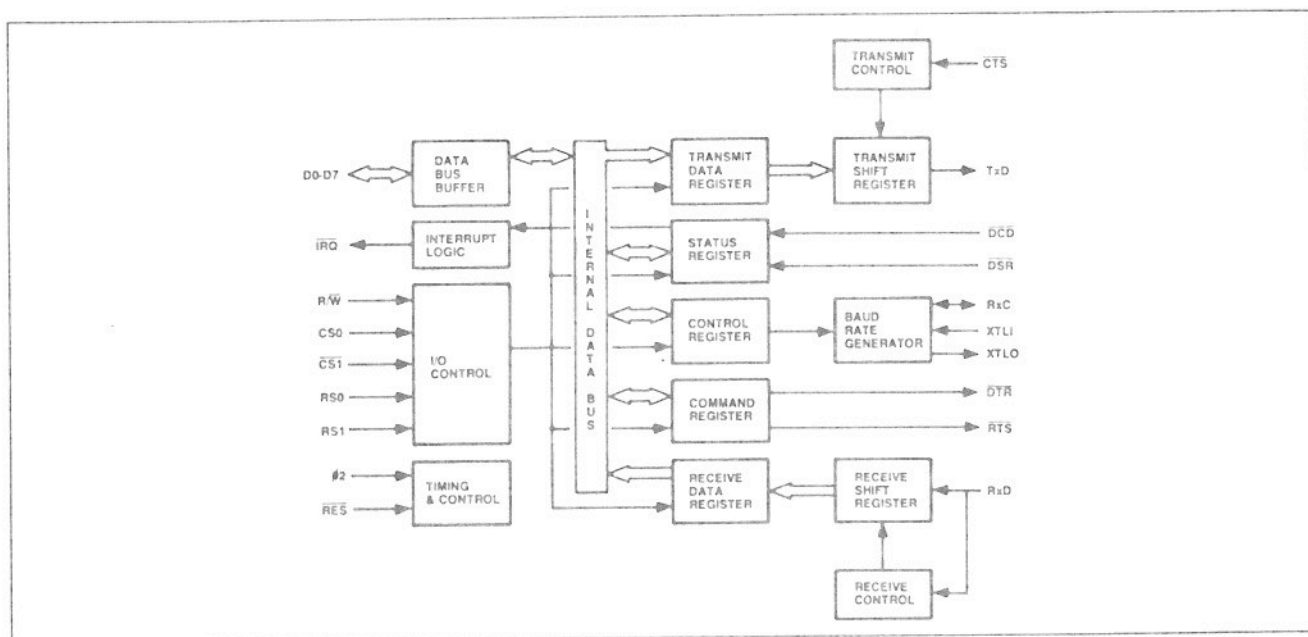


Figure 2. ACIA Internal Organization

FUNCTIONAL DESCRIPTION

A block diagram of the ACIA is presented in Figure 2 followed by a description of each functional element of the device.

DATA BUS BUFFERS

The Data Bus Buffer interfaces the system data lines to the internal data bus. The Data Bus Buffer is bi-directional. When the $\overline{R/W}$ line is high and the chip is selected, the Data Bus Buffer passes the data from the system data lines to the ACIA internal data bus. When the $\overline{R/W}$ line is low and the chip is selected, the Data Bus Buffer writes the data from the internal data bus to the system data bus.

INTERRUPT LOGIC

The Interrupt Logic will cause the \overline{IRQ} line to the microprocessor to go low when conditions are met that require the attention of the microprocessor. The conditions which can cause an interrupt will set bit 7 and the appropriate bit of bits 3 through 6 in the Status Register, if enabled. Bits 5 and 6 correspond to the Data Carrier Detect (\overline{DCD}) logic and the Data Set Ready (\overline{DSR}) logic. Bits 3 and 4 correspond to the Receiver Data Register full and the Transmitter Data Register empty conditions. These conditions can cause an interrupt request if enabled by the Command Register.

I/O CONTROL

The I/O Control Logic controls the selection of internal registers in preparation for a data transfer on the internal data bus and the direction of the transfer to or from the register.

The registers are selected by the Receiver Select ($RS1, RS0$) and Read/Write ($\overline{R/W}$) lines as described later in Table 1.

TIMING AND CONTROL

The Timing and Control logic controls the timing of data transfers on the internal data bus and the registers, the Data Bus Buffer, and the microprocessor data bus, and the hardware reset features.

Timing is controlled by the system $\phi 2$ clock input. The chip will perform data transfers to or from the microcomputer data bus during the $\phi 2$ high period when selected.

All registers will be initialized by the Timing and Control Logic when the Reset (\overline{RES}) line goes low. See the individual register description for the state of the registers following a hardware reset.

TRANSMITTER AND RECEIVER DATA REGISTERS

These registers are used as temporary data storage for the ACIA Transmitter and Receive Circuits. Both the Transmitter and Receiver are selected by a Register Select 0 ($RS0$) and Register Select 1 ($RS1$) low condition. The Read/Write ($\overline{R/W}$) line determines which actually uses the internal data bus; the Transmitter Data Register is write only and the Receiver Data Register is read only.

Bit 0 is the first bit to be transmitted from the Transmitter Data Register (least significant bit first). The higher order bits follow in order. Unused bits in this register are "don't care".

The Receiver Data Register holds the first received data bit in bit 0 (least significant bit first). Unused high-order bits are "0". Parity bits are not contained in the Receiver Data Register. They are stripped off after being used for parity checking.

R6551

Asynchronous Communications Interface Adapter (ACIA)

STATUS REGISTER

The Status Register indicates the state of interrupt conditions and other non-interrupt status lines. The interrupt conditions are the Data Set Ready, Data Carrier Detect, Transmitter Data Register Empty and Receiver Data Register Full as reported in bits 6 through 3, respectively. If any of these bits are set, the Interrupt (IRQ) indicator (bit 7) is also set. Overrun, Framing Error, and Parity Error are also reported (bits 2 through 0 respectively).

| | | | | | | | |
|-----|------------------|------------------|------|------|------|----|----|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| IRQ | \overline{DSR} | \overline{DCD} | TDRE | RDRE | OVRN | FE | PE |

- Bit 7 Interrupt (IRQ)**
0 No interrupt
1 Interrupt has occurred
- Bit 6 Data Set Ready (\overline{DSR})**
0 \overline{DSR} low (ready)
1 \overline{DSR} high (not ready)
- Bit 5 Data Carrier Detect (\overline{DCD})**
0 \overline{DCD} low (detected)
1 \overline{DCD} high (not detected)
- Bit 4 Transmitter Data Register Empty**
0 Not empty
1 Empty
- Bit 3 Receiver Data Register Full**
0 Not full
1 Full
- Bit 2 Overrun***
0 No overrun
1 Overrun has occurred
- Bit 1 Framing Error***
0 No framing error
1 Framing error detected
- Bit 0 Parity Error***
0 No parity error
1 Parity error detected

*No interrupt occurs for these conditions

Reset Initialization

| | | | | | | | | |
|---|---|---|---|---|---|---|---|----------------|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| 0 | — | — | 1 | 0 | 0 | 0 | 0 | Hardware reset |
| — | — | — | — | — | 0 | — | — | Program reset |

Parity Error (Bit 0), Framing Error (Bit 1), and Overrun (2)

None of these bits causes a processor interrupt to occur, but they are normally checked at the time the Receiver Data Register is read so that the validity of the data can be verified. These bits are self clearing (i.e., they are automatically cleared after a read of the Receiver Data Register).

Receiver Data Register Full (Bit 3)

This bit goes to a 1 when the ACIA transfers data from the Receiver Shift Register to the Receiver Data Register, and goes to a 0 (is cleared) when the processor reads the Receiver Data Register.

Transmitter Data Register Empty (Bit 4)

This bit goes to a 1 when the ACIA transfers data from the Transmitter Data Register to the Transmitter Shift Register, and goes to a 0 (is cleared) when the processor writes new data onto the Transmitter Data Register.

Data Carrier Detect (Bit 5) and Data Set Ready (Bit 6)

These bits reflect the levels of the \overline{DCD} and \overline{DSR} inputs to the ACIA. A 0 indicates a low level (true condition) and a 1 indicates a high level (false). Whenever either of these inputs change state, an immediate processor interrupt (IRQ) occurs, unless bit 1 of the Command Register (IRD) is set to a 1 to disable \overline{IRQ} . When the interrupt occurs, the status bits indicate the levels of the inputs immediately after the change of state occurred. Subsequent level changes will not affect the status bits until the Status Register is interrogated by the processor. At that time, another interrupt will immediately occur and the status bits reflect the new input levels. These bits are not automatically cleared (or reset) by an internal operation.

Interrupt (Bit 7)

This bit goes to a 1 whenever an interrupt condition occurs and goes to a 0 (is cleared) when the Status Register is read.

R6551

Asynchronous Communications Interface Adapter (ACIA)

CONTROL REGISTER

The Control Register selects the desired baud rate, frequency source, word length, and the number of stop bits.

| | | | | | | | |
|-----|-----|-----|-----|------|------|------|------|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SBN | WL | | RCS | SBR | | | |
| | WL1 | WL0 | | SBR3 | SBR2 | SBR1 | SBR0 |

Bit 7 Stop Bit Number (SBN)
 0 1 Stop bit
 1 2 Stop bits
 1 1/2 Stop bits
 For WL = 5 and no parity
 1 1 Stop bit
 For WL = 8 and parity

Bits 6-5 Word Length (WL)

| 6 | 5 | No. Bits |
|---|---|----------|
| 0 | 0 | 8 |
| 0 | 1 | 7 |
| 1 | 0 | 6 |
| 1 | 1 | 5 |

Bit 4 Receiver Clock Source (RCS)
 0 External receiver clock
 1 Baud rate

Bits 3-0 Selected Baud Rate (SBR)

| 3 | 2 | 1 | 0 | Baud |
|---|---|---|---|--------------------|
| 0 | 0 | 0 | 0 | 16x External Clock |
| 0 | 0 | 0 | 1 | 50 |
| 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 |

Reset Initialization

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| - | - | - | - | - | - | - | - |

Hardware reset (\overline{RES})
 Program reset

Selected Baud Rate (Bits 0, 1, 2, 3)

These bits select the Transmitter baud rate, which can be at 1/16 an external clock rate or one of 15 other rates controlled by the internal baud rate generator.

If the Receiver clock uses the same baud rate as the transmitter, then RxC becomes an output and can be used to slave other circuits to the ACIA. Figure 3 shows the Transmitter and Receiver layout.

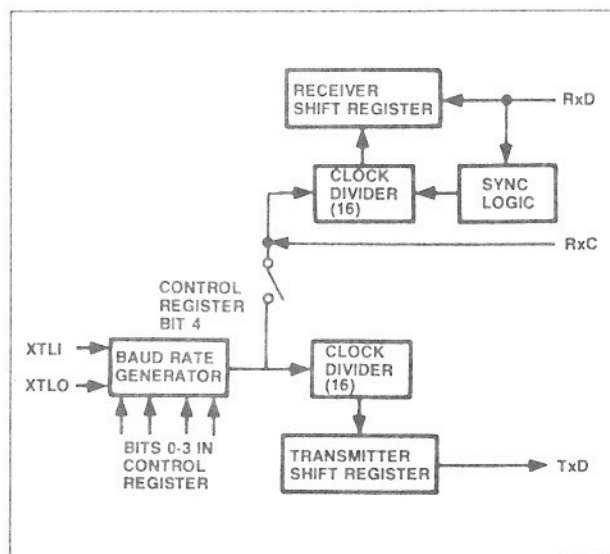


Figure 3. Transmitter/Receiver Clock Circuits

Receiver Clock Source (Bit 4)

This bit controls the clock source to the Receiver. A 0 causes the Receiver to operate at a baud rate of 1/16 an external clock. A 1 causes the Receiver to operate at the same baud rate as is selected for the transmitter.

Word Length (Bits 5, 6)

These bits determine the word length to be used (5, 6, 7 or 8 bits).

Stop Bit Number (Bit 7)

This bit determines the number of stop bits used. A 0 always indicates one stop bit. A 1 indicates 1/2 stop bits if the word length is 5 with no parity selected, 1 stop bit if the word length is 8 with parity selected, or 2 stop bits in all other configurations.

R6551

Asynchronous Communications Interface Adapter (ACIA)

COMMAND REGISTER

The Command Register controls specific modes and functions.

| | | | | | | | |
|------|------|-----|-----|------|------|-----|-----|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PMC | | PME | REM | TIC | | IRD | DTR |
| PMC1 | PMC0 | | | TIC1 | TIC0 | | |

Bits 7-6 Parity Mode Control (PMC)

| | | |
|---|---|---|
| 7 | 6 | |
| 0 | 0 | Odd parity transmitted/received |
| 0 | 1 | Even parity transmitted/received |
| 1 | 0 | Mark parity bit transmitted Parity check disabled |
| 1 | 1 | Space parity bit transmitted Parity check disabled |

Bit 5 Parity Mode Enabled (PME)

| | |
|---|--|
| 0 | Parity mode disabled No parity bit generated Parity check disabled |
| 1 | Parity mode enabled |

Bit 4 Receiver Echo Mode (REM)

| | |
|---|--|
| 0 | Receiver normal mode |
| 1 | Receiver echo mode bits 2 and 3 Must be zero for receiver echo mode, RTS will be low. |

Bits 3-2 Transmitter Interrupt Control (TIC)

| | | |
|---|---|--|
| 3 | 2 | |
| 0 | 0 | \overline{RTS} = High, transmit interrupt disabled |
| 0 | 1 | \overline{RTS} = Low, transmit interrupt enabled |
| 1 | 0 | \overline{RTS} = Low, transmit interrupt disabled |
| 1 | 1 | \overline{RTS} = Low, transmit interrupt disabled transmit break on TxD |

Bit 1 Interrupt Request Disabled (IRD)

| | |
|---|---------------------------|
| 0 | \overline{IRQ} enabled |
| 1 | \overline{IRQ} disabled |

Bit 0 Data Terminal Ready (DTR)

| | |
|---|------------------------------------|
| 0 | Data terminal not ready (DTR high) |
| 1 | Data terminal ready (DTR low) |

Data Terminal Ready (Bit 0)

This bit enables all selected interrupts and controls the state of the Data Terminal Ready (\overline{DTR}) line. A 0 indicates the microcomputer system is not ready by setting the \overline{DTR} line high. A 1 indicates the microcomputer system is ready by setting the \overline{DTR} line low.

Receiver Interrupt Control (Bit 1)

This bit disables the Receiver from generating an interrupt when set to a 1. The Receiver interrupt is enabled when this bit is set to a 0 and Bit 0 is set to a 1.

Transmitter Interrupt Control (Bits 2, 3)

These bits control the state of the Ready to Send (\overline{RTS}) line and the Transmitter interrupt.

Receiver Echo Mode (Bit 4)

A 1 enables the Receiver Echo Mode and a 0 enables the Receiver Echo Mode. When bit 4 is a 1, bits 2 and 3 must be 0. In the Receiver Echo Mode, the Transmitter returns each transmission received by the Receiver delayed by one-half bit time.

Parity Mode Enable (Bit 5)

This bit enables parity bit generation and checking. A 0 disables parity bit generation by the Transmitter and parity bit checking by the Receiver. A 1 bit enables generation and checking of parity bits.

Parity Mode Control (Bits 6, 7)

These bits determine the type of parity generated by the Transmitter, (even, odd, mark or space) and the type of parity check done by the Receiver (even, odd, or no check).

Reset Initialization

| | | | | | | | | |
|---|---|---|---|---|---|---|---|-------------------------------------|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Hardware reset (\overline{RES}) |
| - | - | - | 0 | 0 | 0 | 0 | 0 | Program reset |

R6551

Asynchronous Communications Interface Adapter (ACIA)

INTERFACE SIGNALS

Figure 4 shows the ACIA interface signals associated with the microprocessor and the modem.

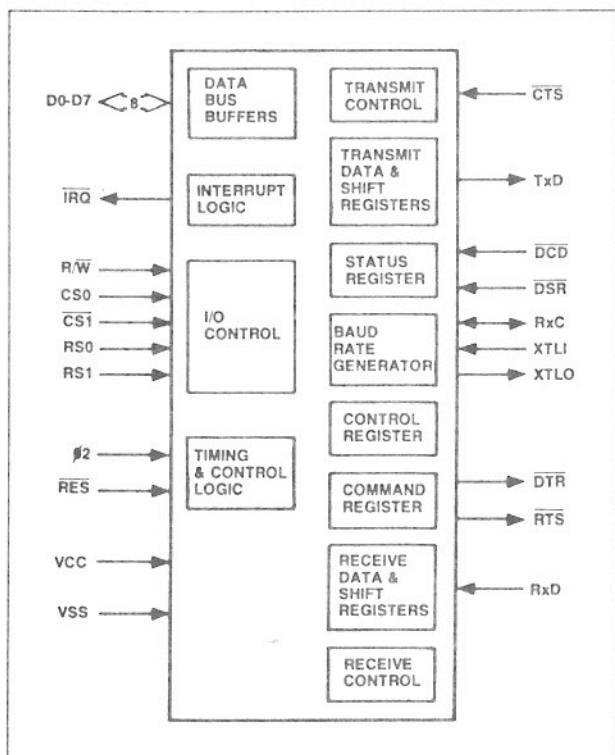


Figure 4. ACIA Interface Diagram

MICROPROCESSOR INTERFACE

Reset (\overline{RES})

During system initialization a low on the \overline{RES} input causes a hardware reset to occur. Upon reset, the Command Register and the Control Register are cleared (all bits set to 0). The Status Register is cleared with the exception of the indications of Data Set Ready and Data Carrier Detect, which are externally controlled by the DSR and DCD lines, and the transmitter Empty bit, which is set. \overline{RES} must be held low for one #2 clock cycle for a reset to occur.

Input Clock (#2)

The input clock is the system #2 clock and clocks all data transfers between the system microprocessor and the ACIA.

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

The R/\overline{W} input, generated by the microprocessor controls the direction of data transfers. A high on the R/\overline{W} pin allows the processor to read the data supplied by the ACIA, a low allows a write to the ACIA.

Interrupt Request (\overline{IRQ})

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

Data Bus (D0-D7)

The eight data line (D0-D7) pins transfer data between the processor and the ACIA. These lines are bi-directional and are normally high-impedance except during Read cycles when the ACIA is selected.

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

The two chip select inputs are normally connected to the processor address lines either directly or through decoders. The ACIA is selected when CS_0 is high and $\overline{CS_1}$ is low. When the ACIA is selected, the internal registers are addressed in accordance with the register select lines (RS_0 , RS_1).

Register Selects (RS_0 , RS_1)

The two register select lines are normally connected to the processor address lines to allow the processor to select the various ACIA internal registers. Table 1 shows the internal register select coding.

Table 1. ACIA Register Selection

| RS1 | RS0 | Register Operation | |
|-----|-----|---|-----------------------------|
| | | R/W = Low | R/W = High |
| L | L | Write Transmit Data Register | Read Receiver Data Register |
| L | H | Programmed Reset (Data is "Don't Care") | Read Status Register |
| H | L | Write Command Register | Read Command Register |
| H | H | Write Control Register | Read Control Register |

Only the Command and Control registers can both be read and written. The programmed Reset operation does not cause any data transfer, but is used to clear bits 4 through 0 in the Command register and bit 2 in the Status Register. The Control Register is unchanged by a programmed Reset. It should be noted that the programmed Reset is slightly different from the hardware Reset (\overline{RES}); refer to the register description.

R6551

Asynchronous Communications Interface Adapter (ACIA)

ACIA/MODEM INTERFACE

Crystal Pins (XTLI, XTLO)

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

Transmit Data (TxD)

The TxD output line transfers serial nonreturn-to-zero (NRZ) data to the modem. The least significant bit (LSB) of the Transmit Data Register is the first data bit transmitted and the rate of data transmission is determined by the baud rate selected or under control of an external clock. This selection is made by programming the Control Register.

Receive Data (RxD)

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

Receive Clock (RxC)

The RxC is a bi-directional pin which is 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.

Request to Send (RTS)

The RTS output pin controls the modem from the processor. The state of the RTS pin is determined by the contents of the Command Register.

Clear to Send (CTS)

The CTS input pin controls the transmitter operation. The enable state is with CTS low. The transmitter is automatically disabled if CTS is high.

Data Terminal Ready (DTR)

This output pin indicates the status of the ACIA to the modem. A low on DTR indicates the ACIA is enabled, a high indicates it is disabled. The processor controls this pin via bit 0 of the Command Register.

Data Set Ready (DSR)

The DSR input pin indicates to the ACIA the status of the modem. A low indicates the "ready" state and a high, "not-ready."

Data Carrier Detect (DCD)

The DCD input pin indicates to the ACIA 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.

TRANSMITTER AND RECEIVER OPERATION

Continuous Data Transmit

In the normal operating mode, the interrupt request output (IRQ) signals when the ACIA is ready to accept the next data word to be transmitted. This interrupt occurs at the beginning of the Start Bit. When the processor reads the Status Register of the ACIA, the interrupt is cleared.

The processor must then identify that the Transmit Data Register is ready to be loaded and must then load it with the next data word. This must occur before the end of the Stop Bit, otherwise a continuous "MARK" will be transmitted. Figure 5 shows the continuous Data Transmit timing relationship.

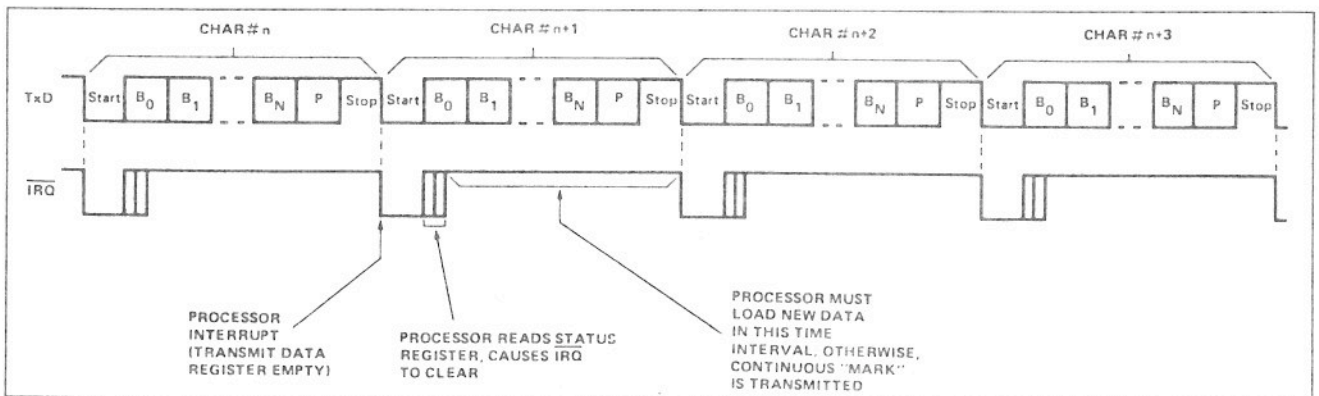


Figure 5. Continuous Data Transmit

R6551

Asynchronous Communications Interface Adapter (ACIA)

Continuous Data Receive

Similar to the Continuous Data Transmit case, the normal operation of this mode is to assert \overline{IRQ} when the ACIA has received a full data word. This occurs at about $9/16$ point through the Stop Bit. The processor must read the Status Register and

read the data word before the next interrupt, otherwise the Overrun condition occurs. Figure 6 shows the continuous Data Receive Timing Relationship.

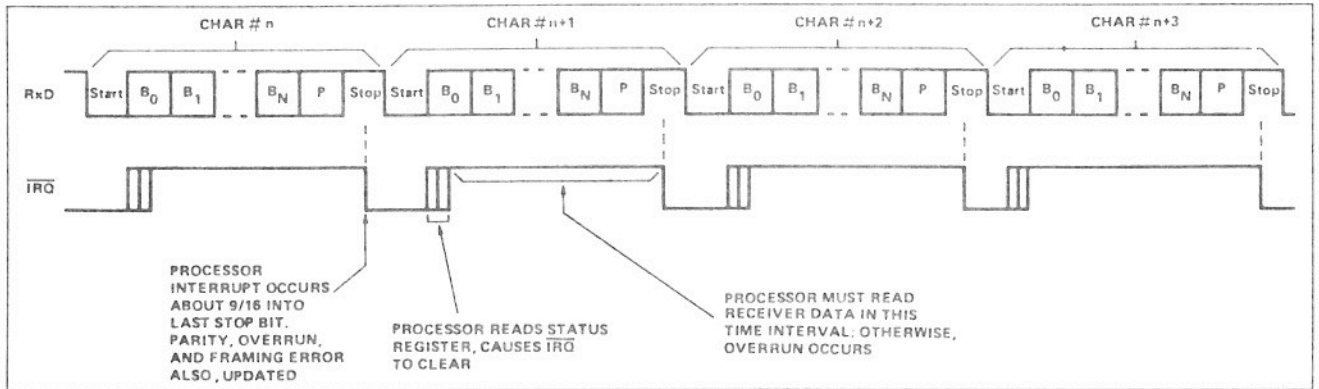


Figure 6. Continuous Data Receive

Transmit Data Register Not Loaded by Processor

If the processor is unable to load the Transmit Data Register in the allocated time, then the TxD line goes to the "MARK" condition until the data is loaded. \overline{IRQ} interrupts continue to occur at the same rate as previously, except no data is transmitted.

When the processor finally loads new data, a Start Bit immediately occurs, the data word transmission is started, and another interrupt is initiated, signaling for the next data word. Figure 7 shows the timing relationship for this mode of operation.

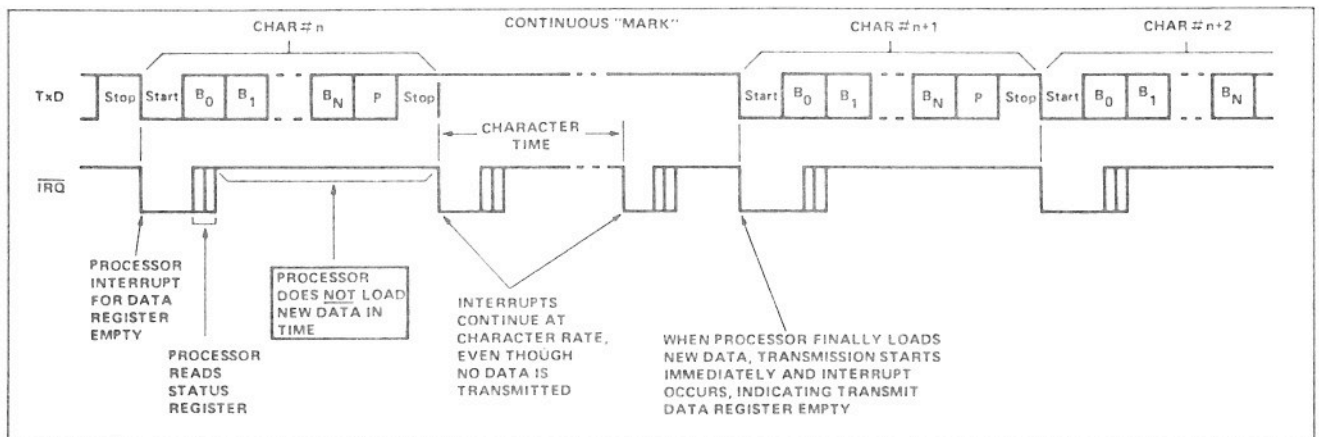


Figure 7. Transmit Data Register Not Loaded by Processor

R6551

Asynchronous Communications Interface Adapter (ACIA)

Effect of CTS on Transmitter

CTS is the Clear-to-Send signal generated by the modem. It is normally low (true state) but may go high in the event of some modem problems. When this occurs, the TxD line goes to the "MARK" condition after the entire last character (including parity and stop bit) have been transmitted. Bit 4 in the Status Register

indicates that the Transmitter Data Register is not empty and IRQ is not asserted. CTS is a transmit control line only, and has no effect on the ACIA Receiver Operation. Figure 8 shows the timing relationship for this mode of operation.

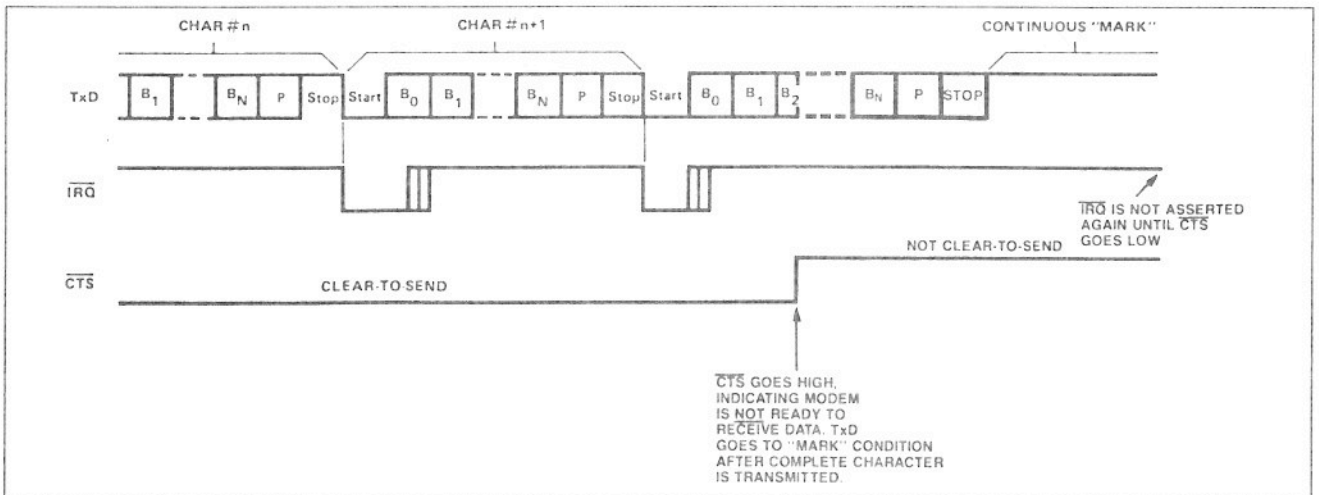


Figure 8. Effect of CTS on Transmitter

Effect of Overrun on Receiver

If the processor does not read the Receiver data Register in the allocated time, then, when the following interrupt occurs, the new data word is not transferred to the Receiver Data Register,

but the Overrun status bit is set. Thus, the Data Register will contain the last valid data word received and all following data is lost. Figure 9 shows the timing relationship for this mode.

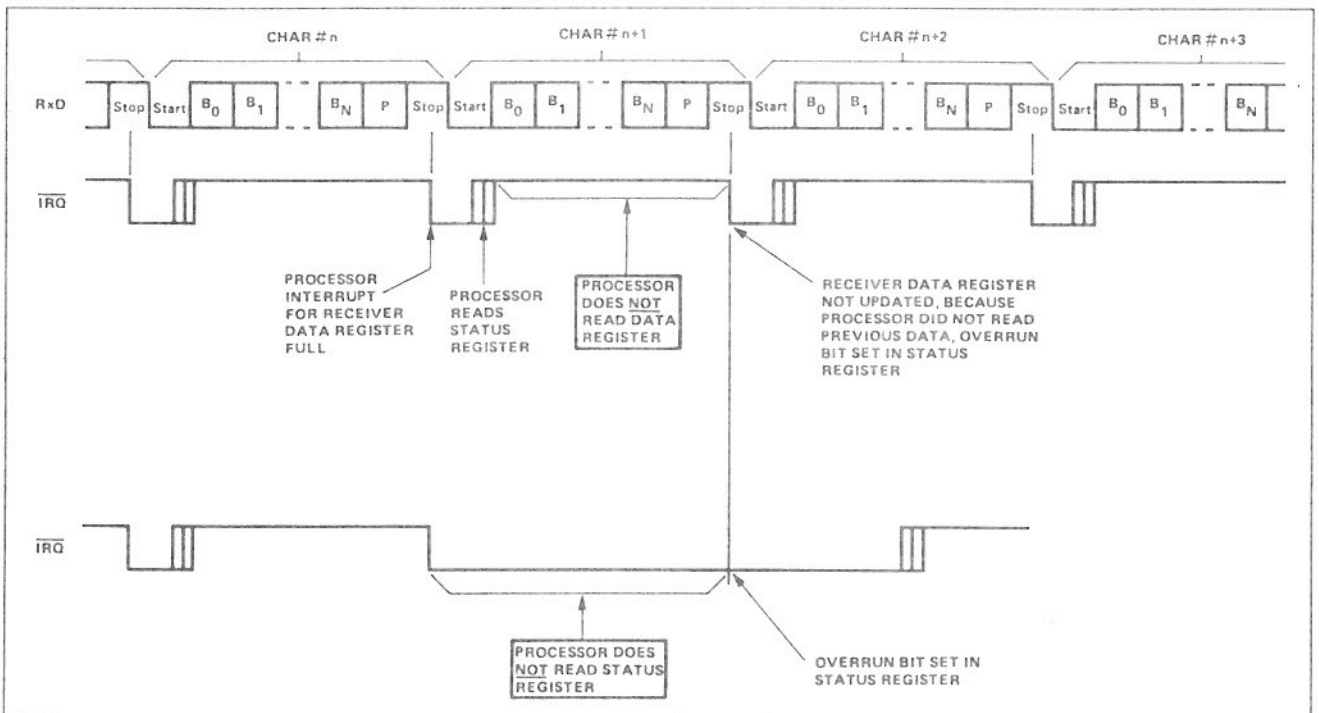


Figure 9. Effect of Overrun on Receiver

R6551

Asynchronous Communications Interface Adapter (ACIA)

Echo Mode Timing

In Echo Mode, the TxD line re-transmits the data on the RxD line, delayed by 1/2 of the bit time, as shown in Figure 10.

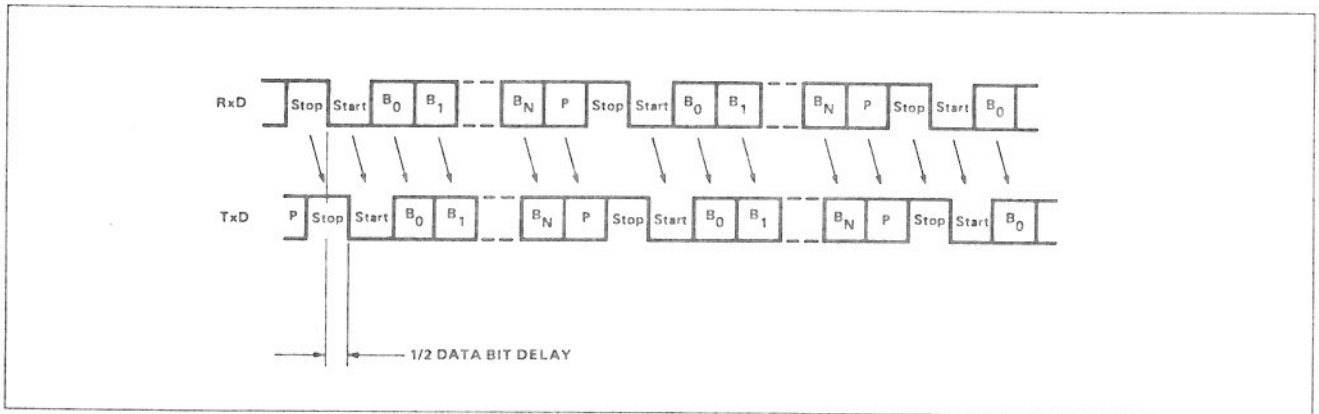


Figure 10. Echo Mode Timing

Effect of $\overline{\text{CTS}}$ on Echo Mode Operation

In Echo Mode, the Receiver operation is unaffected by $\overline{\text{CTS}}$, however, the Transmitter is affected when $\overline{\text{CTS}}$ goes high, i.e., the TxD line immediately goes to a continuous "MARK" condition. In this case, however, the Status Request indicates that

the Receiver Data Register is full in response to an $\overline{\text{IRQ}}$, so the processor has no way of knowing that the Transmitter has ceased to echo. See Figure 11 for the timing relationship of this mode.

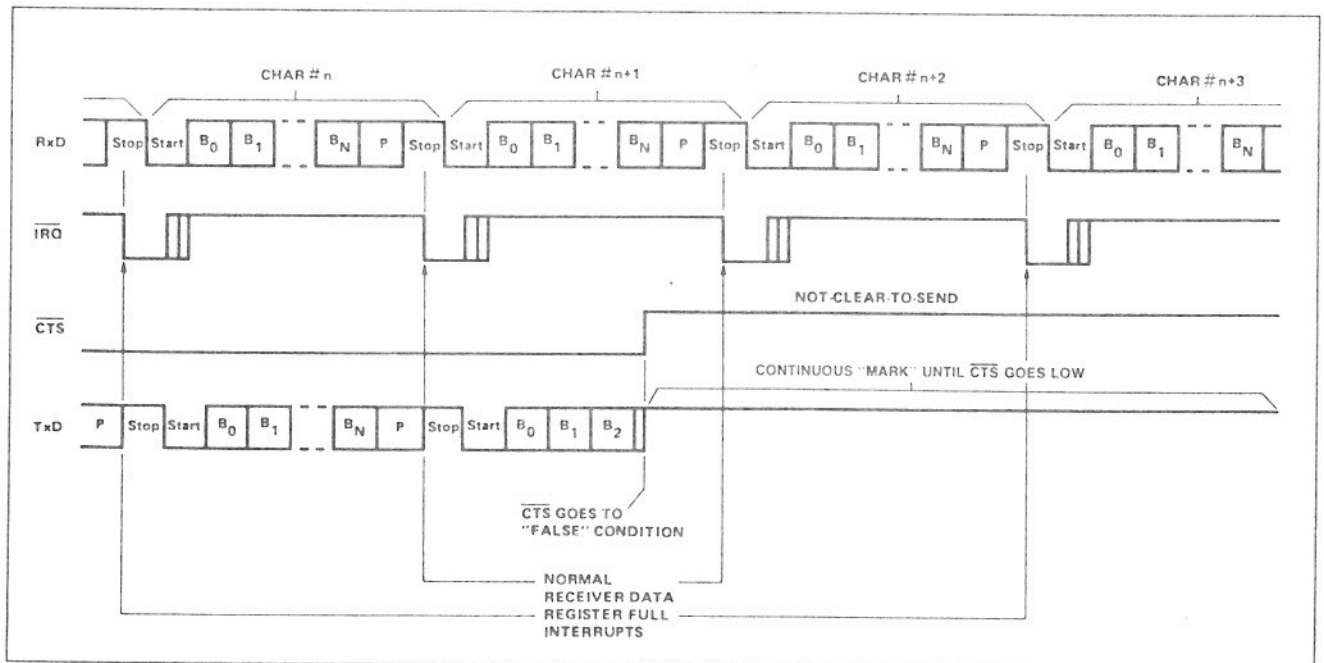


Figure 11. Effect of $\overline{\text{CTS}}$ on Echo Mode

R6551

Asynchronous Communications Interface Adapter (ACIA)

Overrun in Echo Mode

If Overrun occurs in Echo Mode, the Receiver is affected the same way as a normal overrun in Receive Mode. For the re-transmitted data, when overrun occurs, the TxD line goes to the

"MARK" condition until the first Start Bit after the Receiver Data Register is read by the processor. Figure 12 shows the timing relationship for this mode.

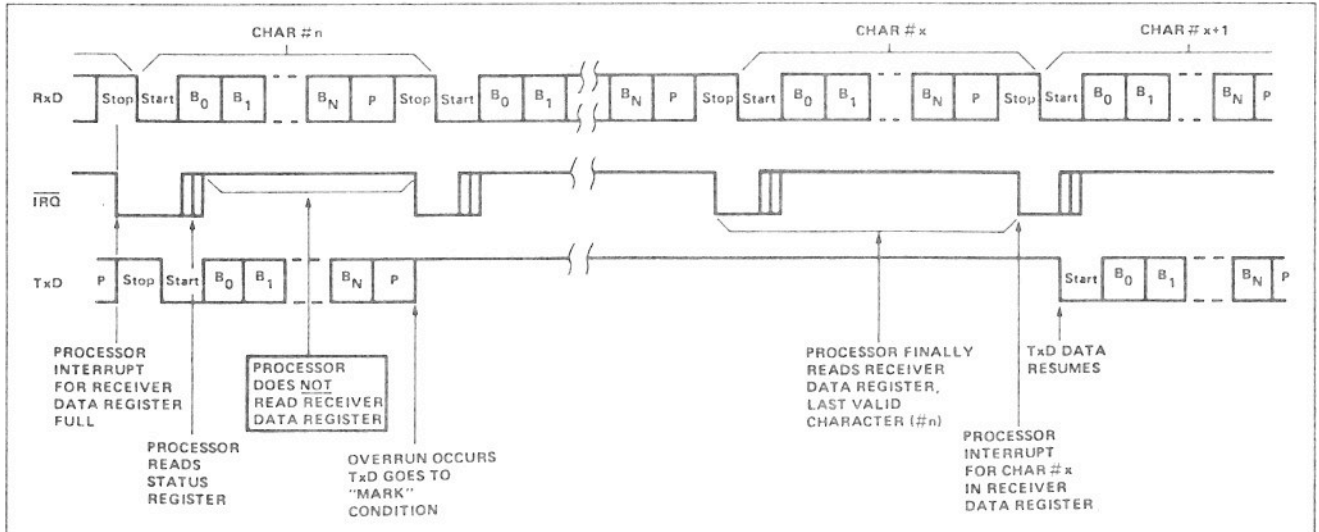


Figure 12. Overrun in Echo Mode

Framing Error

Framing Error is caused by the absence of Stop Bit(s) on received data. A Framing Error is indicated by the setting of bit 4 in the Status Register at the same time the Receiver Data Register Full bit is set, also in the Status Register. In response to $\overline{\text{IRQ}}$, generated by RDRF, the Status Register can also be

checked for the Framing Error. Subsequent data words are tested for Framing Error separately, so the status bit will always reflect the last data word received. See Figure 13 for Framing Error timing relationship.

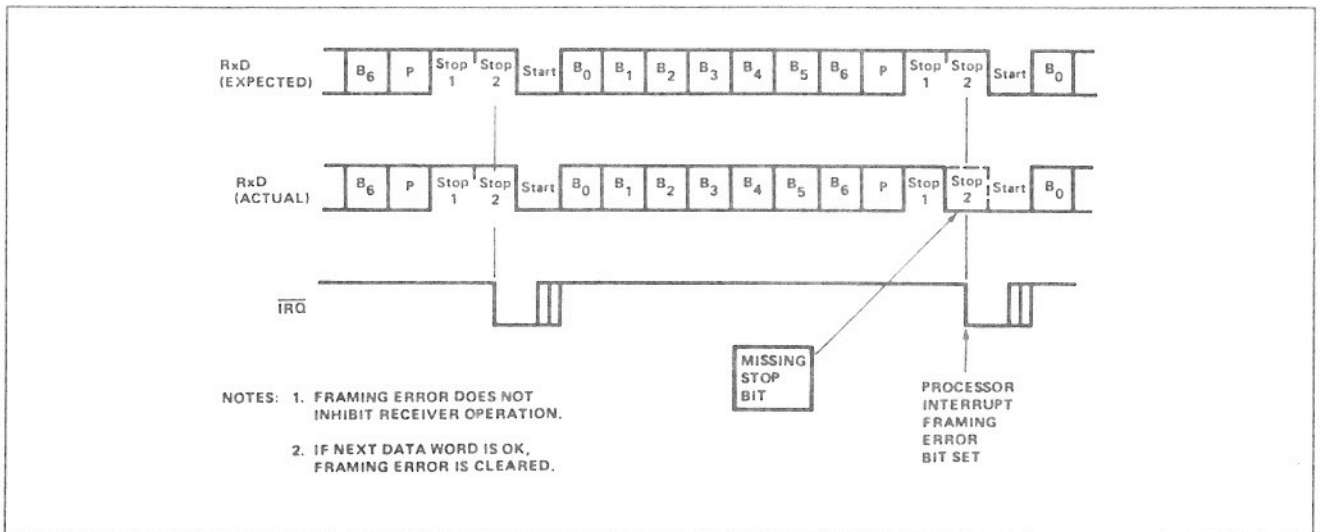


Figure 13. Framing Error

R6551

Asynchronous Communications Interface Adapter (ACIA)

Effect of \overline{DCD} on Receiver

\overline{DCD} is a modem output indicating the status of the carrier-frequency-detection circuit of the modem. This line goes high for a loss of carrier. Normally, when this occurs, the modem will stop transmitting data some time later. The ACIA asserts \overline{IRQ} whenever \overline{DCD} changes state and indicates this condition via bit 5 in the Status Register.

Once such a change of state occurs, subsequent transitions will not cause interrupts or changes in the Status Register until the first interrupt is serviced. When the Status Register is read by the processor, the ACIA automatically checks the level of the \overline{DCD} line, and if it has changed, another \overline{IRQ} occurs (see Figure 14).

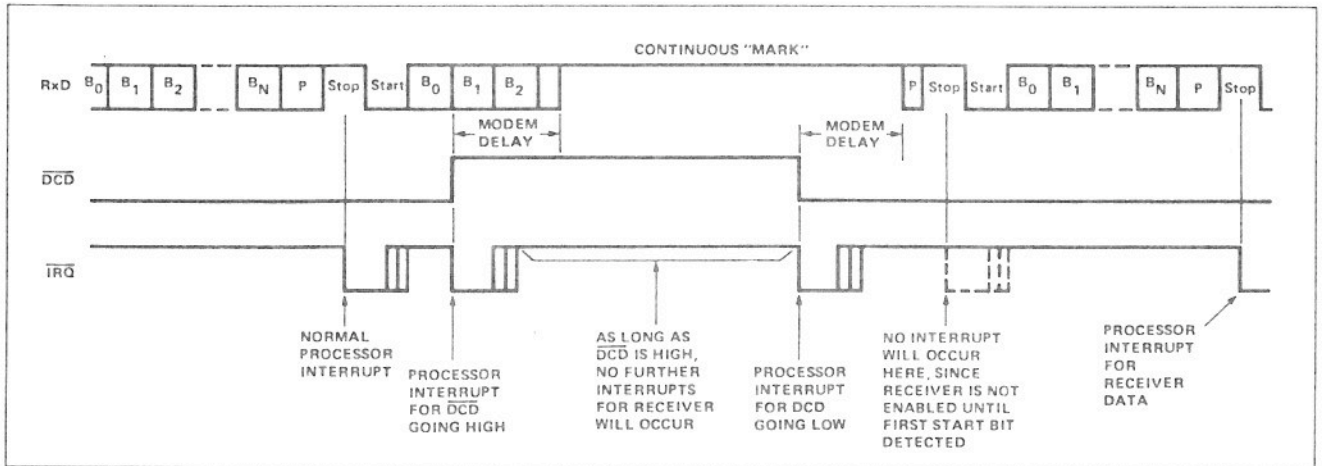


Figure 14. Effect of DCD on Receiver

Timing with 1/2 Stop Bits

It is possible to select 1/2 Stop Bits, but this occurs only for 5-bit data words with no parity bit. In this case, the \overline{IRQ} asserted for Receiver Data Register Full occurs halfway through the

trailing half-Stop Bit. Figure 15 shows the timing relationship for this mode.

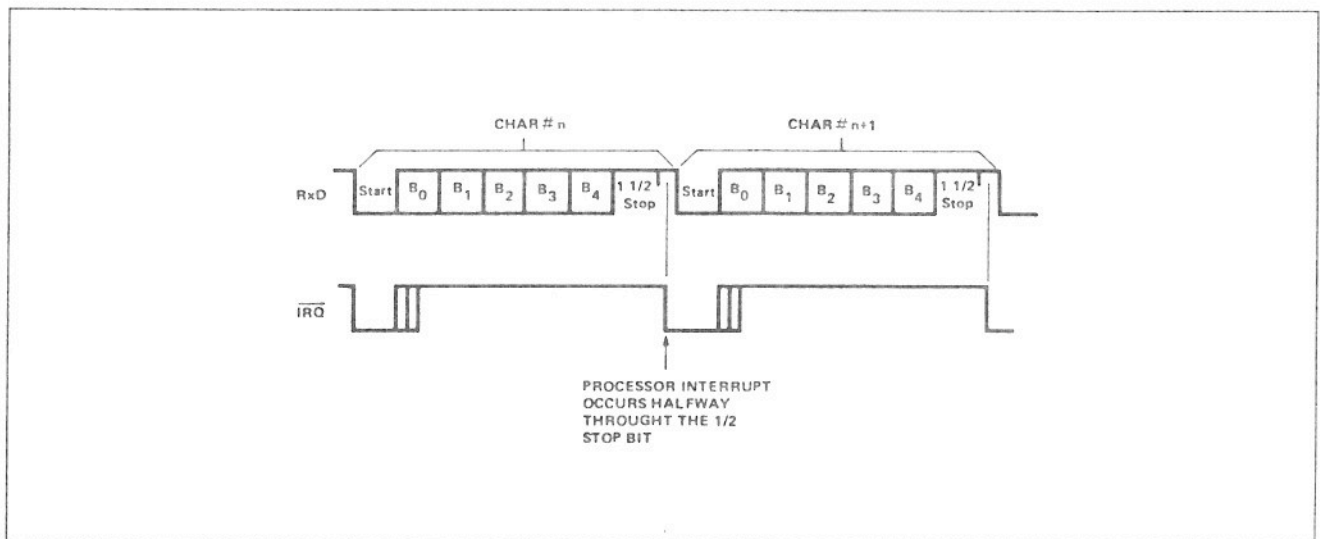


Figure 15. Timing with 1/2 Stop Bits

Transmit Continuous "BREAK"

This mode is selected via the ACIA Command Register and causes the Transmitter to send continuous "BREAK" characters, beginning with the next character transmitted. At least one full "BREAK" character will be transmitted, even if the processor quickly re-programs the Command Register transmit mode. Later, when the Command Register is programmed back to normal transmit mode, an immediate Stop Bit will be generated and transmission will resume. Figure 16 shows the timing relationship for this mode.

Note

If, while operating in the Transmit Continuous "BREAK" mode, the \overline{CTS} should go to a high, the TxD will be overridden by the \overline{CTS} and will go to continuous "MARK" at the beginning of the next character transmitted after the \overline{CTS} goes high.

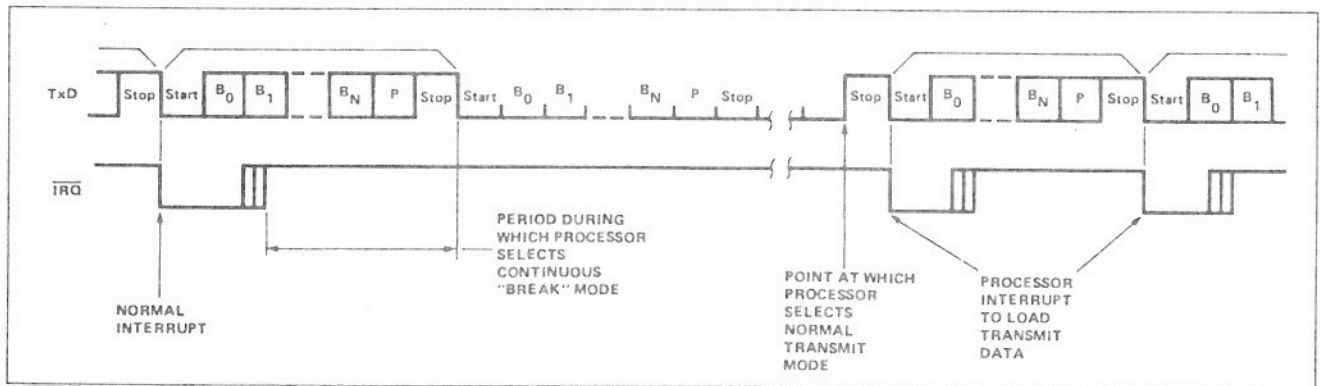


Figure 16. Transmit Continuous "BREAK"

Receive Continuous "BREAK"

In the event the modem transmits continuous "BREAK" characters, the ACIA will terminate receiving. Reception will resume only after a Stop Bit is encountered by the ACIA. Figure 17

shows the timing relationship for continuous "BREAK" characters.

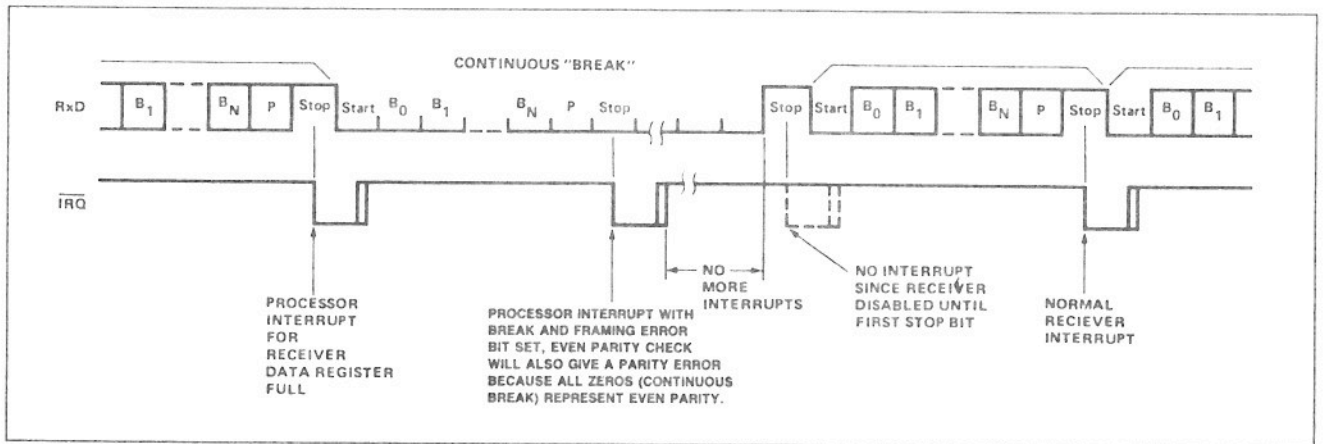


Figure 17. Receive Continuous "BREAK"

R6551**Asynchronous Communications Interface Adapter (ACIA)****STATUS REGISTER OPERATION**

Because of the special functions of the various status bits, there is a suggested sequence for checking them. When an interrupt occurs, the ACIA should be interrogated, as follows:

1. Read Status Register

This operation automatically clears Bit 7 (\overline{IRQ}). Subsequent transitions on \overline{DSR} and \overline{DCD} will cause another interrupt.

2. Check IRQ (Bit 7) in the data read from the Status Register

If not set, the interrupt source is not the ACIA.

3. Check \overline{DCD} and \overline{DSR}

These must be compared to their previous levels, which must have been saved by the processor. If they are both 0 (modem "on-line") and they are unchanged then the remaining bits must be checked.

4. Check RDRF (Bit 3)

Check for Receiver Data Register Full.

5. Check Parity, Overrun, and Framing Error (Bits 0-2) if the Receiver Data Register is full.

6. Check TDRE (Bit 4)

Check for Transmitter Data Register Empty.

7. If none of the above conditions exist, then \overline{CTS} must have gone to the false (high) state.**PROGRAM RESET OPERATION**

A program reset occurs when the processor performs a write operation to the ACIA with RS0 low and RS1 high. The program reset operates somewhat different from the hardware reset (\overline{RES} pin) and is described as follows:

1. Internal registers are not completely cleared. Check register formats for the effect of a program reset on internal registers.

2. The \overline{DTR} line goes high immediately.3. Receiver and transmitter interrupts are disabled immediately. If \overline{IRQ} is low when the reset occurs, it stays low until serviced, unless interrupt was caused by \overline{DCD} or \overline{DSR} transition.4. \overline{DCD} and \overline{DSR} interrupts are disabled immediately. If \overline{IRQ} is low and was caused by \overline{DCD} or \overline{DSR} , then it goes high, also \overline{DCD} and \overline{DSR} status bits subsequently will follow the input lines, although no interrupt will occur.

5. Overrun cleared, if set.

MISCELLANEOUS

1. If Echo Mode is selected, \overline{RTS} goes low.
2. If Bit 0 of Command Register is 0 (disabled), then:
 - a) All interrupts are disabled, including those caused by \overline{DCD} and \overline{DSR} transitions.
 - b) Transmitter is disabled immediately.
 - c) Receiver is disabled, but a character currently being received will be completed first.
3. Odd parity occurs when the sum of all the 1 bits in the data word (including the parity bit) is odd.
4. In the receive mode, the received parity bit does not go into the Receiver Data Register, but generates parity error or no parity error for the Status Register.
5. Transmitter and Receiver may be in full operation simultaneously. This is "full-duplex" mode.
6. If the RxD line inadvertently goes low and then high right after a Stop Bit, the ACIA does not interpret this as a Start Bit, but samples the line again halfway into the bit to determine if it is a true Start Bit or a false one. For false Start Bit detection, the ACIA does not begin to receive data, instead, only a true Start Bit initiates receiver operation.
7. Precautions to consider with the crystal oscillator circuit:
 - a) The external crystal should be a "series" mode crystal.
 - b) The XTALI input may be used as an external clock input. The unused pin (EXTALO) must be floating and may not be used for any other function.
8. \overline{DCD} and \overline{DSR} transitions, although causing immediate processor interrupts, have no effect on transmitter operation. Data will continue to be sent, unless the processor forces transmitter to turn off. Since these are high-impedance inputs, they must not be permitted to float (un-connected). If unused, they must be terminated either to GND or V_{CC} .

GENERATION OF NON-STANDARD BAUD RATES**Divisors**

The internal counter/divider circuit selects the appropriate divisor for the crystal frequency by means of bits 0-3 of the ACIA Control Register, as shown in Table 2.

Generating Other Baud Rates

By using a different crystal, other baud rates may be generated. These can be determined by:

$$\text{Baud Rate} = \frac{\text{Crystal Frequency}}{\text{Divisor}}$$

Furthermore, it is possible to drive the ACIA with an off-chip oscillator to achieve other baud rates. In this case, XTALI (pin 6) must be the clock input and XTALO (pin 7) must be a no-connect.

Table 2. Divisor Selection

| Control Register Bits | Divisor Selected For The Internal Counter | Baud Rate Generated With 1.8432 MHz Crtstal | Baud Rate Generated With a Crystal of Frequency (F) |
|-----------------------|---|--|---|
| 3 2 1 0 | | | |
| 0 0 0 0 | No Divisor Selected | 16 × External Clock at Pin RxC | 16 × External Clock at Pin RxC |
| 0 0 0 1 | 36,864 | $\frac{1.8432 \times 10^6}{36,864} = 50$ | $\frac{F}{36,864}$ |
| 0 0 1 0 | 24,576 | $\frac{1.8432 \times 10^6}{24,576} = 75$ | $\frac{F}{24,576}$ |
| 0 0 1 1 | 16,769 | $\frac{1.8432 \times 10^6}{16,769} = 109.92$ | $\frac{F}{16,769}$ |
| 0 1 0 0 | 13,704 | $\frac{1.8432 \times 10^6}{13,704} = 134.51$ | $\frac{F}{13,704}$ |
| 0 1 0 1 | 12,288 | $\frac{1.8432 \times 10^6}{12,288} = 150$ | $\frac{F}{12,288}$ |
| 0 1 1 0 | 6,144 | $\frac{1.8432 \times 10^6}{6,144} = 300$ | $\frac{F}{6,144}$ |
| 0 1 1 1 | 3,072 | $\frac{1.8432 \times 10^6}{3,072} = 600$ | $\frac{F}{3,072}$ |
| 1 0 0 0 | 1,536 | $\frac{1.8432 \times 10^6}{1,536} = 1,200$ | $\frac{F}{1,536}$ |
| 1 0 0 1 | 1,024 | $\frac{1.8432 \times 10^6}{1,024} = 1,800$ | $\frac{F}{1,024}$ |
| 1 0 1 0 | 768 | $\frac{1.8432 \times 10^6}{768} = 2,400$ | $\frac{F}{768}$ |
| 1 0 1 1 | 512 | $\frac{1.8432 \times 10^6}{512} = 3,600$ | $\frac{F}{512}$ |
| 1 1 0 0 | 384 | $\frac{1.8432 \times 10^6}{384} = 4,800$ | $\frac{F}{384}$ |
| 1 1 0 1 | 256 | $\frac{1.8432 \times 10^6}{256} = 7,200$ | $\frac{F}{256}$ |
| 1 1 1 0 | 192 | $\frac{1.8432 \times 10^6}{192} = 9,600$ | $\frac{F}{192}$ |
| 1 1 1 1 | 96 | $\frac{1.8432 \times 10^6}{96} = 19,200$ | $\frac{F}{96}$ |

R6551

Asynchronous Communications Interface Adapter (ACIA)

DIAGNOSTIC LOOP-BACK OPERATING MODES

A simplified block diagram for a system incorporating an ACIA is shown in Figure 18.

It may be desirable to include in the system a facility for "loop-back" testing, of which there are two kinds:

1. Local Loop-Back

Loop-back from the point of view of the processor. In this case, the Modem and Data Link must be effectively disconnected and the ACIA transmitter connected back to its own receiver, so that the processor can perform diagnostic checks on the system, excluding the actual data channel.

2. Remote Loop-Back

Loop-back from the point of view of the Data Link and Modem. In this case, the processor, itself, is disconnected and all received data is immediately retransmitted, so the system on the other end of the Data Link may operate independent of the local system.

The ACIA does not contain automatic loop-back operating modes, but they may be implemented with the addition of a small amount of external circuitry. Figure 19 indicates the necessary logic to be used with the ACIA. The LLB line is the positive-true signal to enable local loop-back operation. Essentially, LLB = high does the following:

1. Disables outputs Tx_D, $\overline{\text{DTR}}$, and $\overline{\text{RTS}}$ (to Modem).
2. Disables inputs Rx_D, $\overline{\text{DCD}}$, $\overline{\text{CTS}}$, $\overline{\text{DSR}}$ (from Modem).
3. Connects transmitter outputs to respective receiver inputs (i.e., Tx_D to Rx_D, $\overline{\text{DTR}}$ to $\overline{\text{DCD}}$, $\overline{\text{RTS}}$ to $\overline{\text{CTS}}$).

LLB may be tied to a peripheral control pin (from an R6520 or R6522, for example) to provide processor control of local loop-

back operation. In this way, the processor can easily perform local loop-back diagnostic testing.

Remote loop-back does not require this circuitry, so LLB must be set low. However, the processor must select the following:

1. Control Register bit 4 must be 1, so that the transmitter clock equals the receiver clock.
2. Command Register bit 4 must be 1 to select Echo Mode.
3. Command Register bits 3 and 2 must be 1 and 0, respectively to disable $\overline{\text{IRQ}}$ interrupt to transmitter.
4. Command Register bit 1 must be 0 to disable $\overline{\text{IRQ}}$ interrupt for receiver.

In this way, the system re-transmits received data without any effect on the local system.

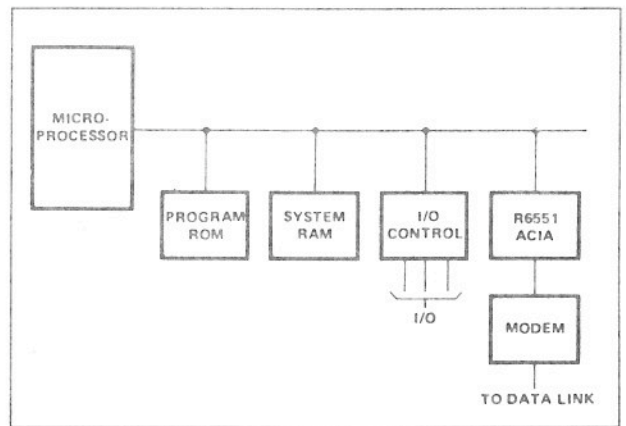


Figure 18. Simplified System Diagram

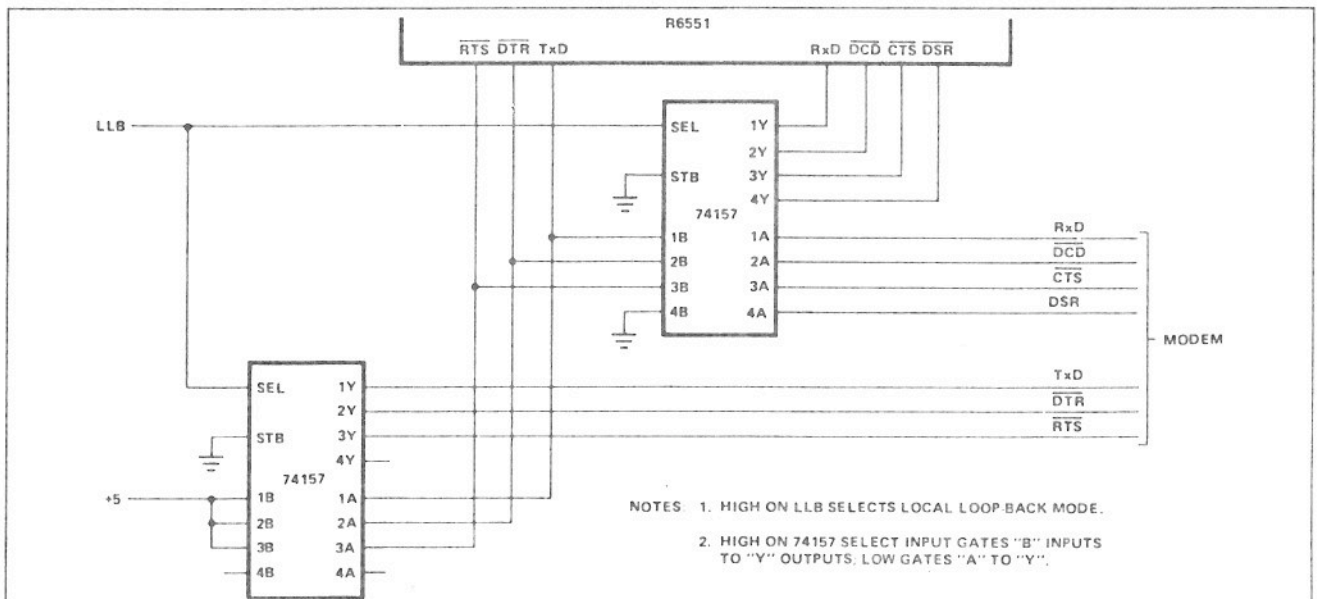


Figure 19. Loop-Back Circuit Schematic

R6551

Asynchronous Communications Interface Adapter (ACIA)

READ TIMING DIAGRAM

Timing diagrams for transmit with external clock, receive with external clock, and $\overline{\text{IRQ}}$ generation are shown in Figures 20, 21 and 22, respectively. The corresponding timing characteristics are listed in Table 3.

Table 3. Transmit/Receive Characteristics

| Characteristic | Symbol | 1 MHz | | 2 MHz | | Unit |
|---|------------------|-------|-----------|-------|-----------|----------|
| | | Min | Max | Min | Max | |
| Transmit/Receive Clock Rate | t_{CCY} | 400* | — | 400* | — | ns |
| Transmit/Receive Clock High Time | t_{CH} | 175 | — | 175 | — | ns |
| Transmit/Receive Clock Low Time | t_{CL} | 175 | — | 175 | — | ns |
| XTLI to TxD Propagation Delay | t_{DD} | — | 500 | — | 500 | ns |
| RTS Propagation Delay | t_{DLY} | — | 500 | — | 500 | ns |
| $\overline{\text{IRQ}}$ Propagation Delay (Clear) | t_{IRQ} | — | 500 | — | 500 | ns |
| Load Capacitance DTR, RTS TxD | C_L | — | 130 30 | — | 130 30 | pF pF |

Notes:
 $(t_R, t_F = 10 \text{ to } 30 \text{ ns})$
 *The baud rate with external clocking is: $\text{Baud Rate} = \frac{1}{16 \times t_{\text{CCY}}}$

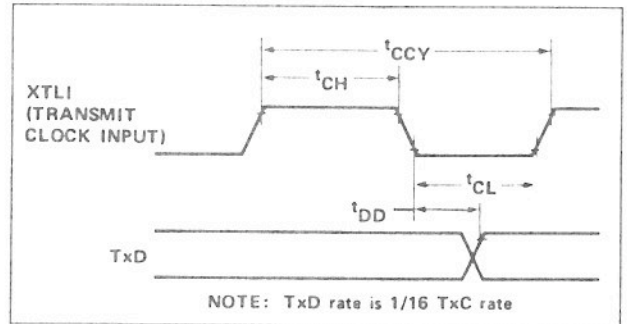


Figure 20. Transmit Timing with External Clock

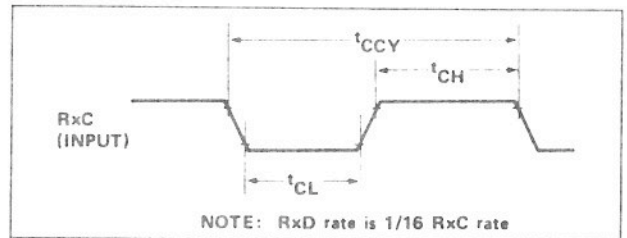


Figure 21. Receive External Clock Timing

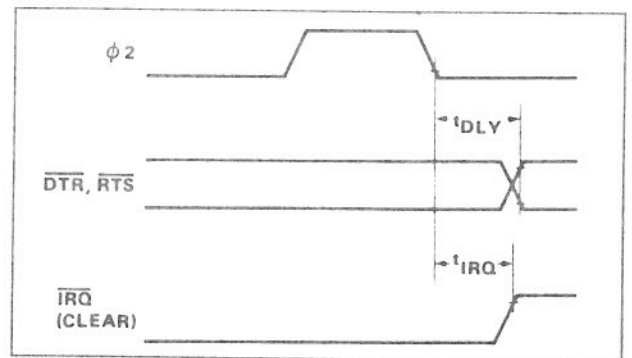


Figure 22. Interrupt and Output Timing

R6551

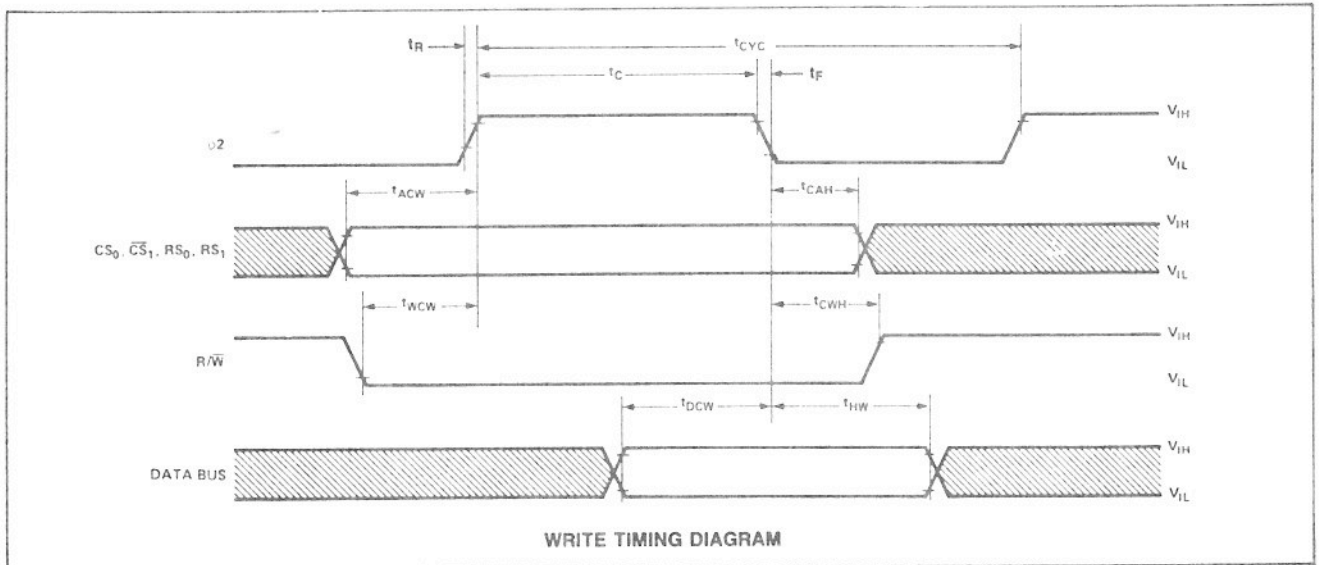
Asynchronous Communications Interface Adapter (ACIA)

AC CHARACTERISTICS

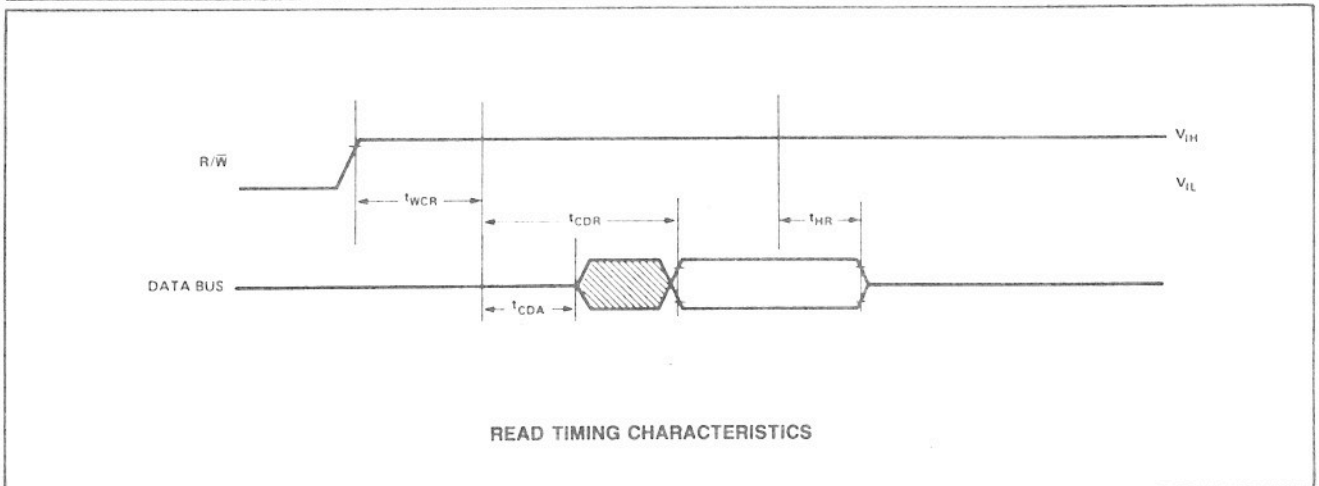
($V_{CC} = 5.0V \pm 5\%$, $V_{SS} = 0$, $T_A = T_L$ to T_H , unless otherwise noted)

| Parameter | Symbol | 1 MHz | | 2 MHz | | Unit |
|--------------------------------|-----------|-------|-----|-------|-----|---------|
| | | Min | Max | Min | Max | |
| Cycle Time | t_{CYC} | 1.0 | 40 | 0.5 | 40 | μs |
| $\phi 2$ Pulse Width | t_C | 400 | — | 200 | — | ns |
| Address Set-Up Time | t_{AC} | 120 | — | 70 | — | ns |
| Address Hold Time | t_{CAH} | 0 | — | 0 | — | ns |
| R/W Set-Up Time | t_{WC} | 120 | — | 70 | — | ns |
| R/W Hold Time | t_{CWH} | 0 | — | 0 | — | ns |
| Data Bus Set-Up Time | t_{DCW} | 150 | — | 60 | — | ns |
| Data Bus Hold Time | t_{HW} | 20 | — | 20 | — | ns |
| Read Access Time (Valid Data) | t_{CDR} | — | 200 | — | 150 | ns |
| Read Hold Time | t_{HR} | 20 | — | 20 | — | ns |
| Bus Active Time (Invalid Data) | t_{CDA} | 40 | — | 40 | — | ns |

- Notes: 1. $V_{CC} = 5.0V \pm 5\%$.
 2. $T_A = T_L$ to T_H .
 3. t_R and $t_F = 10$ to 30 ns.
 4. D0-D7 load capacitance = 130 pF.



WRITE TIMING DIAGRAM



READ TIMING CHARACTERISTICS

R6551

Asynchronous Communications Interface Adapter (ACIA)

ABSOLUTE MAXIMUM RATINGS

| Parameter | Symbol | Value | Unit |
|-----------------------|-----------|------------------|------|
| Supply Voltage | V_{CC} | -0.3 to +7.0 | Vdc |
| Input Voltage | V_{IN} | -0.3 to V_{CC} | Vdc |
| Output Voltage | V_{OUT} | -0.3 to V_{CC} | Vdc |
| Operating Temperature | T_A | 0 to +70 | °C |
| Storage Temperature | T_{STG} | -55 to +150 | °C |

***NOTE:** 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 any other conditions above those indicated in other sections of this document is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

OPERATING CONDITIONS

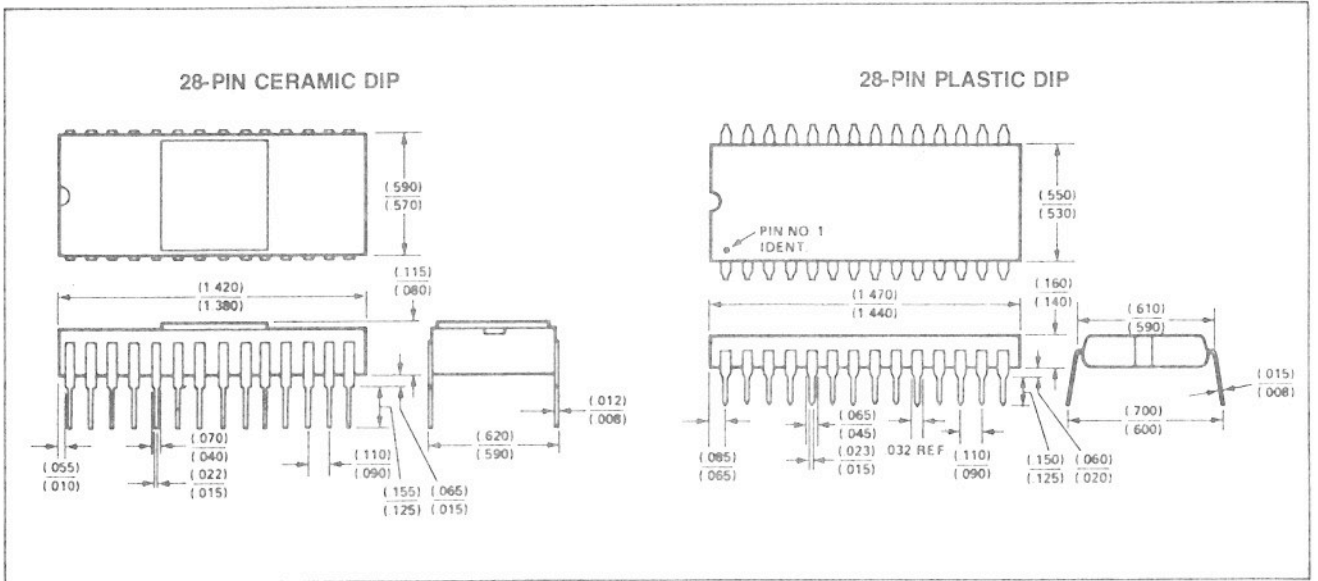
| Parameter | Symbol | Value |
|---|----------|------------------------------|
| Supply Voltage | V_{CC} | 5V ±5% |
| Temperature Range Commercial Industrial | T_A | 0° to 70°C -40°C to +85°C |

DC CHARACTERISTICS

($V_{CC} = 5.0V \pm 5\%$, $V_{SS} = 0$, $T_A = T_L$ to T_H , unless otherwise noted)

| Parameter | Symbol | Min | Typ | Max | Unit | Test Conditions |
|---|-----------|----------------------|--------|----------------------|------|---|
| Input High Voltage Except XTLI and XTLO XTLI and XTLO | V_{IH} | 2.0 2.4 | — — | V_{CC} V_{CC} | V | |
| Input Low Voltage Except XTLI and XTLO XTLI and XTLO | V_{IL} | V_{SS} V_{SS} | — — | 0.8 0.4 | V | |
| Input Leakage Current Ø2, R/W, RES, CS0, CS1, RS0, RS1, CTS, RxD, DCD, DSR | I_{IN} | — | — | 2.5 | µA | $V_{IN} = 0V$ to 5.25V $V_{CC} = 0V$ |
| Input Leakage Current for High Impedance (Three State Off) D0-D7 | I_{TSI} | — | ±2 | ±10.0 | µA | $V_{IN} = 0.4V$ to 2.4V $V_{CC} = 5.25V$ |
| Output High Voltage D0-D7, TxD, RxC, RTS, DTR | V_{OH} | 2.4 | — | — | V | $I_{LOAD} = -100 \mu A$ $V_{CC} = 4.75V$ |
| Output Low Voltage D0-D7, TxD, RxC, RTS, DTR, IRQ | V_{OL} | — | — | 0.4 | V | $V_{CC} = 4.75V$ $I_{LOAD} = 1.6 mA$ |
| Output High Current (Sourcing) D0-D7, TxD, RxC, RTS, DTR | I_{OH} | -100 | — | — | µA | $V_{OH} = 2.4V$ |
| Output Low Current (Sinking) D0-D7, TxD, RxC, RTS, DTR, IRQ | I_{OL} | 1.6 | — | — | mA | $V_{OL} = 0.4V$ |
| Output Leakage Current (off state) IRQ | I_{OFF} | — | — | 10.0 | µA | $V_{OUT} = 5V$ |
| Clock Capacitance (Ø2) | C_{CLK} | — | — | 20 | pF | $V_{CC} = 5V$ $V_{IN} = 0V$ $f = 1 MHz$ $T_A = 25^\circ C$ |
| Input Capacitance except Ø2 | C_{IN} | — | — | 10 | pF | |
| Output Capacitance | C_{OUT} | — | — | 10 | pF | |
| Power Dissipation | P_D | — | 170 | 300 | mW | $T_A = 25^\circ C$ |

PACKAGE DIMENSIONS



Information furnished by Rockwell International Corporation is believed to be accurate and reliable. However, no responsibility is assumed by Rockwell International for its use, nor any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Rockwell International other than for circuitry embodied in a Rockwell product. Rockwell International reserves the right to change circuitry at any time without notice. This specification is subject to change without notice.

©Rockwell International Corporation 1984
All Rights Reserved

Printed in U.S.A.

SEMICONDUCTOR PRODUCTS DIVISION REGIONAL ROCKWELL SALES OFFICES

HOME OFFICE

Semiconductor Products Division
Rockwell International
4311 Jamboree Road
P O Box C, MS 501-300
Newport Beach, California
92658-8902
(714) 833-4700
TWX 910 591-1698

UNITED STATES

Semiconductor Products Division
Rockwell International
1842 Reynolds
Irvine, California 92714
(714) 833-4655
TWX 910 595-2518

Semiconductor Products Division
Rockwell International
3375 Scott Blvd, Suite 410
Santa Clara, California 95051
(408) 980-1900

Semiconductor Products Division
Rockwell International
921 Bowser Road
Richardson, Texas 75080
(214) 996-6500
TLX 73-307

Semiconductor Products Division
Rockwell International
10700 West Higgins Rd., Suite 102
Rosemont, Illinois 60018
(312) 297-8862
TWX 910 233-0179 (RI MED ROSM)

Semiconductor Products Division
Rockwell International
5001B Greentree
Executive Campus, Rt. 73
Marlton, New Jersey 08053
(609) 596-0090
TWX 710 940-1377

FAR EAST

Semiconductor Products Division
Rockwell International Overseas Corp
Itohopia Hirakawa cho Bldg
7-6, 2-chome, Hirakawa-cho
Chiyoda-ku, Tokyo 102, Japan
(03) 265-8806
TLX J22198

Rockwell Collins International
Tai Sang Commercial Bldg, 11th Floor
24-34 Hennessy Rd
Hong Kong
(5) 274-321
TLX 74071 HK

EUROPE

Semiconductor Products Division
Rockwell International GmbH
Fraunhoferstrasse 11
D-8033 Munchen-Martinsried
West Germany
(089) 857-6016
TLX 0521/2650 rimd d

Semiconductor Products Division
Rockwell International
Heathrow House, Bath Rd.
Cranford, Hounslow
Middlesex, England
(01) 759-9911
TLX 851-25463

Semiconductor Products
Rockwell Collins Italiana S P A
Via Boccaccio, 23
20123 Milano, Italy
(02) 498 74 79
TLX: 316562 RCIMIL 1

YOUR LOCAL REPRESENTATIVE

Klokhuis Vierlingkaart

Klokhuis Vierlingkaart

MSM5832RS

MICROPROCESSOR REAL-TIME CLOCK/CALENDAR

T-750019 ISSUE 1B
APRIL 24, 1981

Klokhuis Vierlingkaart

MSM5832 MICROPROCESSOR REAL-TIME CLOCK/CALENDAR

GENERAL DESCRIPTION

The MSM5832 is a monolithic, metal-gate CMOS integrated circuit that functions as a real time clock/calendar for use in bus-oriented microprocessor applications. The on-chip 32.768Hz crystal controlled oscillator time base is counted down to provide addressable 4-bit I/O data of SECONDS, MINUTES, HOURS, DAY-OF-WEEK, DATE, MONTH, and YEAR. Data access is controlled by 4-bit address, chip select, read, write and hold inputs. Other functions include 12H/24H format selection, leap year identification and manual ± 30 second correction.

The MS5832 normally operates from a 5 volt $\pm 5\%$ supply. Battery back-up operation down to 2.2 volts allows continuation of time keeping when main power is off. One test input facilitates rapid testing of the time keeping operations. The MS5832 is offered in an 18-lead dual-in-line plastic (RS suffix) package.

FEATURES

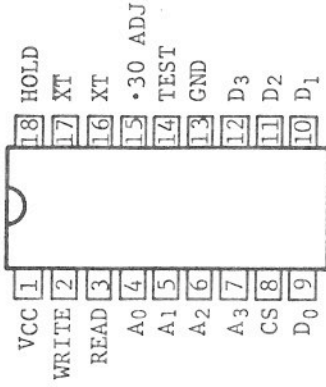
- Microprocessor bus-oriented

| TIME | MONTH | DATE | YEAR | DAY OF WEEK |
|----------|-------|------|------|-------------|
| 23:59:59 | 12 | - 31 | - 99 | - 7 |

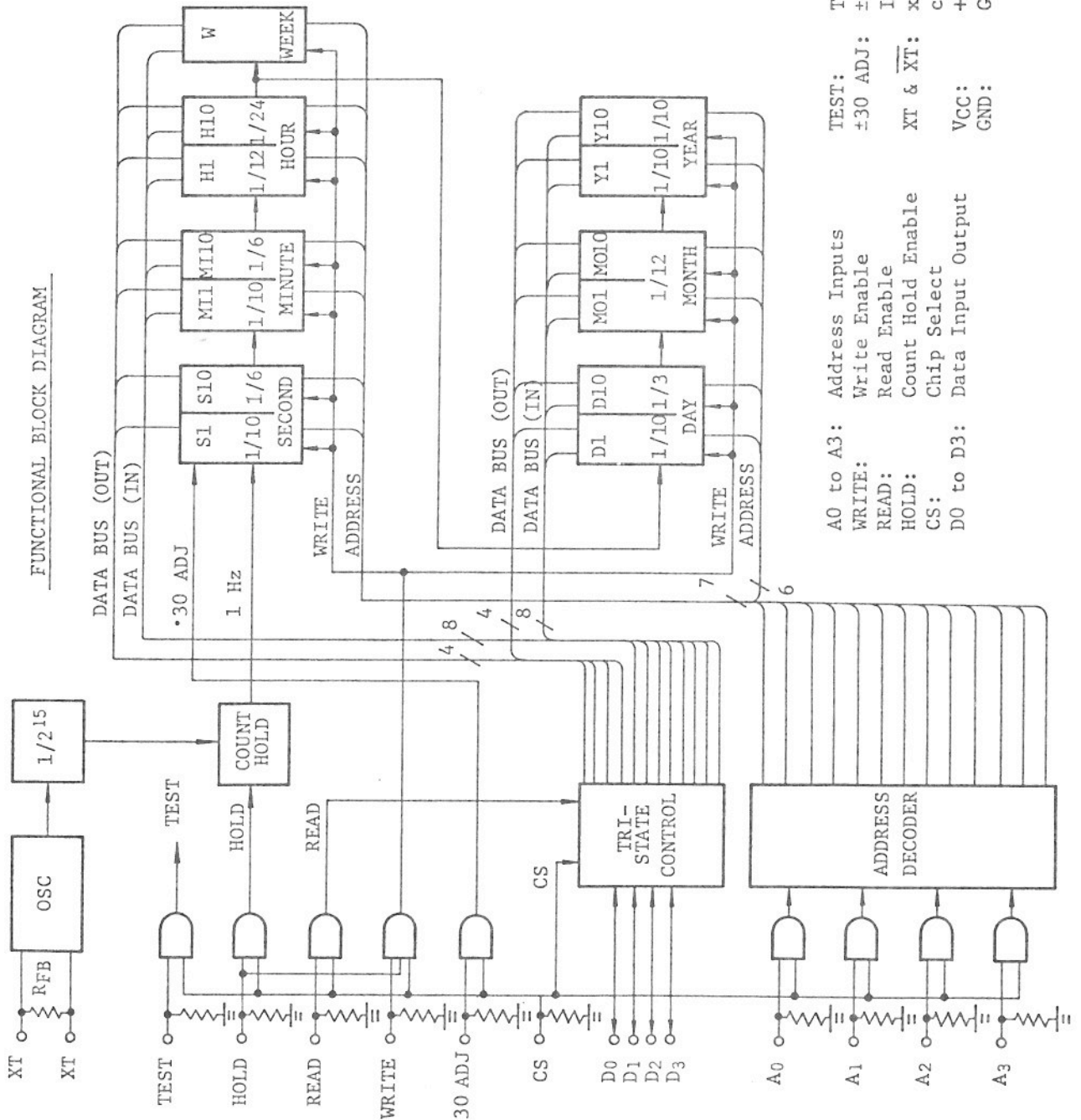
- 4-BIT DATA BUS
- 4-BIT ADDRESS
- Read, Write, Hold Chip select inputs
- Interrupt signal outputs -- 1024, 1, 1/60, 1/3600Hz
- 32.768kHz crystal controlled operation
- Leap year register bit
- 12 or 24 hour format
- ± 30 second error correction
- Single 5 volt power supply
- Back-up battery operation to $V_{CC} = 2.2V$
- Low Power Dissipation
 - 90 μW Max. at $V_{CC} = 3V$
 - 2.5 mW Max. at $V_{CC} = 5V$
- High Density 300 mil 18-Pin Package

Klokhuis Vierlingkaart

PIN CONFIGURATION



FUNCTIONAL BLOCK DIAGRAM



- A0 to A3: Address Inputs
- WRITE: Write Enable
- READ: Read Enable
- HOLD: Count Hold Enable
- CS: Chip Select
- D0 to D3: Data Input Output
- TEST: Test Input
- *30 ADJ: ±30 Second Correction
- XT & XT: xtal oscillator connections
- VCC: +5V Supply
- GND: Ground

Klokhuis Vierlingkaart

FUNCTION TABLE

FIGURE 1

| ADDRESS INPUTS | | | | INTERNAL COUNTER | DATA I/O | | | | DATA LIMITS | NOTES |
|----------------|----------------|----------------|----------------|------------------|----------------|----------------|----------------|----------------|-------------|---|
| A ₀ | A ₁ | A ₂ | A ₃ | | D ₀ | D ₁ | D ₂ | D ₃ | | |
| 0 | 0 | 0 | 0 | S 1 | * | * | * | * | 0~9 | S1 or S10 are reset to zero irrespective of input data D0~D3 when write instruction is executed with address selection. |
| 1 | 0 | 0 | 0 | S 10 | * | * | * | | 0~5 | |
| 0 | 1 | 0 | 0 | MI 1 | * | * | * | * | 0~9 | |
| 1 | 1 | 0 | 0 | MI 10 | * | * | * | | 0~5 | |
| 0 | 0 | 1 | 0 | H 1 | * | * | * | * | 0~9 | |
| 1 | 0 | 1 | 0 | H 10 | * | * | † | † | 0~1 0~2 | D2 = "1" for PM D3 = "1" for 24 hour format D2 = "0" for AM D3 = "0" for 12 hour format |
| 0 | 1 | 1 | 0 | W | * | * | * | | 0~6 | |
| 1 | 1 | 1 | 0 | D 1 | * | * | * | * | 0~9 | |
| 0 | 0 | 0 | 1 | D 10 | * | * | † | | 0~3 | D2 = "1" for 29 days in month 2 D2 = "0" for 28 days in month 2 (2) |
| 1 | 0 | 0 | 1 | MO 1 | * | * | * | * | 0~9 | |
| 0 | 1 | 0 | 1 | MO 10 | * | | | | 0~1 | |
| 1 | 1 | 0 | 1 | Y 1 | * | * | * | * | 0~9 | |
| 0 | 0 | 1 | 1 | Y 10 | * | * | * | * | 0~9 | |

(1) * data valid as "0" or "1".

Blank does not exist (unrecognized during a write and held at "0" during a read)

† databits used for AM/PM, 12/24 HOUR and leap year.

(2) If D2 previously set to "1", upon completion of month 2 day 29, D2 will be internally reset to "0".

TYPICAL CHARACTERISTICS — Oscillator Frequency Deviations

Frequency Deviation vs Temperature

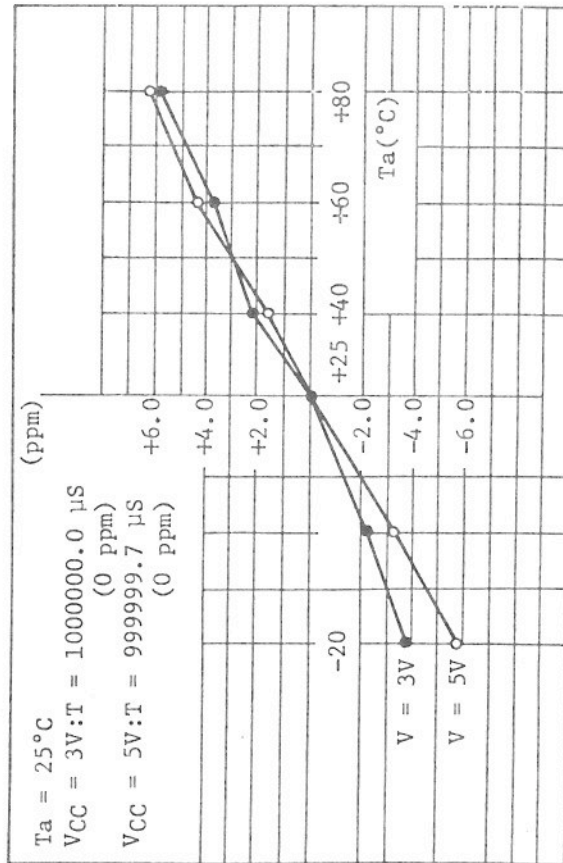


FIGURE 2

Klokhuis Vierlingkaart

Frequency Deviation vs Supply Voltage

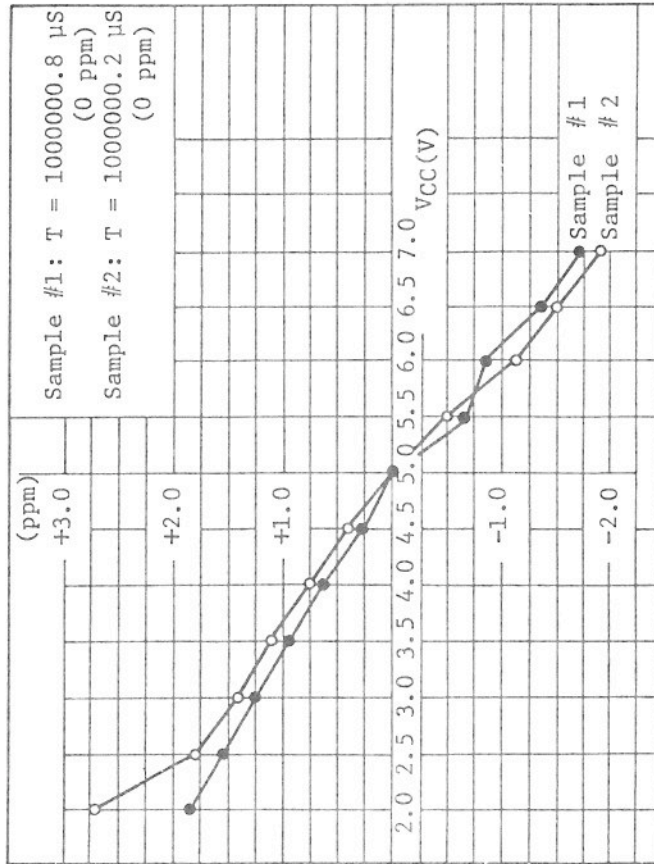


FIGURE 3

Klokhuis Vierlingkaart

ABSOLUTE MAXIMUM RATINGS

| Rating | Symbol | Value | Unit |
|---------------------|------------------|------------------------------|------|
| Supply voltage | V _{CC} | -0.3 ~ 7.0 | V |
| Input Voltage | V _{IN} | -0.3 ~ V _{CC} + 0.3 | V |
| Data I/O Voltage | V _D | -0.3 ~ V _{CC} + 0.3 | V |
| Storage Temperature | T _{stg} | -55 ~ 150 | °C |

Note: 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 at any other condition 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.

OPERATING CONDITIONS

| Parameter | Symbol | Min. | Typ. | Max. | Unit | Conditions |
|--------------------------|------------------|------|--------|-----------------|------|---|
| Supply Voltage | V _{CC} | 4.75 | 5 | 5.25 | V | 5V ±5% |
| Standby Supply Voltage | V _{CCS} | 2.2 | 5 | 7 | V | |
| Input Signal Level | V _{IH} | 3.6 | 5 | V _{CC} | V | V _{CC} = 5V ± 5% Respect to GND |
| | V _{IL} | -0.3 | 0 | 0.8 | V | |
| Crystal Oscillator Freq. | f(XT) | | 32.768 | | kHz | |
| Operating Temperature | T _a | -30 | | +85 | °C | |

Klokhuis Vierlingkaart

DC CHARACTERISTICS

($V_{CC} = 5V \pm 5\%$; $T_A = -30$ to $+85^\circ\text{C}$, unless otherwise noted)

| Parameter | Symbol | Min. | Typ. | Max. | Unit | Conditions |
|--------------------------|-----------|------|------|------|---------------|--|
| Input Current (1) | I_{TH} | 10 | 25 | 50 | μA | $V_{IN} = 5V$ |
| | I_{IL} | -1 | | 1 | μA | $V_{IN} = 0V$ |
| Data I/O Leakage Current | I_{LD} | -1 | | 1 | μA | $V_{I/O} = 0$ to V_{CC} , CS = "0" |
| Output Low Voltage | V_{OL} | | | 0.4 | V | $I_O = 1.6$ ma, CS = "1", READ = "1" |
| Output Low Current | I_{OL} | 1.6 | | | mA | $V_O = 0.4V$, CS = "1", READ = "1" |
| Operating Supply Current | I_{CCS} | | | 30 | μA | $V_{CC} = 3V$, $T_a = 25^\circ\text{C}$ |
| | I_{CC} | | | 500 | μA | $V_{CC} = 5V$, $T_a = 25^\circ\text{C}$ |

(1) XT, \overline{XT} and $D_0 \sim D_3$ excluded.

AC CHARACTERISTICS

CAPACITANCE

$T_A = 25^\circ\text{C}$, $f = 1\text{MHz}$

| Parameter | Symbol | Min. | Typ. | Max. | Unit |
|--------------------------|-----------|------|------|------|------|
| Input/Output Capacitance | $C_{I/O}$ | | | 8 | pF |
| Input Capacitance | C_{IN} | | | 5 | pF |

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

Klokhuis Vierlingkaart

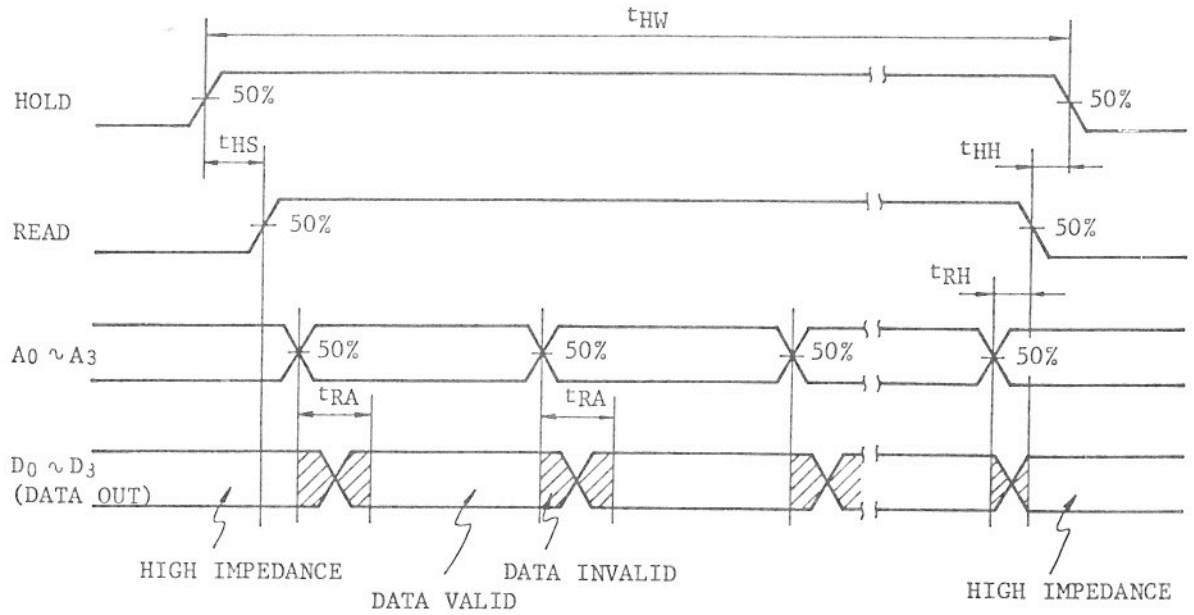
READ CYCLE

($V_{CC} = 5V \pm 5\%$; $T_a = 25^\circ C$)

| Parameter | Symbol | Min. | Typ. | Max. | Unit |
|------------------|----------|------|------|------|---------|
| HOLD Set-up Time | t_{HS} | 150 | | | μS |
| HOLD Hold Time | t_{HH} | 0 | | | μS |
| HOLD Pulse Width | t_{HW} | | | 1 | SEC |
| READ Hold Time | t_{RH} | 0 | | | μS |
| READ Access Time | t_{RA} | | | 6 | μS |

READ CYCLE

FIGURE 4



- Notes:
1. A Read occurs during the overlap of a high CS and a high READ.
 2. Output Load: 1 TTL Gate, $C_L = 50$ pf and $R_L = 4.7k\Omega$
 3. CS may be a permanent "1", or may be coincident with HOLD pulse.

Klokhuis Vierlingkaart

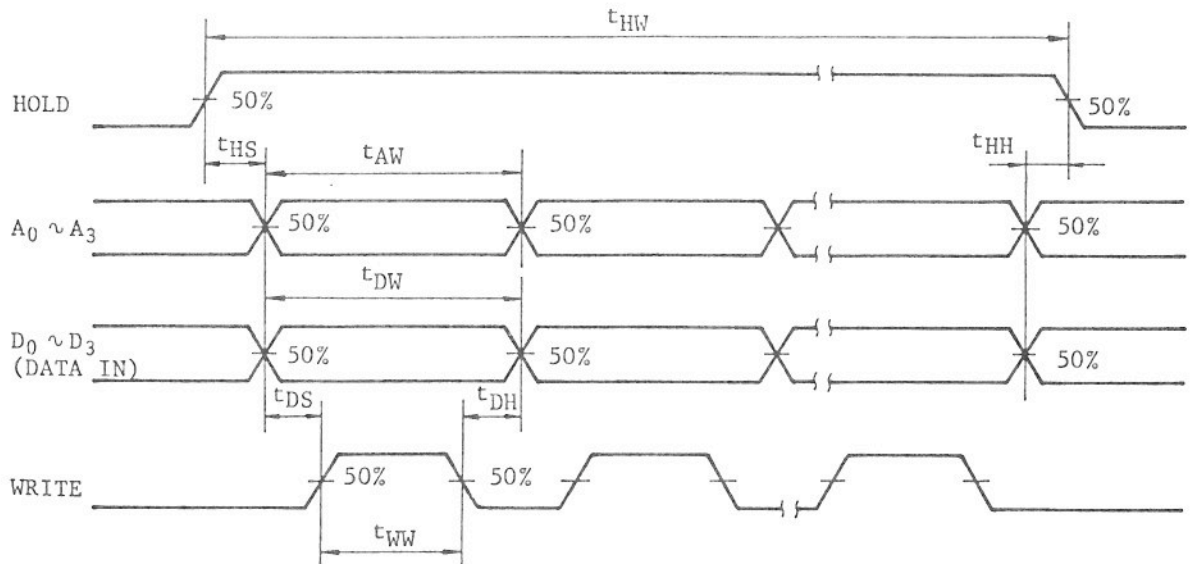
WRITE CYCLE

($V_{CC} = 5V \pm 5\%$; $T_a = 25^\circ C$)

| Parameter | Symbol | Min. | Typ. | Max. | Unit |
|---------------------|----------|------|------|------|---------|
| HOLD Set-up Time | t_{HS} | 150 | | | μS |
| HOLD Hold Time | t_{HH} | 0 | | | μS |
| HOLD Pulse Width | t_{HW} | | | 1 | SEC |
| ADDRESS Pulse Width | t_{AW} | 1.7 | | | μS |
| DATA Pulse Width | t_{DW} | 1.7 | | | μS |
| DATA Set-up Time | t_{DS} | 0.5 | | | μS |
| DATA Hold Time | t_{DH} | 0.2 | | | μS |
| WRITE Pulse Width | t_{WW} | 1.0 | | | μS |

WRITE CYCLE

FIGURE 5



- Notes: 1. A WRITE occurs during the overlap of a high CS, a high HOLD and a high WRITE.
 2. CS may be permanent "1", or may be coincident with HOLD pulse.

Klokhuis Vierlingkaart

FUNCTIONAL DESCRIPTION

A block diagram of the MSM5832 microprocessor real-time clock/calendar and a package connection diagram are shown on the first page. Figure 9 illustrates a method of interfacing between the clock/calendar circuit and a micro processor. Figures 9, 10 and 11 illustrate alternative standby power supply circuits. A function table listing relationships between address inputs, data input/output and internal counter selection is shown in Figure 1. Unless otherwise indicated, the following descriptions are based on the block diagram.

32.768kHz OSCILLATOR (pins 16 and 17): An internal inverting amplifier with feedback resistor, R_{FB} , is connected with a crystal and two capacitors as shown in Figure 6 to form a stable, accurate oscillator -- which serves as the precision time base of the circuit. Capacitors C1 and C2 in series provide the parallel load capacitance required for precise tuning of the quartz crystal. Typical oscillator performance as a function of ambient temperature and supply voltage is shown in Figures 2 and 3 respectively.

A0 ~ A3 (pins 4 ~ 7): Address inputs, used to select internal counters for read/write operations (see function table -- Figure 1). A "1" is defined as V_{CC} ; a "0" is GND. Pull-down to GND is provided by internal resistors.

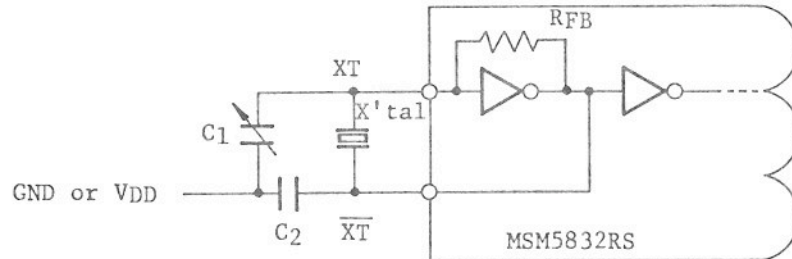
D0 ~ D3 (pins 9 ~ 12): Data Inputs/Outputs, two-way bus lines controlled by READ and WRITE inputs. As shown in Figure 7 external pull-up resistors of 4.7K or higher are required by the open-drain N-channel MOS outputs. D3 is the MSB; D0 is the LSB.

TEST (pin 14): Normally this input is unconnected -- pull-down to GND is provided by an internal resistor -- or connected to GND. With CS at V_{CC} , pulses to V_{CC} on the TEST input will directly clock the S₁, MI₁₀, W, D₁ and Y₁ counters, depending on which counter is addressed (W and D₁ are select-d by D₁ address in this mode only). Roll-over to next counter is enabled in this mode.

Klokhuis Vierlingkaart

OSCILLATOR CIRCUIT

FIGURE 6



$$C_1 \sim C_2 = 15 \sim 30\text{pF}$$

CHIP SELECT (pin 8): Connecting CS input to V_{CC} enables all inputs and outputs. Unconnected -- pull-down to GND is provided by an internal resistor -- or connecting CS to GND will disable HOLD, WRITE, READ, ± 30 ADJ, $D_0 \sim D_3$, $A_0 \sim D_3$ and TEST.

As shown in Figure 9 CS can be used to detect system power failure by connecting system power (+5V) to CS, so that when system power is on, all inputs and outputs will be enabled, and when system power is off, all inputs and outputs will be disabled. The threshold voltage of CS is higher than all other inputs to insure correct operation of this function.

HOLD (pin 18): Switching this input to V_{CC} inhibits the internal 1Hz clock to the S1 counter. After the specified HOLD set-up time (150 μs), all counters will be in a static state, thus allowing error-free read or write operations. So long as the HOLD pulse width is less than 1 second, accuracy of the real time will be undisturbed. Pull-down to GND is provided by an internal resistor.

READ (pin 3): Read function as shown in Figure 4 is enabled when READ is switched to V_{CC} . Pull-down to GND is provided by an internal resistor.

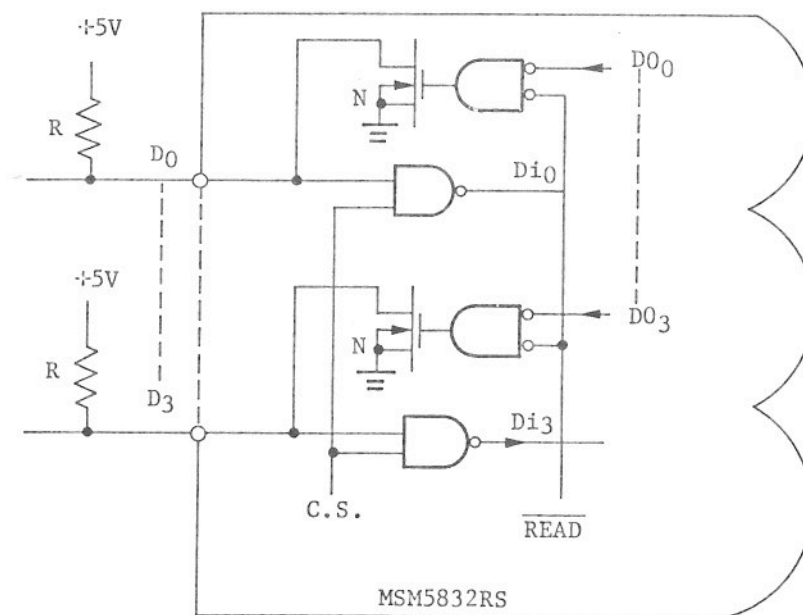
Klokhuis Vierlingkaart

WRITE (pin 2): Write function as shown in Figure 5 is enabled when WRITE is switched to VCC. Pull-down to GND is provided by an internal resistor.

± 30 ADJ (pin 15): Momentarily connecting this input to VCC (>31.25 ms) will reset seconds (S1, S10 counters and $2^{11} \sim 2^{15}$ frequency dividers) to 00; if seconds were 30 or more, one minute is added to the minutes (MI 1 counter) and if seconds were less than 30, the minutes are unchanged. Pull-down to GND is provided by an internal resistor.

DATA I/O CIRCUIT

FIGURE 7



REFERENCE SIGNAL OUTPUT

Reference signals are available as outputs on $D_0 \sim D_3$ if CS, READ and $A_0 \sim A_3$ are at VCC. Refer to Figure 8 for specifics. As shown in Figure 9 these signals may be used to generate interrupts for the micro-processor.

Klokhuis Vierlingkaart

REFERENCE SIGNAL OUTPUTS

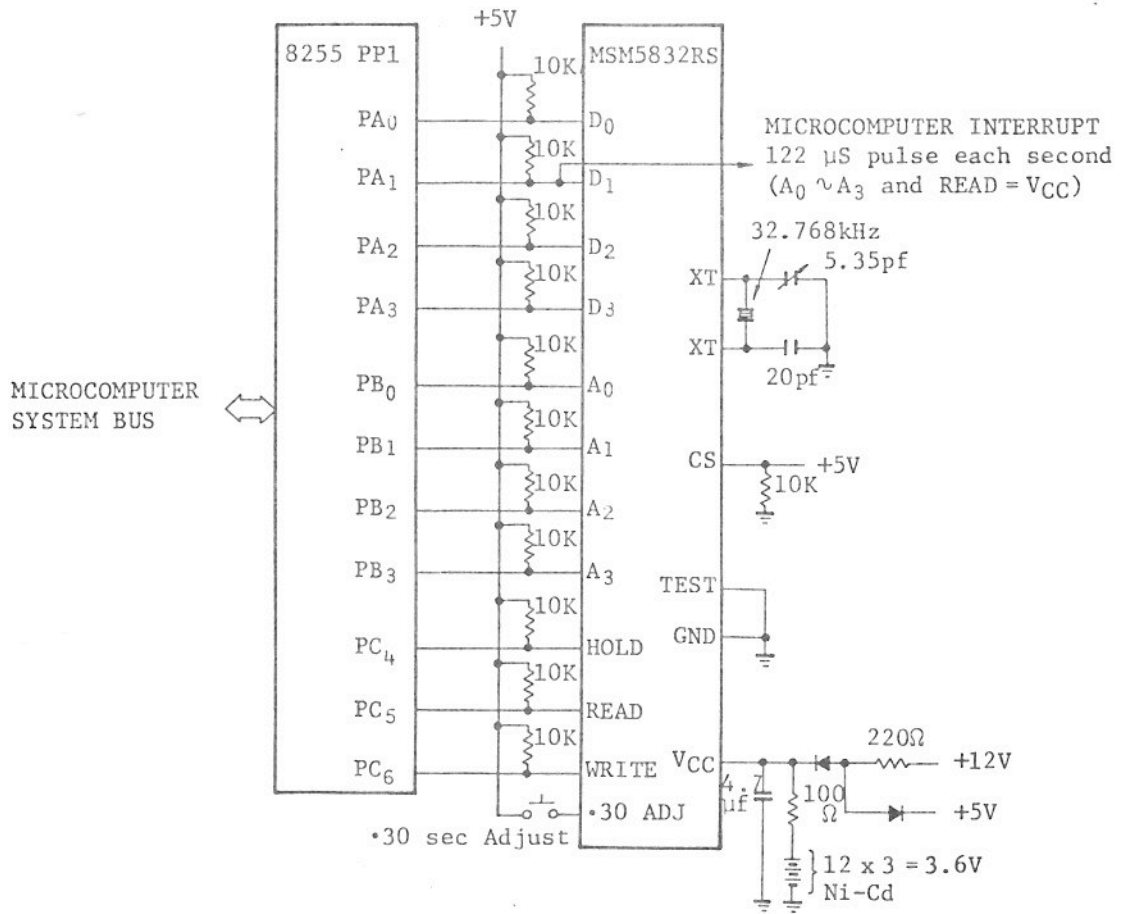
FIGURE 8

| CONDITIONS | OUTPUT | REFERENCE FREQUENCY | PULSE WIDTH |
|-------------------------------------|--------------------|---------------------|-------------|
| HOLD = L | D ₀ (1) | 1024 Hz | duty 50% |
| READ = H | D ₁ | 1 Hz | 122.1 μS |
| C.S. = H | D ₂ | 1/60 Hz | 122.1 μS |
| A ₀ ~ A ₃ = H | D ₃ | 1/3600 Hz | 122.1 μS |

(1) 1024 Hz signal at D₀ not dependent on HOLD input level

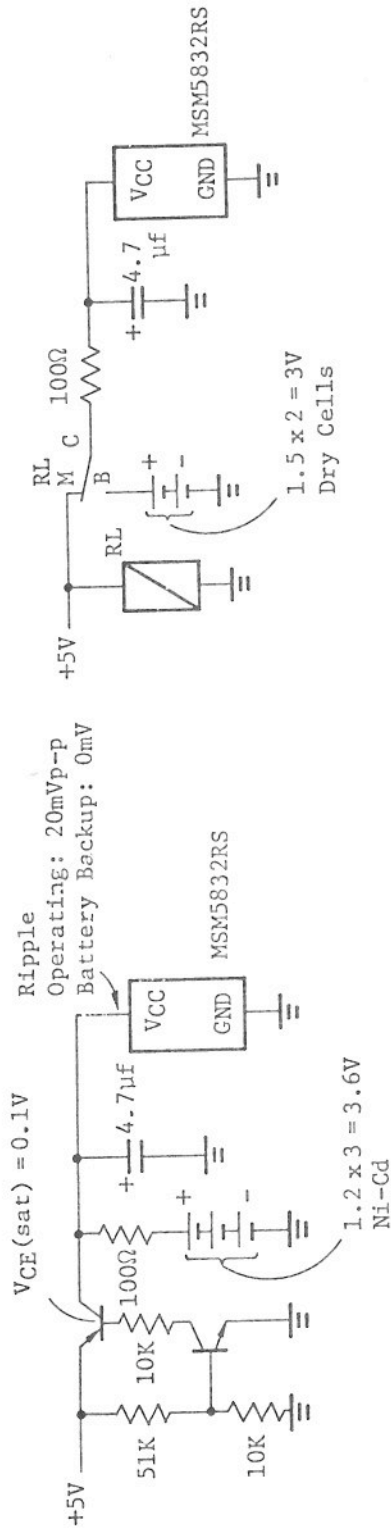
TYPICAL APPLICATION — Use with Programmable Peripheral Interface (PPI)

FIGURE 9



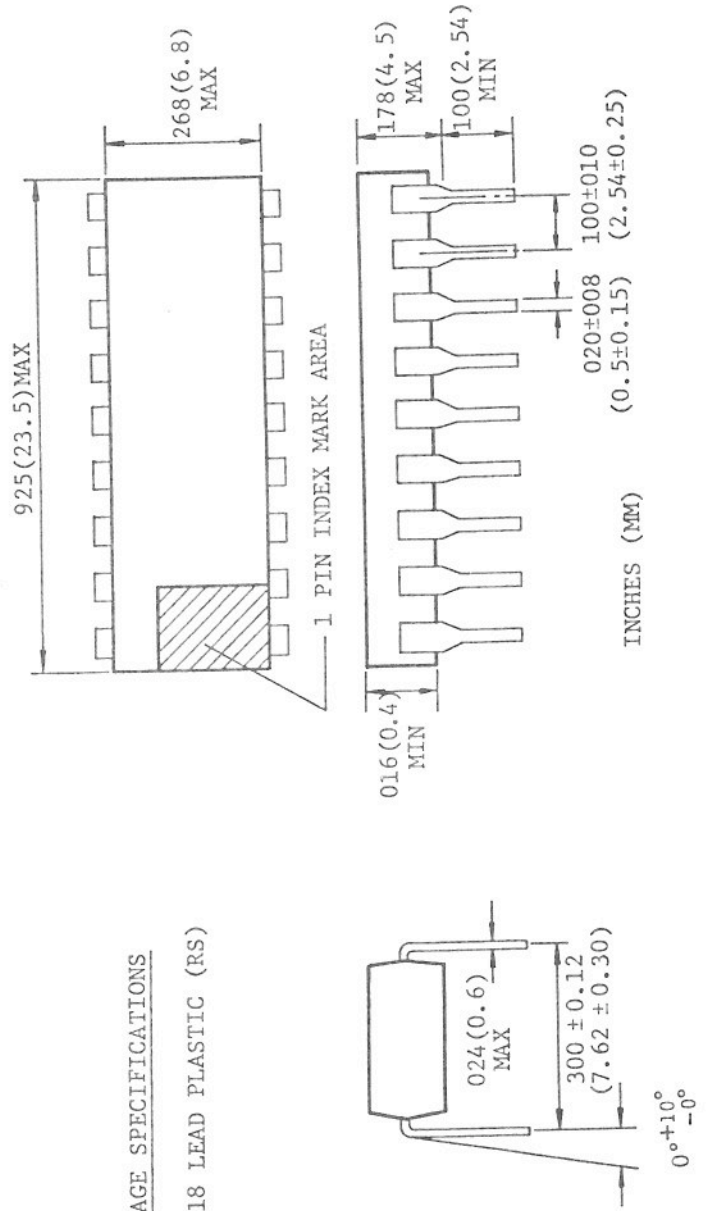
Klokhuis Vierlingkaart

TYPICAL APPLICATIONS — Alternative Standby Power Supply Circuits



PACKAGE SPECIFICATIONS

18 LEAD PLASTIC (RS)



INCHES (MM)

020±008 100±010
(0.5±0.15) (2.54±0.25)

1 PIN INDEX MARK AREA

Klokhuis Vierlingkaart

Klokhuis Vierlingkaart