

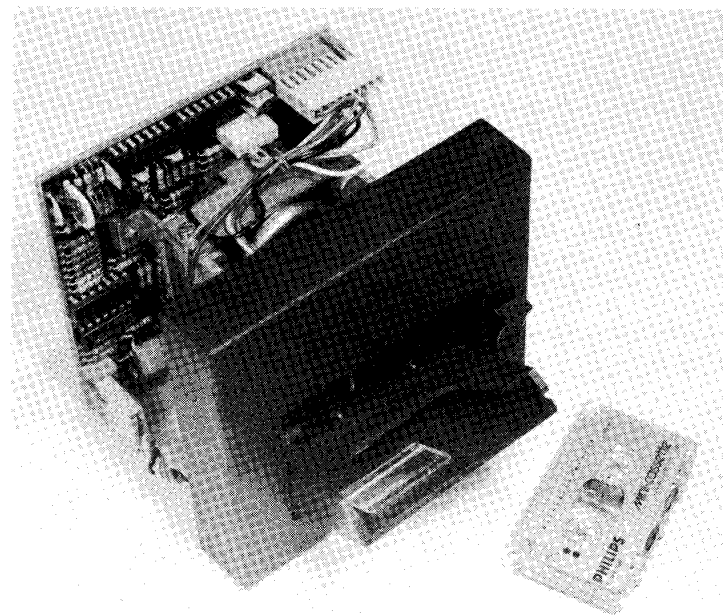
MINI-DIGITALE CASSETTERECORDER INTERFACE VOOR DE 6502

P. G. J. DE BEER
H. J. C. OTTEN

Digitale cassetterecorders staan bekend als betrouwbare opslagmedia voor bijvoorbeeld microcomputers. Philips levert een mini-digitale cassetterecorder (MDCR) die tegen een aanzienlijk lagere prijs (ongeveer f 400,00) dan gewone digitale cassette-recorders een behoorlijke opslagcapaciteit biedt van in principe 128K en een vrij hoge overdrachtsnelheid van 6000 baud.

De MDCR werkt met de bekende mini-cassettes die ook in veel dicteerapparatuur wordt toegepast. Het is een klein apparaat compleet met lees- en schrijfelektronica. Vergeleken met de veel gebruikte audio-cassetterecorder biedt de MDCR een hogere data-overdrachtsnelheid en het voordeel van volledig door de computer bestuurbaar te zijn.

In dit artikel wordt een interface tussen een 6502-computer en één of meer MDCR's beschreven en een aantal elementaire routines in 6502-assembler om gebruik te maken van de MDCR. Met opzet is de software zo universeel mogelijk ontwikkeld. Van de microcomputers met de 6502-microprocessor heeft ieder zijn eigen



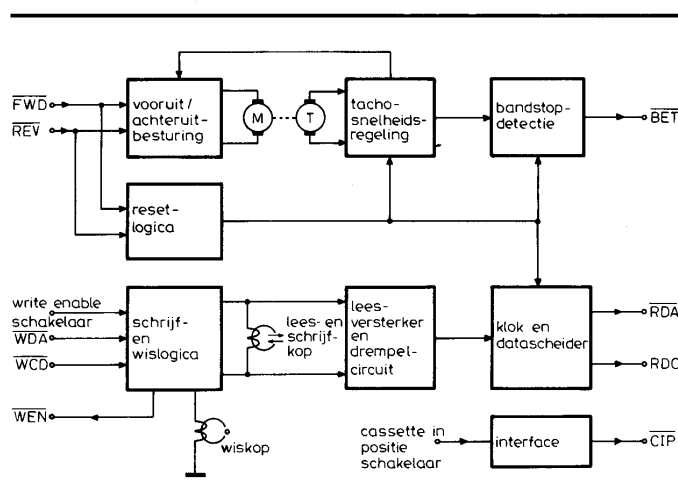
systeemsoftware, zodat inpassen in een bepaalde computer afbreuk doet aan het elementair zijn van de routine. De interface is gegeven als aanvulling op de al meerdere malen ter sprake gekomen uitbreiding voor kleine 6502-systemen. In principe is alleen een 6502-microprocessor met een kloksnelheid van 1 MHz, uitgebreid met een PIA, type 6820, 6821 of een VIA, type 6522 en de interface-elektronica zoals hier beschreven, nodig. De adressen die in de elementai-

re routines worden gebruikt, zoals zero-page, RAM-werkruimte-adressen, PIA-adressen en het adres van de code zelf zijn vrij willekeurig gekozen. Aanpassen aan de al gebruikte en nog beschikbare ruimte zal voor iedere computer verschillen.

■ MDCR

De MDCR is een zeer compacte recorder voor gestandaardiseerde mini-cassettes. Deze cassettes bevatten ca. 60 m band met een

breedte van 2,81 mm en zien er verder uit als een verkleinde versie van de ons bekende audio-cassettes. Door middel van pen-netjes kunnen eventueel beide zijden van de cassette tegen schrijven worden beschermd. De MDCR bevat slechts één motor, waarvan de draairichting door logische signalen wordt bepaald. De draaisnelheid is voor alle functies dezelfde en wel 360 toeren per minuut. De overbrenging geschiedt door middel van twee tandwielen die, afhankelijk van de richting, één van de twee spoelen aandrijven. Er is dus geen capstan, wat tevens inhoudt dat de bandsnelheid van begin tot eind niet dezelfde is. Deze varieert tussen 500 en 300 mm/s voor respectievelijk het begin en het einde van de band. Voor- en achteruitspoelen gaat even snel als lezen of schrijven en bij al deze acties blijven de koppen in contact met de band. Er is een opneem-weergeefkop en een wiskop. Een cassette heeft twee sporen en kan dus aan beide zijden worden gebruikt. De totale gebruiksduur is ca. 3 minuten en 90 seconden per kant. Er kan met een snelheid van 6000 bit/s worden geschreven en gelezen. Een overzicht



Afb. 1 Blokschema van de stuelelektronica van de MDCR.

van de signaallamen, hun betekenis en de aansluitgegevens vindt u in tabel 1. De stuurschakeling van de MDCR is gebouwd met MOS-IC's en vereist een voedingsspanning van 12 V bij 120 mA. Het blokschema is in afb. 1 te zien. Te onderscheiden zijn een motorsturing, een schrijf- en een leesversterker, een klok- en een dataseparator, een wisver-

sterker, een bandstopdetector en resetlogica. De motorsturing bestaat uit een vermogenstrap, die door middel van de signalen REV en FWD worden omgepooled. De motor blijft onbekrachtigd wanneer beide stuursignalen hoog zijn. De snelheid van de motor wordt constant gehouden door een terugkoppelsysteem met een tachometer. Het hiervan afkomstige signaal wordt rechthoekig gemaakt en gecorrigeerd door middel van een monostabiele multivibrator. Vervolgens worden deze pulsen geïntegreerd en vergeleken met een vaste referentiespanning. De motor wil langzamer gaan draaien zodat er minder pulsen van de tachometer komen. De geïntegreerde spanning wordt lager dan de referentie. De resulterende foutspanning wordt toegevoerd aan een vermogenstrap, die de spanning over de motor vergroot, waardoor deze sneller gaat lopen. Door een tweede integrator wordt geconstateerd of er nog wel pulsen van de tachometer komen. Zo niet, dan mag worden aangenomen dat het einde van de band is bereikt. De schrijfversterker bestaat uit een aantal NAND-poorten, die gezamenlijk kunnen worden geblokkeerd door de schakelaar „Write protect” en vrijgegeven door het signaal „Write command”. De leesstrap bestaat uit een verschilversterker, gevolgd door een drempelschakeling. Op

Tabel 1 Gegevens van de MDCR.

Pennr.	Signaalnaam	Functie
1	12 V	voedingsspanning
2	0 V	signaalaarde
3	0 V	voedingsaarde
4	aarde	beveiligingsaarde
5	—	niet gebruikt, referentiepunt
6	WDA-niet	write data
7	BET-niet	te schrijven data, fase-gecodeerd
8	WCD-niet	begin/end of tape
9	REV-niet	write command
10	FWD-niet	schrijven door MDCR toegestaan
11	RDC	reverse
12	RDA-niet	draai band terug
13	CIP-niet	draai band vooruit
14	WEN-niet	read clock
		klok voor gelezen data
		gelezen data
		cassette in position
		cassette in MDCR aanwezig
		write enable

worden gebracht. Tijdens het lezen regelt de opamp via een RC-tijd de puls-pauzeverhouding van het kloksignaal. Het resultaat is dat de neergaande flank van het kloksignaal midden in een databit terecht komt. Dit vermindert de kans op leesfouten aanzienlijk.

De resetlogica bestaat uit een paar poorten en stuurt de servo-regelaar, de bandstopdetector en de flipflop van de klok- en data-separator.

Om de 6502-processor in staat te stellen van de MDCR gebruik te

maken moet deze in een bepaalde volgorde signalen aftasten en sturen. Daarvoor zijn in- en uitvoerkanalen nodig. Voor de 6502 en de 6800 kan hiervoor een zogenaamde PIA (Peripheral Interface Adapter) of VIA (Versatile Interface Adapter) uitstekend dienst doen. In dit geval is gekozen voor de PIA 6821 van Motorola, maar ook de VIA 6522 van MOS-Technology kan dienst doen.

Een aantal lijnen van de poort worden geconfigureerd als ingang, een aantal als uitgang. Voor bepaalde toepassingen is het handig als men over twee of meer MDCR's kan beschikken.

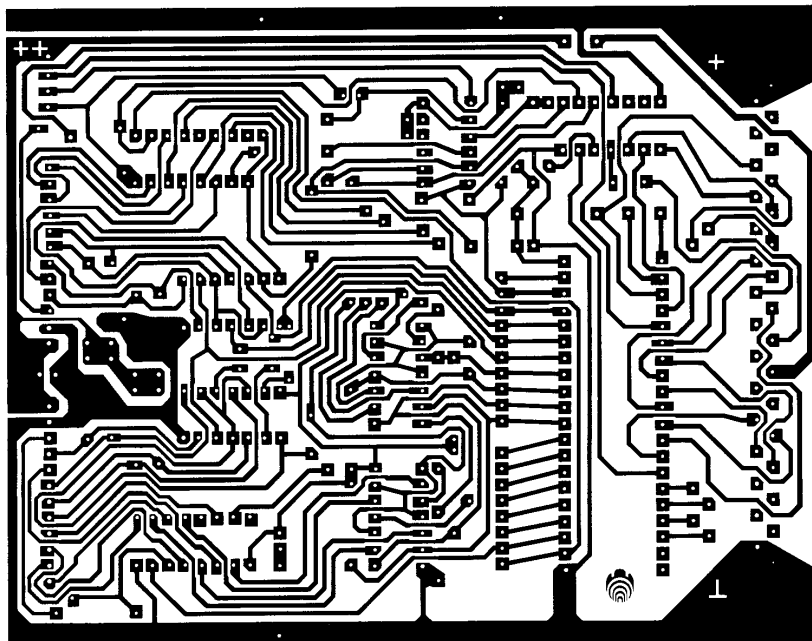
[illegible]

bijvoorbeeld bij een assembleerprogramma. Hierbij kan dan de ene recorder de broncode en de ander de machinecode bevatten. De keuze, welke recorder op een bepaalde tijd wordt gebruikt, wordt door de processor geregeld via een poortuitgang. Met een PIA is dit alles uitstekend te realiseren. Poort B wordt gebruikt voor de diverse in- en uitvoersignalen. Poort A voor de selectie van een MDCR en lijn CB1 voor de kloksynchronisatie. De 6821 is uitgevoerd in NMOS, is TTL-overeenkomstig en werkt met een voedingsspanning van 5 V. Daar de MDCR werkt met CMOS en een voedingsspanning van 12 V, moet een interface worden tussengevoegd als niveau-aanpassing. Tegelijk biedt dit de mogelijkheid de selectie te doen voor twee of meer MDCR's. Hierbij wordt gebruik gemaakt van poorten met opencollector-uitgangen, die een hogere spanning kunnen verdragen. Gekozen is voor de SN74LS26, een viervoudige NAND-poort. Door het toepassen van de zogenoem-

de „wired or“-constructie kunnen meerdere uitgangen parallel worden gezet. Het totale schema is te zien in afb. 2. De aansluiting van de 6821 op de processor geschiedt met een minimaal aantal lijnen. Nodig zijn $\Phi 2$, reset, R/W, A1 tot en met A3, D0 tot en met D7 en een selectielijn. Voor de reset is een aparte constructie gecreëerd. Deze reageert normaal bij een reset van de computer, doch heeft ook een reset van de 6821 tot gevolg, wanneer de voedingsspanning wordt ingeschakeld. Hierdoor is men er altijd zeker van dat de uitgangen van de 6821 in een vaste positie staan, wanneer de voeding wordt ingeschakeld of wanneer een reset wordt gegeven. De MDCR's zijn dan niet geselecteerd en worden ook niet bekrachtigd. Het selectiesignaal wordt verkregen van de adresdecoder, welke of reeds op de processorkaart aanwezig is of uitwendig is aangebracht. Wanneer van de eerder in RB gepubliceerde moederkaart gebruik wordt gemaakt (zie RB, juli 1981) kan de interfaceprint direct in een van de connectoren worden gestoken. De selectie geschiedt nu door

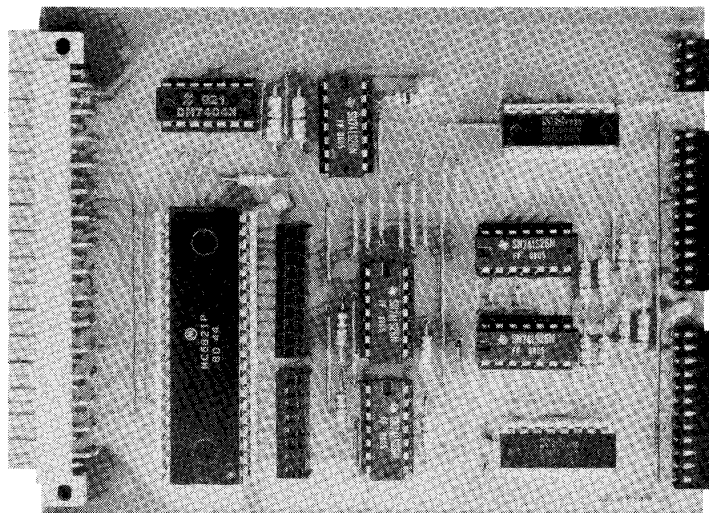
middel van het onderste 1K-selectiesignaal uit een blok van 4K. De adressen van de diverse registers van de PIA zijn dan te vinden vanaf \$X000, waarbij de X een waarde tussen 2 en E hex heeft. De adreslijnen A1 en A2 dienen voor de selectie van de vier interne registers van de PIA, namelijk datarichtingsregister en het dataregister voor respectievelijk poort A en B. Een 6821 heeft verder nog twee selectie-ingangen, te weten CS0 en CS1. Hieraan worden geïnverteerd de adreslijnen A3 en A4 gekoppeld, waardoor wordt voorkomen dat de bedoelde adressen zich onmiddellijk herhalen in de adresruimte. Door middel van draadbruggen kan worden geselecteerd tussen een 6821 of een 6522. Daartoe worden de adreslijnen A0, A1 en R/W aan andere pennen gelegd. De poortuitgangen van de PIA zijn uitgevoerd via een connector. Dit maakt het mogelijk deze interfaceprint ook met een andere PIA te besturen. Wel dient u dan de PIA of VIA van de print te verwijderen. De voedingsspanningen van 5 en 12 V kunnen van het computersysteem worden betrokken. De

Afb. 3 Printontwerp voor de interfaceschakeling, schaal 1 : 1.



spanning van 12 V kan echter ook via een aparte connector van buiten af worden toegevoerd. In dat geval moet draadbrug D worden verwijderd. De signalen reset en $\Phi 2$ komen eveneens via deze connector binnen. Dit is gedaan, omdat deze signalen niet op de bus van de moederkaart aanwezig zijn. De niveauomzetting van TTL naar CMOS en omgekeerd wordt respectievelijk verzorgd door de 74LS26 en de 4050. De 7426 is een viervoudige poort met opencollectoruitgangen, die bestand zijn tegen hoge spanningen. De 4050 is een in dit geval met 5 V gevoede buffer, welke aan de ingang bestand is tegen niveaus van 12 V. Aldus is een goed werkende niveau-omzetter gerealiseerd. De selectie van een bepaalde recorder wordt aangegeven door middel van een LED, welke is verbonden met de buffer achter de uitgangen van poort A. Via een weerstand van 180 Ω dient deze te worden verbonden met de voedingsspanning van 5 V en te worden ge-

Afb. 4 Componentenopstelling van de interface-print.



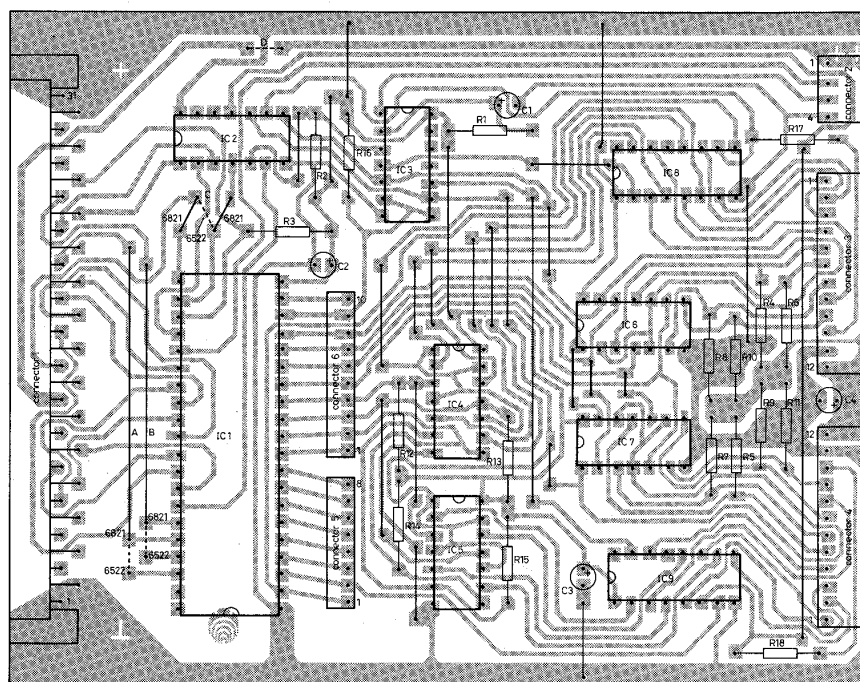
Afb. 5 Compleet gemonteerde print voor de MDCR-interface.

monteerd in de buurt van de bijbehorende recorder.

■ Print

Voor de interface is een print ontworpen, zie afb. 3, waarvan de componenten zijn geplaatst volgens afb. 4. Bij het aanbre-

ngen van de draadbruggen dient te worden vastgesteld of een 6821 of een 6522 wordt toegepast. De software is geschreven voor de 6821, doch de print is ook voorbereid voor een 6522 VIA. Bij A, B en C moeten de draadbruggen worden aangepast. De bovenkant van de print moet zo ver worden afgevild dat de connectoren (mini-module, type 2,5 MBPH, tweemaal 12 polen en



éénmaal 4 polen) er precies omheen passen. Voor de verbinding met de processor wordt een 31-polige connector volgens DIN41617 gebruikt.

Fasecodering

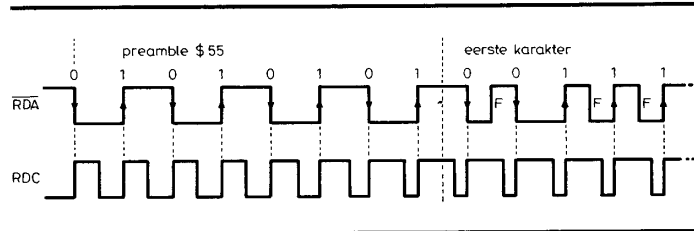
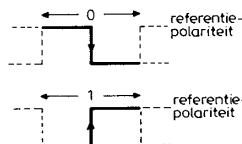
De MDCR maakt gebruik van zogenaamde fasecodering om signalen op de band vast te leggen. Databits, nullen en enen, worden als verandering in magnetiseringspolariteit opgetekend. Anders dan bij audio-cassettecorders wordt de data digitaal opgetekend, de band is of in de ene of in de andere richting gepolariseerd.

Overgangen van polariteit noemt men fluxovergangen. Bij fasecodering wordt een „0” als een fluxovergang van referentie-polariteit naar tegengestelde polariteit opgetekend en een „1” bij precies de omgekeerde fluxovergang. In afb. 6 is dit zichtbaar. Daarbij valt op dat het polariteitsniveau de juiste waarde moet hebben voor een fluxovergang kan worden geschreven. Als bijvoorbeeld twee nullen achter elkaar worden geschreven zal er tussenin een fluxovergang in omgekeerde richting worden geschreven. Dit noemt men fase-fluxovergangen. Bij het teruglezen wordt de fasege-decodeerde informatie door de MDCR-schakeling teruggewonnen zoals boven is beschreven. Een kloksignaal geeft aan wanneer geldige data beschikbaar is.

Datablokken

Data wordt bit voor bit weggeschreven en teruggelezen. Gebruikelijk is om deze verzamelingen bits te organiseren als blokken van databytes met identificatiebytes en een checksum

Afb. 6 Fasegecodeerde schrijfsignalen.



Afb. 7 Gelezen data en bijbehorend kloksignaal van MDCR. Het eerst gelezen karakter, de preamble wordt gevolgd door een gewoon karakter.

voor controle. Voor de MDCR moet aan een dergelijk blok tevens een zogenaamd preamble-karakter (hex 55) worden toegevoegd om de leeselektronica te synchroniseren en een blok worden afgesloten met een postamble (hex 00), zie afb. 7. Dit leidt tot de volgende opbouw van een blok met de elementaire routines:

- Preamblebyte, hex 55.
- Synchronisatiebyte, hex 16, voor extra veiligheid.
- Bloktypebyte, een identificatiebyte van het type blok.
- Bloknummerbyte.
- Aantal bytes in blok, één byte.
- Databytes.
- Checksum, twee bytes.
- Postamblebyte, hex 00.

Een blok bestaat uit 1 tot 256 bytes, zoals blijkt uit de ene byte die het aantal bytes in het blok aangeeft. Bij het ontwerpen van de elementaire routes bleek vanwege tijdsproblemen een keuze te moeten worden gemaakt uit:

- a. Een blok met maximaal 256 bytes en een checksum van twee bytes.
- b. Een blok met maximaal 65536 bytes en een checksum van één byte.

Programmatuur

De twee lijsten (voor een PIA 6820 of 6821) kunnen bij De Muiderkring worden besteld door f 7,50 over te maken op girorekening 83214, ten name van De Muiderkring BV, afd. Verkoop te Bussum, onder vermelding „Mini-digitale cassette-recorder-interface voor de 6502”.

De laatste mogelijkheid leek niet betrouwbaar genoeg. Met blokken van 256 bytes kan, inclusief de noodzakelijke tussenruimten om tussen de blokken te kunnen stoppen, aan iedere kant van de mini-cassette 32K worden opgeslagen in totaal 64K per mini-cassette.

Overzicht van elementaire MDCR-routines

In lijst 1 zijn de geassembleerde elementaire routines te vinden. De eigenlijke elementaire routines worden bereikt door een sprongtabel, waarmee de code van lijst 2 begint. Deze twee lijsten zijn bij De Muiderkring te bestellen. Alle routines zijn uitgevoerd als subroutines. De sprongtabel verwijst naar de volgende elementaire routines:

TINIT, initialisatie van PIA en stoppen van alle MDCR-acties.
TSTOP, stoppen van acties met geselecteerde recorder.
TREAD, lees blok van band.
TWRITE, schrijf blok naar band.
TWEDG, schrijf einde van data-ruimte, een leeg stuk op de band.
TRWND, spoel band terug.
TCIP, test of er een cassette in de MDCR aanwezig is.
TBET, test of het begin of einde van de band is bereikt.
TWEN, test of de cassette tegen schrijven is beschermd doordat de plug er uit is verwijderd.
TSELD, selecteer een MDCR.
PDELAY, programmeerbare vertraging in stappen van 60 ms.

In lijst 1 is bij elke elementaire routine aangegeven wat bij het aanroeren moet worden meegegeven: 'In:', wat bij terugkeer wordt meegegeven: 'Uit:', en wat door de routine wordt gebruikt: 'Gebruikt:'. Bij complexe routines is tevens aangegeven welke

subroutines er vanuit de routine worden aangeroepen. Deze informatie is afdoende om gebruik te kunnen maken van de elementaire routines.

Alle routines geven bij terugkeer met de Z-vlag aan of de actie met succes is geëindigd. Z = 0 betekent geen fout, anders bevat FOUTNR de code van de geconstateerde fout. Gedefinieerd zijn de volgende foutcodes:

- 1 is time-out bij lezen van bit.
- 2 is cassette niet aanwezig.
- 3 is checksum van gelezen blok is fout.
- 4 is cassette is beschermd tegen schrijven.
- 6 is einde van dataruimte gevonden in plaats van blok.
- 7 is ander bloknummer gevonden dan in aanroep van TREAD gespecificeerd. Deze foutcode treedt niet op met Z = 0, maar met C = 1 na terugkeer uit TREAD.
- 8 is begin of einde van band bereikt.

■ Schrijfroutine

Het schrijven van een blok is de meest kritische routine van de elementaire routines. Het datasignaal, dat naar de MDCR wordt gestuurd, moet aan vrij zware eisen voldoen. Zo moet de totale tijd per bit 166,7 microseconden bedragen (voor een overdrachtsnelheid van 6000 baud) en de fluxovergangen moeten op de juiste ogenblikken plaatsvinden. Het datasignaal wordt geheel via software opgewekt. De juiste tijden worden bereikt door wachtlopen en een precies uitgetelde totale instructietijd. Dit is vrij kritisch en daarom is het verstandig niets aan de schrijfroutine zelf te veranderen. Ook het plaatsen van de routines „Schrijf blok”, XDELY en CDELY, en „Schrijf byte” binnen één geheugenpagina is aan te bevelen om variabele instructietijden van spronginstructies te voorkomen. Op zich bevatten de schrijfroutines weinig opmerkelijke zaken. De boven beschreven blokopbouw wordt er mee gegenereerd. De tijdsproblemen maken ze echter minder doorzichtig.

■ Leesroutine

De routine om een blok te lezen, verricht precies de omgekeerde handeling van de schrijfroutine. In plaats van bits naar de band te schuiven worden ze nu in het geheugen geschoven. De „Lees blok”-routine probeert eerst een begin van een blok te vinden. Voordat echter naar de ingelezen data en klok kan worden gekeken moet de lees-elektronica worden gereset. Na een begin van een blok te hebben gevonden worden de identificatiebytes zoals bloktype en bloknummer ingelezen. Het bloknummer wordt vergeleken met het opgegeven verlangde bloknummer en bij niet gelijk zijn, wordt het aanwezige blok overgeslagen. Byte voor byte wordt de data binnengehaald en aan het eind de gelezen checksum vergeleken met de berekende checksum. Bij het lezen van een bit in de routine „Lees bit” wordt een time-out-teller gebruikt. Als binnen een redelijke tijd geen bit komt, is er duidelijk iets mis en wordt het lezen onderbroken.

■ Testroutine

Lijst 2 bevat een eenvoudige testroutine voor de elementaire routines. Een aantal routines uit de KIM-monitor worden daarin aangeroepen, essentieel zijn deze routines niet en andere computers hebben ongetwijfeld overeenkomstige routines. De testroutine heeft als functie een opgegeven aantal blokken naar en van de band te transporteren. De grootte van de blokken is voor de eenvoud vastgelegd op 256 bytes. Het „blok type”-byte wordt gebruikt om het begin van de band aan te geven (00), gewoon datablok (01) en laatste blok (FF). De testroutine demonstreert in eerste instantie het gebruik van de elementaire routines, zoals de argumenten die moeten worden meegegeven en wat er terug wordt gegeven. Daarnaast geeft de testroutine een mogelijkheid de elementaire routines te testen. Tijdens het lezen of schrijven van blokken worden de blokken geïdentificeerd op de terminal en een fout gemeld met foutnummer.

■ Tijden

De elementaire routines, zoals voor lezen en schrijven van een blok, vereisen bij het aanroepen in het X-register een tijdteller. In plaats van vaste waarden te kiezen kan op deze manier voor elke toepassing de optimale tijden voor ruimte tussen blokken worden gekozen. De volgende tijden zijn zo gekozen dat het mogelijk is bij zowel lezen als schrijven tussen elk blok te stoppen. Het is mogelijk de blokken op de band dichter tegen elkaar te plaatsen, als van het stoppen tussen blokken wordt afgezien.

Schrijftijden

Het eerste blok vereist meer tussenblokruimte dan gewone blokken vanwege het aanloopgedeelte.

Eerste bloktussentijd is 1000 ms, met X = hex 10 wordt 960 ms gehaald. Een gewoon blok vereist 300 ms, wat met een waarde van X = 5 wordt bereikt.

Leestijden

Opgeven van tussenbloktijden bij lezen is nodig omdat, vlak voordat het blok wordt gelezen, de leeselektronica moet worden gereset.

Eerste blok tussentijd is 480 ms, met X = $8 \times 60 \text{ ms} = 480 \text{ ms}$. Gewone blok tussentijd is 60 ms, x = 1 voldoet.

Na de reset wordt een bit-time-out-telling gestart, een waarde van A = hex FF voldoet.

Einde datatussenruimte

Als het laatste blok is weggeschreven, is het nodig om een stuk band achter dat blok schoon te wissen. Een schrijftijd van 1000 ms voldoet met X = hex 40 ($64 \times 15 = 960 \text{ ms}$).

De bij de MDCR behorende documentatie van Philips bevat een meer uitgebreide discussie over deze tijden. De bovengenoemde tijden voldoen betrouwbaar met de elementaire routines, maar misschien is er nog meer uit te halen of is het nodig blokken te herschrijven tussen al bestaande blokken.