

HANDLEIDING  
VOOR DE  
INSTRUCTOR 50

© 1980 N.V. Philips' Gloeilampenfabrieken  
EINDHOVEN — Nederland

Deze publikatie mag niet geheel of gedeeltelijk worden  
vermenigvuldigd, geregistreerd en/of openbaar gemaakt  
door middel van druk, fotokopie, microfilm of op welke  
wijze dan ook, zonder voorgaande schriftelijke toestem-  
ming van N.V. Philips' Gloeilampenfabrieken, Eindhoven.

## INHOUD

<b>Inleiding</b>	5
Stroomvoorziening	5
Opbouw van het bedieningspaneel	5
Functiemodes	5
Het inbrengen van programma's, adresseren, data enz.	6
Het corrigeren van fouten	6
Het geheugen en de I/O organisatie	6
<b>Toetsen, schakelaars en indicators</b>	7
Het besturingstoetsenveld	7
Schakelaars voor het invoeren van externe data	8
Indicators voor het uitvoeren van externe data	8
DIRECT/INDIRECT-schakelaar	8
Poort-adres selectieschakelaar	8
BREEKPUNT-toets	8
Snelheid	8
<b>Bediening van de INSTRUCTOR 50</b>	9
Inleiding	9
Lezen en veranderen van CPU registers en PSW	9
Lezen en veranderen van geheugencellen	10
Snel laden van het geheugen	10
Het zichtbaar maken en veranderen van de inhoud van het instructie-adresregister	11
Stap-voor-stap functie	11
Het aanbrengen van een BREEKPUNT in het programma	12
Meldingen van fouten bij het gebruik van een breekpunt	12
RUN-toets	12
RESET-toets	13
Fout-meldingen	13
<b>Het gebruik van de INSTRUCTOR 50</b>	14
Het gebruik van INTERRUPTS	14
Doorverbindingen in veld A	16
Het gebruik van de I/O schakelaars en LED's	16
FLAG- en SENSE-methodiek	17
NON-EXTENDED I/O-methodiek	17
EXTENDED I/O-methodiek	17
MEMORY MAPPED I/O-methodiek	17
Keuze klokfrequentie – S100 bus	17
Keuze van de spanningsbron	18
<b>Ingebouwde subroutines</b>	18
MOVE-subroutines	19
DISPLAY-subroutine	19
USER DISPLAY-subroutine	20
NIBBLE-subroutine	21
INPUT DATA-subroutine	22
MODIFY DATA-subroutine	22
<b>Systeem-expansie</b>	23
<b>De interface naar de cassette-recorder</b>	26
Schrijven op de cassette	26
Het afregelen van de cassette	27
Lezen van de cassette	27

*Deze handleiding bevat de gegevens die nodig zijn om met een "INSTRUCTOR 50" oefenprogramma's te kunnen testen en uitvoeren.*

*De INSTRUCTOR 50 is gebaseerd op de microprocessor 2650. Alle instructies van deze processor kunnen met behulp van de INSTRUCTOR 50 uitgevoerd worden; ook de instructies die de in- en uitvoer van gegevens bewerkstelligen. Voor dit laatste is geen additionele apparatuur vereist, aangezien het paneel de nodige toetsen bevat om data in te voeren, benevens een scherm met acht symboolplaatsen, om data zichtbaar te maken. Daarnaast zijn nog acht schakelaars met acht speciale lampen aanwezig om de communicatie met de buitenwereld te simuleren. Verder zijn er een aantal functietoetsen waarmee het mogelijk is, een proces dat onder invloed van een programma wordt uitgevoerd, te onderbreken, te wijzigen of bepaalde gegevens zichtbaar te maken. Via de in- en uitgang van de cassette-recorder kan informatie op cassetteband worden bewaard en weer teruggelezen in de INSTRUCTOR 50.*

*Ten behoeve van de lezers die zich willen verdiepen in de werking van de INSTRUCTOR 50, bevatten de bijlagen gedetailleerde gegevens over de opbouw van de INSTRUCTOR 50 en het MONITOR-programma.*

## INLEIDING

### Stroomvoorziening

De INSTRUCTOR 50 wordt gevoed via een transformator, aangesloten op het 220 V/50 Hz wisselstroomnet. Het snoer aan de transformator is voorzien van een steker die aan de achterzijde van de INSTRUCTOR 50 in de stekerbuis moet worden gestoken. Er is geen speciale schakelaar aanwezig voor het in- en uitschakelen van de INSTRUCTOR 50. Bij het inschakelen moet het woord "HELLO" verschijnen op het beeldscherm. Indien dit niet het geval is, moet de toets "MON" ingedrukt worden.

### Opbouw van het bedieningspaneel

Aan de bovenzijde van de INSTRUCTOR 50 (zie fig. 1) bevinden zich drie velden met schakelaars (toetsen), benevens het scherm waarop symbolen (zie fig. 2) zichtbaar kunnen worden gemaakt. Het rechtse toetsenbord bevat zestien toetsen, genummerd 0 t/m 9 en A t/m F. Deze toetsen worden gebruikt om hexadecimale symbolen aan de INSTRUCTOR 50 aan te bieden voor de invoer van programma's, data, het raadplegen van de toestand van de microprocessor, enz. Het middelste toetsenbord bevat een aantal functietoetsen, die het inbrengen en uitlezen van data moeten begeleiden. Tevens worden deze toetsen gebruikt om programma's te onderbreken en vast te stellen hoe de uitvoering van een programma verloopt. Met behulp van deze toetsen en het veld van de hexadecimale toetsen worden de opdrachten aan de instructor gegeven; de instructor zal via het scherm aan de gebruiker de gewenste informatie verstrekken. Het linkse veld van acht schakelaars en acht lampjes dient ter simulatie van een in- en uitvoer van een microprocessor 2650 systeem. De soort van in- en uitvoerapparatuur kan

door middel van de middelste schakelaar worden gekozen. Hierop wordt nog nader teruggekomen bij de behandeling van de in- en uitvoer van data. Aan de onderzijde is de microprocessor voorzien van een aantal gaten. Door deze gaten kan men bepaalde snelheden van de machinewerking kiezen, de wijze van interruptie aanbrengen en speciale voorzieningen treffen voor de uitbreiding van de INSTRUCTOR 50. Aan de achterzijde is de montageplaat zodanig uitgevoerd dat hierin een steker kan worden gestoken, zodat men later desgewenst speciale in- en uitvoerapparatuur op de microprocessor kan aansluiten.

P.S. Bij het uitvoeren van de eerste oefeningen moet de schakelaar aan de onderzijde in de stand "KEYBOARD" staan.

### Functiemodes

De instructor heeft twee functiemodes. De eerste wordt de "MONITOR-MODE" genoemd en de andere de "EXECUTION-MODE". De MONITOR-MODE wordt ingeschakeld zodra men de INSTRUCTOR 50 voor de eerste maal inschakelt of als men de "MON"-toets indrukt. Op het scherm verschijnt dan het woord "HELLO". Gedurende deze MONITOR-MODE kunnen de volgende handelingen verricht worden:

1. Het inbrengen, lezen en veranderen van een programma.
2. Het schrijven van een programma naar de band van een cassetterecorder.
3. Het lezen van een band van een cassetterecorder.
4. Op het scherm weergeven van de inhoud van de registers van de microprocessor 2650 resp. het veranderen van deze inhoud.

5. Idem voor het PROGRAM-STATUS-WORD (PSW).
6. Het onderzoeken en veranderen van de inhoud van het geheugen.
7. Het onderzoeken en veranderen van de inhoud van het instructie-adresregister (IAR).
8. Het specificeren en onderzoeken van een breekpunt in het programma.
9. Het stap-voor-stap laten uitvoeren van een programma.

De EXECUTION-MODE wordt gekozen door het indrukken van de "RUN"-toets, de "STEP"-toets of de "RESET"-toets (RST-toets). Als de RUN-toets wordt ingedrukt, begint de programma-uitvoer met die instructie waarvan het adres aanwezig is in het instructieadres-register (IAR). Drukt men de STEP-toets in, dan wordt een enkele instructie gelezen en uitgevoerd, waarna de INSTRUCTOR 50 weer in de MONITOR-MODE terugkeert. Het indrukken van de RST-toets heeft tot gevolg dat de instructor zijn activiteit beëindigt en dat het instructie-adresregister in de stand H'0000' wordt gezet (H duidt aan, dat de notatie in hexadecimale code geschiedt).

### Het inbrengen van programma's, adressen, data, enz.

Zoals eerder vermeld, wordt voor het inbrengen van instructies het rechtse hexadecimale veld toetsen gebruikt. In fig. 3 wordt het verband weergegeven tussen de decimale, binaire en hexadecimale getallen. Op de instructor zijn alle instructies in hexadecimale code weergegeven, waarbij rekening is gehouden met registernummers, de zogenaamde pagina's, enz. Als data of een adres aan de INSTRUCTOR 50 kunnen worden toevertrouwd, verschijnt uiterst links op het scherm een punt. Ontbreekt deze punt, kan men geen gegevens inbrengen. Wenst men dit wel te doen, dan moet de functie-toets (zie later) eerst ingedrukt worden.

### Het corrigeren van fouten

Zodra men met behulp van de zestien hexadecimale toetsen gegevens in de machine brengt, verschijnen deze gegevens op het scherm. Elk nieuw aangeslagen element verschijnt rechts op het scherm terwijl het bestaande patroon van symbolen naar links verschuift. Op deze wijze kan men een eventuele fout corrigeren, door de gewenste waarde eenvoudig opnieuw aan te slaan. Het foutieve gegeven verdwijnt

dan, geleidelijk naar links schuivend, van het scherm en de nieuwe, juiste data verschijnt in de plaats daarvan.

### Oefening

Druk op de toets "MON". De INSTRUCTOR 50 zal antwoorden met het bericht "HELLO" op het scherm. Druk vervolgens op de toets "MEM". Nu verschijnt ".AD.=". De INSTRUCTOR 50 is nu gereed om een geheugenadres in hexadecimale code in ontvangst te nemen. Wil men nu b.v. het adres 0120 selecteren en slaat men abusievelijk 0121 aan, dan zal men op het scherm waarnemen ".Ad.= 0121".

Deze fout is eenvoudig te corrigeren door de toetsen 0, 1, 2, 0 achtereenvolgens in te drukken. Voer deze correctie zelf uit.

### Het geheugen en de I/O-organisatie

Behalve in de microprocessor 2650, is ook in de INSTRUCTOR 50 een geheugen aanwezig. In dit geheugen is de functie van de INSTRUCTOR 50 door middel van een ROM vastgelegd. Voorts bevat dit geheugen een aantal plaatsen die zijn gereserveerd voor het verrichten van de functies van de INSTRUCTOR 50. Om deze reden kunnen niet alle geheugenplaatsen vrij gebruikt worden. De indeling van het geheugen is in fig. 4 weergegeven. De indeling van het geheugen zal niet direct duidelijk zijn. Voorlopig is het echter voldoende te weten dat het aan de gebruiker ter beschikking staande geheugen als eerste adres H'0000' en als laatste adres H'01FF' heeft. Het kan worden voortgezet in het gebied H'0200' tot H'0FFF' en van H'2000' tot H'7FFF'. Dit moet geschieden via de stekeraansluiting aan de achterzijde van de INSTRUCTOR 50.

Het gebied tussen adres H'17C0' en H'1FFF' wordt gebruikt voor het programma van de INSTRUCTOR 50 zelf. Hierin bevindt zich het z.g. "MONITOR"-programma.

### Opmerking

Het wordt ontraden in programma's gebruik te maken van de geheugencel H'0FFF', daar deze gereserveerd is voor I/O operaties ten behoeve van de programma's die men zelf schrijft. Voor dat doel kan men de cel dus wel gebruiken.

De zogenaamde "EXTENDED I/O" beslaat de adressen H'00' tot H'F7'. In de INSTRUCTOR 50 is echter alleen adres H'07' bruikbaar.

## TOETSEN, SCHAKELAARS EN INDICATORS

### Het besturingstoetsenveld

Het toetsenbord met de besturingstoetsen heeft ten doel het inbrengen van programma's te begeleiden, het testen hiervan, het opvragen van informatie uit de INSTRUCTOR 50 en het schrijven, resp. lezen van het ingebrachte programma op een externe cassetterecorder.

De functies van de voor dit doel bestemde toetsen zijn hieronder opgesomd.

*toets*      *functie*

**MON** De "MONITOR"-toets wordt gebruikt om een programma dat op een bepaald moment werkzaam is, te beëindigen, en veroorzaakt een sprong naar het programma van de monitor. De machine geeft dit aan door het woord "HELLO" te laten verschijnen. Voordat men andere commando's geeft, is het indrukken van deze toets gewenst en doorgaans zelfs vereist.

**MEM** Door het indrukken van de "MEMORY"-toets krijgt men toegang tot het geheugen van de INSTRUCTOR 50 voor het inbrengen en opvragen van instructies en data. De INSTRUCTOR 50 geeft als reactie ".Ad.=" Dit betekent dat de INSTRUCTOR 50 om een adres vraagt, d.w.z. men moet nu het adres invoeren waarvan men de data wil onderzoeken, resp. nieuwe data wenst in te brengen of bestaande data wenst te veranderen.

**REG** De "REGISTER"-toets geeft toegang tot de registers van de microprocessor 2650, de programmateller, of tot de inhoud van het z.g. PSW. De INSTRUCTOR 50 reageert met het geven van "r=". Om aan te geven welk register, resp. welk deel van het PSW, men na het indrukken van deze toets kiest, wordt een van de hexadecimale toetsen ingedrukt.

**ENT NXT** Deze "ENTER NEXT"-toets dient om eerder ingebrachte informatie in het geheugen resp. de registers op te slaan. De informatie kan er betrekking op hebben of een adres dan wel data, of wel een wijziging van data, inclusief de instructies in het geheugen ingevoerd wordt. Voorts kan men met behulp van deze toets snel de inhoud van het geheugen of de registers nagaan. Op het gebruik van deze toets in samenwerking met de toetsen REG en MEM wordt nog nader ingegaan.

**STEP** In de monitor-mode is het mogelijk het programma, instructie na instructie, uit te laten voeren door deze toets telkenmale in te drukken. Op het scherm verschijnt nu het adres van de volgende instructie, die daarna wordt uitgevoerd. Op deze wijze kan men in zijn eigen programma nagaan of de uitvoering aan de verwachtingen voldoet.

**RUN** Als deze toets wordt ingedrukt, zal een programma dat om een of ander reden onderbroken is, b.v. door het overgaan in de MONITOR-MODE bij het drukken van de toets "MON", wederom verder worden uitgevoerd. De instructie, die als eerste wordt uitgevoerd, wordt gegeven door de inhoud van het instructie-adres register. De uitvoering van het programma wordt voortgezet tot een van de onderstaande gebeurtenissen plaats heeft:

1. Het BREEKPUNT is bereikt.
2. De toets "MON" of "RST" wordt ingedrukt.
3. Het programma voert een "HALT"- of een "WRTC"-instructie uit.

**RST** Drukt men de "RESET"-toets in, dan wordt de instructie die in uitvoering is, beëindigd en een RESET-sigitaal gegeven naar de microprocessor 2650. Dit heeft tot gevolg dat het programma opnieuw wordt begonnen met het adres H'000'. Indien deze toets wordt ingedrukt, komt het systeem niet in de MONITOR-MODE.

**SENSE** De "SENSE"-toets is een nabootsing van een extern signaal, dat aan de SENSE-pen van de 2650 kan worden toegevoerd. Dit signaal kan nu met de hand worden nagebootst. SENSE-input is in de normale toestand een logische "0". Het indrukken van de SENSE-toets zal tot gevolg hebben dat een logische "1" wordt gegeven.

**INT** De microprocessor 2650 kan worden geïnterrupteerd tijdens de uitvoering van een programma. Deze "INTERRUPT" kan met behulp van deze toets worden nagebootst. Het resultaat is een "VECTORED INTERRUPT" naar de plaats H'07' in het geheugen. Het hangt af van de stand van de "DIRECT/INDIRECT"-schakelaar op het bedieningspaneel, of het programma, dat op grond van deze interruptie moet worden uitgevoerd, begint op de locatie H'07',

dan wel dat deze plaats verwijst (indirecte adressering) naar een andere locatie. De INSTRUCTOR 50 is voorzien van een schakelaar die via een gat in de bodem toegankelijk is. Bij de normale problemen dient deze schakelaar in de stand "KEYBOARD" te staan. Alleen als men externe interrupts vanuit het wisselspanningsnet toestaat, dient de schakelaar in de stand "A.C.LINE" te staan.

**BKPT** De "BKPT"-toets geeft de opdracht om een "BREEKPUNT" te specificeren. Slaat men het adres van dit breekpunt aan via het hexadecimale toetsenveld, dan wordt de inhoud van de betreffende geheugencel op het scherm weergegeven zodra dit breekpunt in het programma wordt bereikt.

**WCAS** Het indrukken van deze "WRITE CASSETTE"-toets draagt er zorg voor dat het programma dat in het INSTRUCTOR 50 geheugen is geregistreerd, wordt overgedragen op de band van de cassetterecorder.

**RCAS** Het indrukken van de "READ CASSETTE"-toets bewerkstelligt dat de data van de cassette-recorder wordt overgenomen in het geheugen van de INSTRUCTOR 50.

### Schakelaars voor het invoeren van externe data

De microprocessor 2650 kan d.m.v. I/O apparatuur met zijn omgeving corresponderen. Een van de I/O apparaten is nagebootst met behulp van acht poortschakelaars. Deze dienen om data aan de 2650 te verstrekken en wel in de vorm van een z.g. "byte". Wordt een leesopdracht met een bijbehorend adres aan de 2650 verstrekt, dan zal de stand van deze schakelaars in de 2650 worden overgenomen.

### Indicators voor het uitvoeren van externe data

Voor het uitvoeren van data naar zijn omgeving is eveneens I/O apparatuur noodzakelijk. De uitgifte is nagebootst m.b.v. acht poort-LED's, aangesloten op acht FLIP-FLOPS. Deze FLIP-FLOPS kunnen bij een uitvoeringshandeling met de daarbij behorende adressering in de stand '0' of '1' worden geplaatst, met als gevolg dat de LED uit, resp. aan is.

### DIRECT/INDIRECT-schakelaar

Als de microprocessor 2650 geïnterrupteerd wordt, zal deze een sprong maken naar een z.g. "SUBROUTINE". In het geval van de INSTRUCTOR 50 zal dit de plaats H'07' zijn. Op deze plaats begint het serviceprogramma van de subroutine, resp. bevindt zich de eerste byte van de verwijzing voor de indirecte adressering naar de plaats waar de subroutine begint. De gebruiker van de INSTRUCTOR 50 heeft dus niet de mogelijkheid om bij een interrupt er zonder meer voor te zorgen dat de interrupt-routine van een bepaald adres uit start. Of de adressering direct dan wel indirect is, wordt aangegeven door de stand van de "DIRECT/INDIRECT" schakelaar. Als externe I/O apparaten met de INSTRUCTOR verbonden worden d.m.v. de stekker aan de achterzijde, dan kan een ander interrupt-adres gebruikt worden ("EXTENDED I/O").

### Poort-adres selectieschakelaar

Deze schakelaar kiest de wijze, waarop de parallel geschakelde in- en uitvoerapparatuur wordt geadresseerd. De schakelaar heeft drie standen:

"MEMORY OFFF"

"EXTENDED I/O PORT 07"

"NON EXTENDED DATA PORT"

### BREEKPUNT-toets

Een hulpmiddel bij het controleren van programma's op de correcte uitvoering is het z.g. BREEKPUNT. Dit breekpunt wordt opgeroepen met behulp van de "BKPT"-toets op een wijze, die later nog zal worden behandeld. De gebruiker van de INSTRUCTOR 50 draagt met behulp van deze faciliteit het adres van het BREEKPUNT aan de INSTRUCTOR 50 over. Dit adres geeft aan, waar de programma-uitvoering moet worden onderbroken. Zodra het instructie-adresregister deze waarde heeft bereikt, wordt de EXECUTION MODE beëindigd en gaat de machine over tot de MONITOR MODE. Daarbij geeft het scherm het adres van de eerst volgende uit te voeren instructie aan. Nadat de uitvoering van het programma is gestopt, kan men weer continueren door de toets "RUN" in te drukken of herhaaldelijk de toets "STEP" in te drukken, waarna, zoals bekend, telkens één instructie wordt uitgevoerd.

### Snelheid

De INSTRUCTOR 50 is voorzien van een 3,579545 MHz kristal voor het opwekken van de zeer nauwkeurige klokfrequentie.

## BEDIENING VAN DE INSTRUCTOR 50

### Inleiding

De INSTRUCTOR 50 kan diverse opdrachten uitvoeren. De eerste belangrijke opdracht is het laden van instructies en data in het geheugen. Een tweede opdracht is die van het laden van registers. Dit laden kan geschieden gedurende de MONITOR MODE. Dit betekent, dat als, tijdens de uitvoering van een programma, de INSTRUCTOR 50 door een druk op de toets "MON" in de MONITOR MODE wordt gebracht, men de inhoud van de registers kan raadplegen en ook wijzigen. Verder kan men het instructie-adresregister (IAR) zichtbaar maken en eventueel wijzigen. Andere commando's zijn:

- Zichtbaar maken en veranderen van het geheugen
- Zetten van het BREEKPUNT
- Stap-voor-stap uitvoeren van de instructies
- Starten van het programma
- Het geheel in de 0-stand terugzetten (RESET).

In het nu volgende deel zullen al deze functies behandeld worden aan de hand van de betreffende COMMANDO TOETS. In de tekst en op de bijbehorende tekeningen geven de kleine letters gevolgd door een haakje het verband aan tussen de tekst en de plaats op de tekening, waarop deze betrekking heeft.

### Lezen en veranderen van CPU-registers en PSW

#### Functie

Dit commando maakt het mogelijk de inhoud van de registers en het PSW (PROGRAM STATUS WORD) te lezen en zo nodig te wijzigen.

#### Procedure

1. Deze functie wordt ingeleid door in de MONITOR MODE, dus na het indrukken van de "MON"-toets, de knop "REG" in te drukken. Zie fig. 5a). Dit heeft tot gevolg dat op het scherm verschijnt "r=".

Voer oefening 1 uit!

De INSTRUCTOR 50 wacht nu op een nummer dat bij een bepaald register of bij een deel van het PSW behoort. Deze nummers zijn:

nummer	register
0	R0
1	R1, BANK 0
2	R2, BANK 0
3	R3, BANK 0
4	R1, BANK 1
5	R2, BANK 1
6	R3, BANK 1
7	PSW UPPER
8	PSW LOWER

2. Zodra men een van deze getallen aanslaat, in ons voorbeeld b), dan verschijnt op het scherm .r4=A8. Dit is slechts een voorbeeld, en kan op een bepaald moment de inhoud zijn van, in dit geval, register 1 in bank 1. De A8 is hier de weergave van de inhoud van dit register, doch die inhoud is afhankelijk van b.v. het starten van de machine door op RUN te drukken of door het inschakelen, waarbij de inhoud van de registers en de geheugenplaatsen willekeurige waarden aannemen.

Verricht oefening 2!

3. Is men met deze inhoud tevreden, dan kan men de knop "ENT NXT" indrukken, zoals onder c). Nu verschijnt automatisch het nummer en de inhoud van het volgende register op het scherm. In ons geval dus register 5, d.w.z. register 2 in bank 1. De inhoud blijkt hier B6 te zijn. Drukt men wederom op ENT NXT d), dan verschijnt op het scherm .r6=A4 Zo kan men doorgaan: drukt men wederom e), dan verschijnt op het scherm .PU=0A Dit is de inhoud van het PSW UPPER. Drukt men vervolgens weer op het ENT NXT f) dan verschijnt .PL=21 Dit is de inhoud van het PSW LOWER. Hieruit blijkt dat het totale Program Status Word is: H'0A21'. Het hierboven behandelde heeft dus uitsluitend betrekking op het nagaan van de inhoud van de REG 4,5,6,7 en 8, die staan voor resp. R1 BANK 1, R2 BANK 1, R3 BANK 1, PSW UPPER en PSW LOWER.
4. Het veranderen van de inhoud van de registers, resp. het Program Status Word UPPER en LOWER geschiedt op een iets andere wijze. Zodra een register gekozen is of de "ENT NXT" toets is ingedrukt, blijkt links op het scherm een punt te verschijnen. Dit betekent dat de inhoud van het betreffende register, resp. PSW-deel, nog te wijzigen is, voordat de "ENT NXT" toets wordt ingedrukt. Deze punt treedt in fig. 5 op in de gevallen b) t/m f).

We zullen nu behandelen hoe men de inhoud van een register kan wijzigen. We beginnen weer (zie fig. 6) met het indrukken van de toets "REG", zoals onder a) is weergegeven. Vervolgens wordt toets 4 ingedrukt b). Op het scherm verschijnt nu .r4=A8. De inhoud A8 is, evenals in de vorige figuur, slechts een voorbeeld. Nu dienen we de inhoud van dit register (register 1 bank 1) te wijzigen. Dit gebeurt b.v. door het indrukken van de toets E, te weten het meest significante hexadecimale cijfer dat we in het register wensen te plaatsen. Dit is weergegeven onder c). Het scherm vertoont nu .r4=E.

De inhoud van het register is naar links geschoven en op de rechtse plaats op het scherm is de E verschenen. Als dan



de volgende hexadecimale digit wordt ingevoerd, en wel door toets 2 in te drukken, dan verschijnt op het scherm .r4=E2. Nu is de inhoud van register 1 bank 1 gelijk aan H'E2'. Door nu weer "ENT NXT" in te drukken, wordt de inhoud van R4 bevestigd en definitief gemaakt. Het scherm vertoont nu automatisch de inhoud van het volgende register, in ons geval .r5=B6 De inhoud hiervan kunnen we op overeenkomstige wijze veranderen.

Speciale aandacht wordt gevestigd op de mogelijkheid om de 2 bytes PSW U EN PSW L van het Program Status Word op deze wijze te wijzigen.

Verricht de boven omschreven handelingen zelf op de INSTRUCTOR 50!

Verricht oefening 3!

Verricht oefening 4!

### Lezen en veranderen van geheugencellen

#### Functie

Door de "MEM" toets in te drukken, wordt toegang tot het geheugen verkregen. Bij het bedienen van deze toets (zie fig. 7 a) verschijnt op het scherm .Ad.=, d.w.z. de INSTRUCTOR 50 is bereid om het adres via het hexadecimale veld in ontvangst te nemen. Dit adres verwijst naar een cel in het geheugen, die 1 byte groot is. De bytes worden zoals bekend door twee hexadecimale cijfers weergegeven. Nu kan het meest significante hexadecimale cijfer ingevoerd worden. Indien deze (meest significante) cijfers 0 zijn, dan kan men ze eenvoudig weglaten.

#### Procedure

In figuur 7 wordt de geheugencel H'0042' opgeroepen. Daar er twee leidende nullen zijn, is het niet nodig deze aan te slaan en kunnen we als eerste cijfer met de 4 volstaan. Dit is weergegeven onder b). Vervolgens kan de minst significante digit, de 2, ingevoerd worden, als onder c) is weergegeven. Door de toets "ENT NXT" d) in te drukken, toont het scherm het volledige adres, voorafgegaan door een punt en de inhoud van de cel die met dit adres correspondeert. De inhoud van de cel is hier aangenomen als H'A5'. Door "ENT NXT" weer in te drukken, verschijnt het adres en de inhoud van de volgende geheugencel op het scherm, hier weergegeven als .0043 B7, zie e). Zo kan men doorgaan met het verifiëren van de inhoud van de geheugencellen.

Wenst men de inhoud van de geheugencellen niet alleen na te gaan maar ook te wijzigen, dan dient deze wijziging verricht te worden voordat "ENT NXT" ingedrukt wordt. Het begin van het wijzigen van de geheugencelinhoud is analoog aan die van het lezen ervan. In fig. 8 zijn de nodige stappen weergegeven.

De LAAD PROCEDURE begint op dezelfde wijze als het lezen, namelijk door het indrukken van de toets "MEM", zie a). Op het scherm verschijnt .Ad.=, hetgeen betekent dat de INSTRUCTOR 50 om een adres voor de selectie van een geheugencel vraagt. Stel dat weer geheugencel

H'0042' wordt verlangd. Nu wordt toets 4 ingedrukt, zoals onder b) is weergegeven. Vervolgens kan, zoals vermeld onder c), de 2 ingevoerd worden. Door indrukken van "ENT NXT" worden deze aangeslagen gegevens in het geheugen opgenomen, zoals onder d) is weergegeven. De inhoud van de geheugencel is b.v. H'A5'. Laten we aannemen dat de inhoud van deze cel gewijzigd moet worden in H'64'. Het meest significante hexadecimale cijfer wordt nu aangeslagen, namelijk de 6, zie e). Vervolgens wordt de 4 aangeslagen, zie f). Heeft men een fout gemaakt, dan kan men deze eenvoudig herstellen door opnieuw de 6 en de 4 in te drukken. Men kan daarmee, indien men zich weer vergist, zolang doorgaan tot het scherm de juiste hexadecimale waarde H'64' toont. Om te bevestigen dat dit de definitieve waarde is die men in de geheugencel wenst te registreren, drukt men de toets "ENT NXT" in, zie g). Zodra men deze handeling heeft uitgevoerd, wordt het nummer van de volgende geheugencel en de inhoud van deze cel op het scherm zichtbaar gemaakt. Wenst men de byte van deze cel te veranderen, dan kan men dit volgens de hiervoor beschreven procedure verrichten.

Als men een correctie in het geheugen wenst uit te voeren, kan men dit dus eenvoudig doen door de toets "MEM" in te drukken, het adres te verstrekken, "ENT NXT" in te drukken, vervolgens de byte in hexadecimale code te verstrekken. Vervolgens kan men door "ENT NXT" weer in te drukken, bevestigen dat dit de juiste informatie is.

Verricht de boven omschreven handelingen zelf op de INSTRUCTOR 50!

Verricht de oefeningen 5, 6 en 7!

### Snel laden van het geheugen

#### Functie

Als men veel informatie in het geheugen moet laden op opeenvolgende plaatsen, dan zou men telkens de "ENT NXT" toets kunnen indrukken. Het laden van het geheugen kan echter sneller geschieden zonder dat deze toets telkens wordt ingedrukt. Men moet daarbij wel bedenken dat het dan niet mogelijk is, tijdens het intoetsen gemaakte fouten te corrigeren. Men moet dan de snelle laadmethode onderbreken en voor de correctie terugvallen op de werkwijze zoals hiervoor omschreven.

#### Procedure

Het snelle laden wordt ingeleid door de toets "REG" in te drukken, zie fig. 9 a). Het scherm vertoont nu r=. De INSTRUCTOR 50 verlangt nu een hexadecimaal cijfer. Voor het snelle laden moet men nu de F-toets van het hexadecimale veld indrukken. Dit is een hexadecimaal cijfer. De machine reageert hierop door op het scherm .Ad.= zichtbaar te maken, zie b). Dit betekent dat de INSTRUCTOR 50 om een adres vraagt. Stel dat we het laden willen beginnen op het hexadecimale adres H'0012'. Om dit adres in te voeren, slaat men eerst de 1 aan, zie c), vervolgens de

2, zie d), en daarna ENT NXT, zie e). De machine reageert op deze serie handelingen door op het scherm .0012 zichtbaar te maken. De INSTRUCTOR 50 duidt daarmee aan dat het snelle laden begint in geheugencel H'0012', en wegens de uiterst links geplaatste punt dat men deze gegevens nu direct kan laden. Men kan nu beginnen met b.v. op deze plaats H'A1' te laden, zie f) en g). Slaat men nu vervolgens een F en een E aan, zie h) en i), dan wordt bij het aanslaan van de F het volgende adres aangegeven, namelijk .0013 en de aangeslagen F. Nadat de E aangeslagen is, verschijnt op het scherm het beeld .0013 FE, d.w.z. dat de volgende geheugenplaats automatisch wordt gekozen. Men bespaart zich aldus het indrukken van de "ENT NXT"-toets. Men kan hiermee doorgaan tot men alle data aan het geheugen heeft toevertrouwd. Tot slot kan men, zie j), de toets "MEM" indrukken, waarop de INSTRUCTOR 50 reageert met .Ad.=. Van dit moment af kan men eventueel gemaakte fouten corrigeren volgens de methode die hiervoor is aangegeven. Men kan het snelle laden desgewenst beëindigen door elke andere besturingstoets in te drukken, zoals b.v. "MON" of "REG".

Verschijnt tijdens het aanslaan op het scherm het woord "ERROR 3", dan duidt dit erop dat een fout gemaakt werd in de functie van de INSTRUCTOR 50. Het kan zijn, dat in dat geval het geheugen de informatie niet of foutief heeft geregistreerd. Men dient dan de toets "MEM" in te drukken om het gewenste adres opnieuw te verschaffen en de data in te voeren. Ook kan men in plaats hiervan weer de toets "REG" indrukken, vervolgens toets F, het adres, correcte data, ENT NXT, enz.

P.S. Snel laden kan ook geschieden door de informatie afkomstig van een cassetteband over te dragen.

Verricht deze handelingen zelf op de INSTRUCTOR 50!

Verricht de oefeningen 8 en 9!

### Het zichtbaar maken en veranderen van de inhoud van het instructie-adresregister

#### Functie

Voor het elimineren van fouten is het van belang dat men de uitvoering van een programma kan onderbreken en dat men kan bepalen tot waar een programma op een bepaald moment is gekomen. Het instructie-adresregister wijst naar het adres van de *eerstvolgende* instructie die uitgevoerd zal worden. Door dit adres zichtbaar te maken, ziet men welk punt de uitvoering van het programma heeft bereikt.

#### Procedure

Men bepaalt dit door eerst op de toets "REG" te drukken; de INSTRUCTOR 50 toont dan op het scherm r=, zie fig. 10 a). De INSTRUCTOR 50 verwacht nu een hexadecimaal cijfer. Kiest men voor dit hexadecimale cijfer een C, dan is dit voor de INSTRUCTOR 50 een teken dat men de inhoud verlangt van het instructie-adresregister (Program

Counter, afgekort PC). Stel dat op dit moment het programma zo ver gevorderd is dat de inhoud van het instructie-adresregister H'0017' is. De INSTRUCTOR 50 reageert nu, zie b), door op het scherm .PC=0017 te tonen. Is dit voldoende, dan kan men eenvoudig op een van de andere commandotoetsen drukken. Als men echter de inhoud van de PC wil wijzigen, dan kan men nu de nieuwe inhoud presenteren. Stel men wenst de inhoud van het instructie-adresregister te wijzigen in H'0028'. Daar men de leidende nullen niet hoeft in te voeren, kan men volstaan met het aanslaan van 2 en 8, zie c) en d). In totaal kunnen vier hexadecimale digits in het instructie-adresregister worden geplaatst. Heeft men een fout gemaakt, dan kan men doorgaan met aanslaan tot de gewenste inhoud van het instructie-adresregister op het scherm verschijnt. Nu bevestigt men de juiste inhoud die definitief in het register moet worden geplaatst, door het indrukken van de toets "ENT NXT", zie e). Daar de INSTRUCTOR 50 aan het begin van deze serie handelingen eerst het indrukken van de toets "REG" heeft waargenomen, zal nu op het scherm r= verschijnen.

Verricht deze handelingen weer op de INSTRUCTOR 50!

Verricht de oefeningen 10 en 11!

### Stap-voor-stap functie

#### Functie

Nadat men een programma heeft geladen, dan is het voor het opsporen van fouten nuttig om het programma instructie-na-instructie te doorlopen en zodoende telkens te kunnen nagaan of het resultaat aan de verwachtingen voldoet.

#### Procedure

Heeft men met behulp van het invoegen van een bepaald startpunt in het instructie-adres register het begin van het programma aan de INSTRUCTOR 50 meegedeeld, dan kan men met behulp van de "STEP"-toets het programma instructie-na-instructie doorlopen, zie fig. 11. Wenst men dit te doen, dan dient het instructie-adresregister een bepaalde waarde te bezitten, van waar uit men wil beginnen. Men drukt daartoe eerst op de toets "REG", zie a), waarop de INSTRUCTOR 50 reageert door op het scherm r= te tonen.

Vervolgens slaat men C aan, waarop de machine reageert door het in het instructie-adresregister aanwezige adres te laten verschijnen. In figuur 11 b) vertoont het scherm .PC=0017, aangevend dat de stand van het instructie-adresregister H'0017' is. Nu kan men het nieuwe adres invoeren. Als men in positie 0 wil beginnen, dan dient men een 0 aan te slaan. Hierdoor wordt het instructie-adresregister op 0 gezet. Wil men op een ander punt beginnen, dan dient men dat adres eerst in het instructie-adresregister te laden. Stel dat dit adres moet luiden: H'0122'. De leidende 0 behoeft men niet aan te slaan. Als eerste wordt dus een 1 ingevoerd, zie c) en vervolgens twee 2'en, zie d) en e). Om te bewerkstelligen dat het instructie-adresregister nu deze waarde aanneemt, drukt men de toets "ENT NXT" in. De instructie

die thans zal worden uitgevoerd bij het bedienen van de toets "STEP" is gelegen op plaats H'0122' van het geheugen. Als men de "STEP"-toets indrukt g), dan verschijnt het adres van de eerstvolgende instructie die uitgevoerd zal worden. Nemen we aan dat de instructie op H'0122' 2 bytes groot is, dan zal de INSTRUCTOR 50 nu 0124 als adres geven. Tevens wordt de operatiecode op het betreffende adres getoond. In fig. 11 g) is dit aangegeven als 0124 CA. CA is de instructiecode voor STORE RELATIVE REGISTER 2. Drukt men de "STEP"-toets weer in, dan zal het scherm het adres van de volgende instructie weergeven met de bijbehorende instructiecode. Dit zal dus zijn 0126 CC. Zo kan men doorgaan met het doorlopen van het gehele programma. Men kan met deze STEP-functie ophouden door op de toets "RUN" te drukken, waarna het programma geheel zelfstandig verder wordt uitgevoerd. Na elke stap is het mogelijk om de volgende details te verifiëren:

1. alle registers
2. het instructie-adresregister
3. het Program Status Word UPPER
4. het Program Status Word LOWER
5. elke geheugenplaats

Op deze wijze kan men na elke stap, d.w.z. na elke instructie die wordt uitgevoerd, vrijwel elk onderdeel van de machine op zijn inhoud onderzoeken en nagaan of dit aan de opzet voldoet. Bovendien kan men voor elke "STEP" controleren of de ingebrachte waarde inderdaad overeenkomt met die welke men heeft willen invoeren. Als men in plaats van de "STEP"-toets de "RUN"-toets indrukt, dan heeft dit tot gevolg dat het volledige programma wordt uitgevoerd.

#### Opmerking

Raadpleegt men het geheugen, de registers of het instructie-adresregister, dan kan men ook de inhoud hiervan gelijktijdig veranderen, een en ander volgens de hiervoor omschreven procedures.

### Het aanbrengen van een BREEK PUNT in het programma

#### Functie

De zojuist beschreven STEP-methode is bijzonder waardevol als men inderdaad in staat is om het programma stap-voor-stap te doorlopen. Het kan echter voorkomen dat in het programma een lus aanwezig is, die verscheiden honderden malen achter elkaar wordt gebruikt. Bestaat deze lus bovendien uit een aantal instructies, dan moet men eerst enkele duizenden instructies uitvoeren voordat men uit de lus kan ontsnappen. Als men de lus een maal stap-voor-stap doorloopt, dan heeft men zijn functie reeds gecontroleerd en zou men het eigenlijk aan de INSTRUCTOR 50 willen overlaten om de overige lussen zelfstandig uit te voeren. Na afloop daarvan kan dan b.v. nog een instructie worden uitgevoerd. Vervolgens zou men dan de machine willen laten stoppen om daarna de rest van het programma weer stap-voor-stap te doorlopen.

#### Procedure

Deze faciliteit is aanwezig; hiertoe maakt men gebruik van het z.g. BREEK PUNT. Het breekpunt is een adres, dat men aan de INSTRUCTOR 50 moet verstrekken. Dit adres geeft aan waar de INSTRUCTOR 50 moet stoppen bij de uitvoering van een programma. Zodra dit adres door de INSTRUCTOR 50 bereikt wordt, zal de uitvoering van een programma ophouden en het adres met het bijbehorende operatiecode-deel op het scherm verschijnen. Zodra dit geschiedt, kan men dat waarnemen door het verschijnen van een - (streepje) vóór het adres, gevolgd door de instructiecode (-0154 C0) zoals in fig. 12 h) is aangegeven. Als men nu de "STEP"-toets indrukt, zal het programma weer stap-voor-stap uitgevoerd worden, terwijl het indrukken van de "RUN"-toets tot gevolg zal hebben dat het programma ononderbroken wordt uitgevoerd. Om een breekpunt te verwijderen, dient men de toets "BKPT" weer in te drukken, waarop de machine reageert met het tonen van .b.P=0154, te weten het adres dat zojuist werd ingevoegd, zie f). Drukt men thans de toets "BKPT" weer in, dan zal de machine hierop reageren door het tonen van .b.P=, echter zonder een adres weer te geven, zie g). Het breekpunt is nu verwijderd en men kan dus niet meer onderweg stoppen, tenzij men volgens de bovenstaande procedure een nieuw breekpunt invoegt.

Het breekpunt werkt uitsluitend indien men de "RUN"-toets heeft ingedrukt. Voert men een programma stap-voor-stap uit, dan wordt het breekpunt niet in acht genomen.

### Meldingen van fouten bij het gebruik van een breekpunt

Als men het breekpunt heeft gespecificeerd, dan kan de INSTRUCTOR 50 reageren door één van de onderstaande fouten te melden:

- ERROR 1. Dit betekent dat men een breekpunt wenst te specificeren in het ROM-gedeelte van het geheugen dat voor de INSTRUCTOR 50 zelf is gereserveerd. Om deze fout te elimineren kan men ermee volstaan, de toets BKPT weer in te drukken.
- ERROR 2. Deze melding betekent dat men een nieuw breekpunt tracht te introduceren, direct na het inbrengen van een breekpunt door de "ENT NXT"-toets in te drukken. Men kan deze fout herstellen door een willekeurige functietoets in te drukken.

#### R U N -toets

#### Functie

Het indrukken van deze toets beëindigt de MONITOR MODE en heeft tot gevolg dat de uitvoering van het programma begint. De eerste instructie die wordt uitgevoerd, is die welke gespecificeerd is door het instructie-adresregister.

## Procedure

De uitvoering blijft doorgaan totdat één van de onderstaande oorzaken het programma tot stoppen dwingt:

1. Een breekpunt is bereikt.
2. De "RST" of "MON"-toets wordt ingedrukt.
3. In het programma is een HALT- of een WRTC-instructie aanwezig. Deze WRTC-instructie wordt gebruikt door de MONITOR voor de bediening van de poorten en schakelaars op het frontpaneel.

## RESET-toets

### Functie

Als men de "RST" (RESET)-toets indrukt, wordt de activiteit van de INSTRUCTOR 50 onmiddellijk beëindigd en zal de processor opnieuw starten, uitgaande van de instructie op het adres H'000'. De initiële waarden van de registers zijn op dat moment onbekend.

P.S.: Indien een breekpunt is gespecificeerd, dan zal dit niet worden uitgevoerd.

### Procedure

De programma-uitvoering gaat voort tot één van de onderstaande toestanden zich voordoet:

1. De RST-toets wordt opnieuw gedrukt.
2. Een HALT-instructie wordt uitgevoerd. Bij de uitvoering van deze HALT-instructie stopt de processor tot de RST-toets opnieuw wordt ingedrukt, of - indien de INTERRUPT INHIBIT PSW-bit niet is gezet - tot er een INTERRUPT verschijnt.
3. Een WRTC-instructie wordt uitgevoerd of de "MON" toets wordt ingedrukt. In dat geval wordt de besturing overgedragen aan het MONITOR-programma; op het scherm verschijnt dan het bericht "HELLO".  
Als de controle wordt overgedragen aan de MONITOR, dan wordt het adres van de laatste handeling geregistreerd in het instructie-adresregister en worden de registerwaarden gered in het RAM-deel van de MONITOR. Ze kunnen nu met behulp van de desbetreffende stuurtoetsen onderzocht worden.
4. Indien logische waarde van de PAUZE-ingang hoog is, zal de processor eveneens stoppen. Deze ingang bevindt zich aan de achterzijde (stekeraansluiting). Als dit gebeurt, dooft het RUN-licht. De uitvoering van het programma zal weer beginnen als de logische waarde van de PAUZE-ingang laag wordt.

## Fout-meldingen

De MONITOR kan een aantal interne testen uitvoeren. Naast deze interne testen onderzoekt hij bovendien of bepaalde opdrachten wel toegestaan zijn. Treedt een fout op, hetzij in de machine, dan wel in de opdrachten, dan reageert de INSTRUCTOR 50 hierop door op het scherm de gedaante ERROR N te vermelden, waarbij N een cijfer kan

zijn van 1 t/m 9. De betekenis van deze meldingen is als volgt:

- ERROR 1. *Het breekpunt kan niet ingevoerd worden.* Dit is meestal het geval als men tracht een breekpunt aan te geven liggend in het gedeelte van het geheugen dat voor het MONITOR-programma is gereserveerd.
- ERROR 2. *Niet toegestane opdracht* Dit bericht betekent dat de volgorde van de commando's niet juist is en dat men deze moet verbeteren. Dit kan o.a. gebeuren door de commandocycclus te beëindigen, de knop MONITOR in te drukken, enz.
- ERROR 3. *Geheugeninhoud is niet op juiste wijze veranderd.* Dit bericht betekent dat er een poging is gedaan om data te schrijven in een geheugencel die b.v. niet toegankelijk is. Als men bij het invoeren van data dit bericht krijgt, dan kan het betekenen dat een fout is vastgesteld bij de controle of de data juist in het geheugen is geladen. Is dit het geval, dan moet men de toets "MEM" opnieuw indrukken en het laatste deel van de handelingen herhalen.
- ERROR 4. *Cassette-controle fout.* Als data op een cassette-recorder wordt geschreven, dan wordt aan deze data aan het einde een controleteken toegevoegd; aldus kan men later bij het teruglezen van deze data nagaan, of er een fout is opgetreden. Als nu bij het teruglezen van de cassette wordt vastgesteld dat de gelezen informatie niet juist is, dan verschijnt de melding ERROR 4.  
Men kan de cassette dan opnieuw laten inlezen. Indien het een tijdelijke fout was, b.v. stof, dan is het mogelijk dat de informatie nu foutloos ingelezen wordt. Is de band echter beschadigd, dan zal de informatie verloren zijn gegaan en is het desbetreffende deel van de band niet meer bruikbaar.
- ERROR 5. *Een geheugenschrijffout tijdens het inlezen van een cassette.* Bij het inlezen van een cassette en het inschrijven in het geheugen van de INSTRUCTOR 50, worden de geregistreerde bytes gecontroleerd. Indien een fout optreedt, dan zal de INSTRUCTOR 50 ERROR 5 vermelden. Men dient de band dan opnieuw in te lezen.
- ERROR 6. *De code van de magneetband wijkt af van de ASCII code.* De INSTRUCTOR 50 schrijft op de magnetische band in de z.g. ASCII code. Is een andere code op de band aanwezig, dan zal dit op een gegeven ogenblik blijken en gemeld worden. Voorts is het mogelijk dat de volumeregelaar van de cassette-recorder onjuist staat afgesteld. Men dient dit dan te verifiëren en de cassette, zo dit nodig is, opnieuw in te lezen.

- ERROR 7. *START*adres groter dan *STOP*adres. Als men data op een cassette schrijft en men geeft een startadres aan dat groter is dan het stopadres, dan zal dit door middel van deze indicatie worden gemeld.
- ERROR 8. *Er zijn twee toetsen in een kolom ingedrukt.* Dit betekent dat er twee toetsen tegelijk wer-

den ingedrukt. De MONITOR kan dan niet vaststellen welke actie gewenst is.

- ERROR 9. *Volgende instructie bevindt zich in het MONITOR-gebied.* Als de STEP-toets wordt ingedrukt en de volgende instructie zou in het MONITOR-gebied liggen, dan wordt dit gemeld door het verschijnen van de melding ERROR 9.

## HET GEBRUIK VAN DE INSTRUCTOR 50

De schrijver van programma's die de INSTRUCTOR 50 wil gebruiken, heeft de volledige 2650 microprocessor-instructieverzameling tot zijn beschikking. Wegens de samenwerking tussen de INSTRUCTOR 50 en de hardware en software van de gebruiker, moeten daarbij enkele beperkingen in acht worden genomen:

1. De MONITOR reserveert de instructie WRTC (Write Control). Tevens gebruikt hij de registerinstructies H'BO', H'B1', H'B2' en H'B3' om de plaats van een BREEKPUNT in een programma aan te geven. Als een dergelijke instructie wordt uitgevoerd in een programma van een gebruiker, dan zal de INSTRUCTOR 50 zelf terugkeren naar de MONITOR-mode en zal op het scherm het woord 'HELLO' verschijnen.
2. Indien een HALT-instructie (H'40') wordt uitgevoerd, dan zal de werking van de processor worden beëindigd. Dit wordt aangegeven door het doven van de LED "RUN". Er zijn twee wegen om de uitvoering opnieuw te starten. Dit kan geschieden door de RST-toets in te drukken, of – als interrupts niet zijn verboden – door het verrichten van het INTERRUPT. Dit kan b.v. worden bewerkstelligd door het indrukken van de toets INT.  
Als er een BREEKPUNT is gedefinieerd op de plaats van een HALT-instructie, dan zal de MONITOR ervoor zorgen dat deze HALT-instructie niet wordt uitgevoerd; de normale uitvoering van het programma zal dan doorgaan.
3. Van pagina 0 is, zoals bekend uit de instructiehandleiding, het onderste deel van deze pagina, evenals het bovenste deel, in wezen gereserveerd voor de ZBSR-instructies met positieve resp. negatieve displacements. Het MONITOR-programma gebruikt echter het bovenste deel van de pagina voor het springen naar MONITOR-programma's, zodat de gebruiker dit deel van de pagina niet moet gebruiken, tenzij hij de MONITOR wil binnengaan. Hetzelfde geldt voor INTERRUPT-vectors met negatieve displacements.

4. De MONITOR gebruikt voor zijn programma drie niveau's van de 2650-subroutine terugkeeradresstapel. Daar de terugkeeradresstapel is beperkt tot 8 niveau's, kan de gebruiker slechts over vijf niveau's van deze stapel beschikken voor het schrijven van zijn programma. Men dient zich te realiseren, dat tot deze 8 niveau's ook de INTERRUPTS behoren.

### Het gebruik van INTERRUPTS

INTERRUPTS voorzien in een synchronisatie van onderling onafhankelijke processen, die echter op een bepaald ogenblik gegevens moeten uitwisselen. Als een extern proces, d.w.z. een proces buiten de microprocessor 2650, een bepaald stadium heeft bereikt, dan kan dit proces de aandacht van deze microprocessor vragen. Deze zal hierop reageren door naar een bepaalde serviceroutine te springen. Deze serviceroutine kan hetzij specifiek voor het interrupterende externe proces zijn, waarbij dit proces dan hetzij zijn identiteit moet meedelen (vector interrupt), dan wel een routine kan zijn; in het laatste geval gaat de microprocessor op zoek naar het externe proces dat de INTERRUPT veroorzaakt (z.g. polling). De identiteit van het externe proces dat interrupteert, slaat niet zo zeer op het nummer van het betrokken proces, als wel op de fase waarin het proces zich bevindt. De INTERRUPT-vector geeft namelijk de plaats aan waar de vereiste gegevens voor de specifieke subroutine voor dit deel van het proces in het geheugen behorende bij de 2650 microprocessor aanwezig zijn. Het kan dus voorkomen dat één proces meer dan één INTERRUPT-vector kan genereren, die echter afhankelijk zijn van de voortgang van het betreffende proces.

De INSTRUCTOR 50 beschikt over drie methoden om de microprocessor 2650 te interrupteren.

De eerste methode is een INTERRUPT die de gebruiker van de INSTRUCTOR 50 teweeg kan brengen door de toets INT in te drukken.

De tweede methode maakt gebruik van een extern sig-

naal dat afgeleid wordt van de netfrequentie. In Nederland is deze 50 Hz. Hierdoor wordt elke 20 ms een INTERRUPT-verzoek ingediend. Met behulp van deze INTERRUPT zou men in de microprocessor 2650 een klokschakeling kunnen realiseren, in de Engelse literatuur een "real-time clock" genoemd. Wil men deze wijze van interrumperen toestaan, dan moet de INTERRUPT-schakelaar aan de onderzijde in de stand "A.C.LINE" geplaatst worden.

De derde methode van interrumperen is via de z.g. S100-bus interface. Deze S100-bus bevindt zich aan de achterzijde van de INSTRUCTOR 50 en kan m.b.v. de stekker toegankelijk worden gemaakt.

Het interruptmechanisme kan onder "software control" worden in- of uitgeschakeld. Dit kan in wezen geschieden op elk willekeurig punt in het programma door de SET of RESET van het INTERRUPT INHIBIT-bit (II van het PSW).

Als het INTERRUPT INHIBIT-bit in stand 1 staat, dan accepteert de microprocessor 2650 geen verzoeken om te gaan interrumperen. Staat het INTERRUPT INHIBIT-bit daarentegen in de 0-stand, d.w.z. dat de INTERRUPT-aanvragen worden gehonoreerd, dan zal de microprocessor op de interne INTERRUPT reageren. Het INTERRUPT INHIBIT-bit zal in de volgende gevallen in de 0-stand worden gezet (toelaten van INTERRUPT-aanvragen):

1. Bij het indrukken van de RST-toets, d.w.z. het terugzetten van de processor.
2. Door het uitvoeren van de instructie CPSU (Clear Program Status Upper), welke instructie voorzien dient te zijn van het juiste masker.
3. Bij de uitvoering van de instructie RETE (Return from Subroutine and Enable Interrupt).
4. Bij het uitvoeren van de instructie LPSU (Load Program Status Upper), waarbij de II-bit in de 0-stand geplaatst wordt.

Het INTERRUPT-mechanisme van de microprocessor 2650 kan met een vectored INTERRUPT werken. Accepteert de microprocessor een "interrupt request" (INTREQ), dan verstrekt de microprocessor automatisch een "interrupt acknowledge" (INTACK)-signaal. Het proces dat de microprocessor 2650 interrumpeert, antwoordt hierop door het plaatsen van de INTERRUPT-vector op de 2650 data-bus. Deze vector wordt gebruikt als een adres relatief t.o.v. H'0000' op pagina 0, of als een sprong naar een subroutine. De INTERRUPT-vector kan een direct of een indirect adres afgeven aan de data-bus. Bij directe adressering start die subroutine op het aangegeven adres, d.w.z. in de eerste 64 bytes van pagina 0, hetgeen bij een enkel proces geoorloofd kan zijn. Indien meer processen zullen interrumperen, dan kunnen de 64 bytes niet voor het schrijven van een subroutine gebruikt worden en dienen ze elders in het geheugen te worden ondergebracht. Voor dat doel maakt men gebruik van het indirecte adresseren.

De geadresseerde byte op pagina 0 (in het gebied 0 t/m

63) is nu de eerste byte van een tweetal waar het adres van de subroutine zich bevindt. Op grond hiervan wordt een sprong gemaakt naar de desbetreffende subroutine. Deze kan ergens binnen de totale adresseerbare capaciteit van 32 K liggen.

Als het toegestaan was (II=0) te interrumperen, dan beantwoordt de INSTRUCTOR 50 een verzoek tot interruptie door de volgende serie handelingen:

1. Hij voltooit de instructie die op dat moment in uitvoering is.
2. De processor plaatst de INTERRUPT INHIBIT in de logische 1 (II=1).
3. De eerste byte van de ZBSR-instructie wordt door de microprocessor 2650 intern opgewekt en in het instructie-register geplaatst.
4. De processor geeft een INTERRUPT ACKNOWLEDGE (INTACK) uit en wacht op de INTERRUPT-vector, die door het interrumperende proces moet worden afgegeven op de data-bus.
5. De INSTRUCTOR 50 genereert zelf een INTERRUPT-vector, te weten H'07' of H'87' op de data-bus. De 7 duidt aan dat byte 7 van het geheugen wordt gebruikt als eerste byte voor het uitvoeren van de subroutine voor de INTERRUPT. De 0 of de 8 duidt op een directe (H'07') of indirecte (H'87') adressering. Welke van de beide wordt gekozen, hangt af van de stand van de schakelaar DIRECT/INDIRECT INTERRUPT op het bedieningspaneel van de INSTRUCTOR 50. De gebruiker van de INSTRUCTOR 50 dient ervoor te zorgen dat, ingeval de directe adressering (H'07') wordt gebruikt, de subroutine op deze plaats start. Ingeval dat van indirecte adressering gebruik wordt gemaakt, dient op de geheugenplaatsen 7 en 8 het absolute adres aanwezig te zijn van het adres waarop de subroutine elders in het geheugen begint.
6. De 2650 voert vervolgens de ZBSR-instructie uit, hetgeen tot gevolg heeft dat van plaats 7 af het INTERRUPT-programma wordt uitgevoerd, resp. dat er een sprong wordt gemaakt naar de subroutine elders in het geheugen. Het adres van de volgende instructie van het oorspronkelijke programma is in de adresstapel dan gereed en de stapelwijzer wordt verhoogd.
7. Wanneer de INTERRUPT-routine in haar geheel is uitgevoerd, dan zal de laatste instructie zijn een RETE- of een RETC-instructie. Als gevolg hiervan verlaagt de microprocessor 2650 de stapelwijzer en vervangt hij de inhoud van het instructie-adresregister door de inhoud van de top van de stapel, zodat de uitvoering van het oorspronkelijke programma nu kan worden voortgezet.

Een externe INTERRUPT wordt gegenereerd door een onafhankelijk proces buiten de microprocessor 2650. Het is dus niet te voorspellen op welk moment deze INTERRUPT zal ontstaan. Een gevolg hiervan is dat een INTERRUPT kan plaats hebben bij elke willekeurige instructie van het

programma dat de microprocessor 2650 aan het uitvoeren is. Het programma kan evenwel een aantal registers in gebruik hebben. De INTERRUPT-subroutine kan echter ook behoefte hebben aan het gebruik van registers. Dit heeft tot gevolg dat de inhoud van deze registers op de een of andere wijze gered dient te worden. Dit kan worden gedaan op verschillende wijzen.

De eerste is ervoor zorg te dragen dat INTERRUPTS uitsluitend gedurende bepaalde instructies kunnen voorkomen. Zo kan men in het programma b.v. ervoor zorg dragen dat het bit II=1 is. Op een bepaald ogenblik kan men door middel van een instructie het bit II=0 maken en na de volgende instructie wederom II=1. Gedurende deze korte tijd kan een externe INTERRUPT het programma dan interrumperen. Voordat men echter deze interruptiemogelijkheid toestaat, kan men de inhoud van de registers in het geheugen redden. Na afloop van de subroutine dient het programma ervoor zorg te dragen dat de registers wederom geladen worden met de in het geheugen geredde informatie. De tweede methode is het redden van registers bij het betreden van de INTERRUPT-routine. Aan het einde van deze routine kan de geredde informatie dan wederom in de registers geplaatst worden. Een derde mogelijkheid is ervoor te zorgen dat in het hoofdprogramma uitsluitend gebruik gemaakt wordt van de registers in bank 0, terwijl bij de INTERRUPT-routine gebruik gemaakt wordt van de registers in bank 1. Een bijzondere voorziening blijft echter noodzakelijk voor Registers 0 en Program Status Word Lower. De inhoud van deze registers dient altijd nog gered te worden voor zover deze voor het hoofdprogramma belangrijke informatie bevat.

De INSTRUCTOR 50 INTERRUPT-faciliteit kan worden gekozen door de combinatie van de stand van de INTERRUPT-schakelaar en/of een verbinding aan de onderzijde van de INSTRUCTOR 50. Zoals reeds vermeld, doet de DIRECT/INDIRECT-schakelaar op het frontpaneel van de INSTRUCTOR 50 dienst om te bepalen of de INTERRUPT-vector directe, resp. indirecte adressering tot gevolg heeft. De geheugencel H'07' wordt zowel gebruikt voor de INTERRUPT die veroorzaakt wordt door het drukken van de toets INT, als wel door het INTERRUPT dat mogelijk wordt door het omschakelen van de schakelaar aan de onderzijde van de INSTRUCTOR 50, namelijk in de stand "A.C.LINE". Naast deze beide mogelijkheden van interrumperen kan de INSTRUCTOR 50, zoals reeds vermeld, ook geïnterrupteerd worden via de ingang van de S100-bus interface. Om dit te bewerkstelligen moet een verbinding aan de onderzijde van de INSTRUCTOR 50 worden gemaakt. Er wordt echter op gewezen dat dit uitsluitend zin heeft als externe apparatuur met de INSTRUCTOR 50 verbonden wordt. Bij de aflevering van de INSTRUCTOR 50 is de onderzijde zodanig bedraad dat de INTERRUPT hetzij geschiedt via het bedieningspaneel, dan wel door de 50 Hz voeding, een en ander afhankelijk van de stand van de schakelaar INTERRUPT SELECT SWITCH (zie fig. 13). In het

vervolg zullen de verbindingen die bij de aflevering aanwezig zijn, in de tekst worden aangegeven d.m.v. een \*. Het kiezen van een directe of een indirecte adressering blijft echter in alle gevallen bepaald door de stand DIRECT/INDIRECT van de INTERRUPT-schakelaar op het paneel.

### Doorverbindingen in veld A

verbinding	beschrijving
1 - 2 *	Normale werking. De microprocessor 2650 herkent INTERRUPT-verzoeken van de INTERRUPT-toets of van de 50 Hz INTERRUPT-bron, afhankelijk van de stand van de INTERRUPT SELECT SWITCH.
2 - 4	Bus interface. De 2650 herkent het interrumperen van de interface (pen 4 op de S100-bus stekker). Het herkennen van de INTERRUPT wordt gedaan gedurende de wisseling van laag naar hoog van het ingangssignaal.
2 - 39	Bus interface geïnverteerd. De situatie is identiek aan die geschetst onder 2 - 4, evenwel met de uitzondering dat de INTERRUPT wordt herkend bij de overgang van hoog naar laag van het ingangssignaal.
3 - 4	

### Het gebruik van de I/O schakelaars en LED's

De microprocessor 2650 kent een aantal methoden om de toestand en besturing van externe in- en uitvoerapparatuur vast te leggen. Eén van de methoden is de SERIEËLE INPUT-poort, gevormd door de SENSE INPUT-pen en de SERIEËLE OUTPUT-poort, die FLAG wordt genoemd. Voorts heeft de microprocessor 2650 voorzieningen voor 2 soorten parallele in- en uitvoerinstruaties, die men EXTENDED en NON-EXTENDED I/O noemt. De NON-EXTENDED I/O instructies zijn 1 byte groot en maken het mogelijk om data naar en uit twee 8 bit I/O-poorten te schrijven en te lezen, namelijk poort C en poort D. Deze beide poorten zijn via de data-bus en de controle-bus bereikbaar. De extended I/O-instructies zijn 2 bytes groot en kunnen de I/O capaciteiten van de microprocessor 2650 uitbreiden tot 256 bidirectionele I/O-poorten.

Behalve de genoemde I/O-instructies kan men ook een andere wijze van in- en uitvoer kiezen, de z.g. "MAPPED I/O". Deze methode wordt gerealiseerd door een geheugenadres aan te wijzen als een I/O-poort. In dat geval kan dus van de volledige geheugencapaciteit gebruik worden gemaakt. De I/O wordt echter wel benaderd vanuit het stand-

punt alsof het een geheugencel is, d.w.z. het wordt benaderd via de controlelijn, die onderscheid maakt tussen datastroom naar en van geheugen, resp. I/O. Bij "MAPPED I/O" wordt verondersteld dat deze I/O tot het geheugen behoort. Zie hiervoor verder de beschrijving van de microprocessor 2650 in een ander deel van deze handleidingen.

#### FLAG- en SENSE I/O-methodiek

De FLAG-output van de 2650 geeft in feite de toestand weer van een bit van het Program Status Word (PSW). Dit bit kan geschakeld worden met een PSW controleinstructie waaraan een masker is toegevoegd. Zodra de flip-flop in het PSW in de 1 staat, wordt de outputpen van de 2650 microprocessor hoog en als hij in de 0 staat, wordt deze pen laag. Op deze pen is een LED aangesloten, die op het bedieningspaneel zichtbaar is. Als deze LED licht geeft, dan staat de flip-flop in het PSW in de logische 1, als deze LED gedoofd is, dan bevindt zich de flip-flop in de logische 0. Seriële input wordt verkregen door middel van de SENSE-pen van de microprocessor. Deze pen is direct verbonden met een toets op het bedieningspaneel van de INSTRUCTOR 50, de SENSE-toets. Normaal is de spanning op de SENSE-pen laag. Zodra echter de SENSE-toets ingedrukt wordt, dan zal de spanning op een hoog niveau komen, d.w.z. een logische 1 worden.

#### NON-EXTENDED I/O-methodiek

De microprocessor 2650 kan twee externe bidirectionele I/O-poorten besturen m.b.v. vier verschillende 1 byte-instructies. De instructies zijn: WRTC, WRDT, REDC en REDD. Deze instructies bewerkstelligen de overdracht van één byte naar of van de poorten. De plaats van oorsprong, resp. van bestemming, is één van de zeven aanwezige registers. Het nummer van het register komt tot uitdrukking in de operatiecode, terwijl de bank van het betreffende register, zoals bekend, vooraf d.m.v. een instructie voor het PSL gedefinieerd dient te zijn. Het register R0 behoeft niet d.m.v. een dergelijke instructie gedefinieerd te worden, daar R0 gemeenschappelijk is voor de beide registerbanken. Eén van deze NON-EXTENDED poorten, en wel poort D, bevindt zich op het bedieningspaneel van de INSTRUCTOR 50. Zet men de poortadresschakelaar in de stand NON-EXTENDED, dan wordt deze poort automatisch gekozen. Uiteraard dient dan wel de instructie WRDT of REDD in het programma voor te komen. Door deze faciliteit is het mogelijk om met behulp van de acht schakelaars data aan de 2650 over te dragen. Op dezelfde wijze is het mogelijk data van de 2650 te betrekken d.m.v. de acht LED's die boven de schakelaars zijn geplaatst. Het minst significante bit van de overgedragen byte bevindt zich uiterst rechts, het meest significante bit uiterst links.

#### EXTENDED I/O-methodiek

De microprocessor 2650 kan maximaal 256 bidirectionele I/O-poorten bereiken m.b.v. de instructies WRTE (Write Extended) en REDE (Read Extended). Het tweede byte van deze instructie specificeert de desbetreffende I/O-poort. Deze I/O-poort wordt gekozen via de adresbus en d.m.v. de controledraad  $M/\bar{I}O$ . De extended I/O kan in de INSTRUCTOR 50 nagebootst worden m.b.v. dezelfde acht schakelaars en acht LED's, zoals bij de non-extended I/O. Men kan dit realiseren door de I/O-schakelaar in de positie EXTENDED I/O te zetten. Het adres van deze I/O-poort is in dit geval H'07'. Dit adres dient als tweede byte in de REDE- en WRTE-instructies aanwezig te zijn.

#### MEMORY MAPPED I/O-methodiek

Als men in de INSTRUCTOR 50 de "MEMORY MAPPED I/O" wenst na te bootsen, dan dient men de I/O-schakelaar in de bovenste stand te zetten. Nu wordt aan de I/O-schakelaars en indicators het adres H'OFFF' toegevoegd. Hierdoor is het mogelijk data met de normale STORE- of LOAD-instructie naar, resp. van de I/O-schakelaars en LED's te voeren. De LOAD- en STORE-instructies bevatten een tweetal bits, die het register aangeven van waar, resp. waarheen data moet worden gevoerd. Deze data wordt betrokken, resp. overgedragen van de acht schakelaars, resp. acht LED's op het bedieningspaneel.

#### Keuze klokfrequentie – S100 bus

De bus-interface aan de onderzijde van de INSTRUCTOR 50 heeft een drietal aansluitingen voor het geven van bepaalde kloksignalen naar de omliggende logica. (Zie figuur 13 Veld B). De standaardfrequentie van de INSTRUCTOR 50 is 895 kHz. Tevens kan een tweede kloksignaal aan de onderzijde worden afgenomen ter grootte van ongeveer 303 kHz. Dit signaal is afgeleid van de pen OPREQ van de microprocessor 2650. Zoals blijkt uit de handleiding hiervan, behoeft dit signaal (de Operation Request) niet bij elke instructie aanwezig te zijn. De frequentie kan dus geringer zijn dan 303 kHz. Aan de onderzijde is op de pennummers 11 en 12 het systeemkloksignaal van 895 kHz van de INSTRUCTOR 50 aanwezig. Op de pennen 13 en 14 is het signaal OPREQ aanwezig.

De uitgangen van de S100-bus zijn verbonden met de punten 8, 9 en 10. Pen 8 is verbonden met uitgang 25, bekend als bussignaal 01, pen 9 is verbonden met de uitgang 24, bekend als bussignaal 02 en eveneens met uitgang 49, bekend als bussignaal "klok". Bij aflevering van de INSTRUCTOR 50 is pen 9 verbonden met pen 12, zodat op uitgang 24 het 895 kHz signaal staat.



## Keuze van de spanningsbron

De INSTRUCTOR 50 werkt normaal met zijn eigen interne spanningsstabilisatie, een en ander via de transformator die op het lichtnet kan worden aangesloten. Het is echter ook mogelijk om de INSTRUCTOR 50 te laten werken via de interface aansluiting van een geregelde 5 V spanningsvoorziening. Voor dat doel zijn de punten 18, 19 en 20 aanwezig

(Zie figuur 13 Veld C). Zijn de punten 18 – 20 doorverbonden, zoals het geval is bij de aflevering van de INSTRUCTOR 50, dan werkt de INSTRUCTOR met de eigen interne spanningsvoorziening. Worden de punten 18 en 19 doorverbonden, dan moet de externe spanningsvoorziening op de achterzijde van de INSTRUCTOR 50 worden aangelegd.

---

## INGEBOUWDE SUBROUTINES

De INSTRUCTOR 50 heeft in zijn geheugen een zestal subroutines die bijzonder nuttig zijn voor de communicatie met deze INSTRUCTOR en voor speciale bewerkingen die in een programma nodig kunnen zijn. De subroutines zijn: MOVE, DISPLAY, USER DISPLAY, NIBBLE, INPUT DATA en MODIFY DATA.

De subroutines worden opgeroepen door een ZBSR (Zero Branch to Subroutine Relative) instructie. De instructie dient voorzien te zijn van een extra byte. Deze byte is bij de hierna te behandelen instructie aangegeven in de vorm van twee hexadecimale cijfers. De byte begint altijd met een 1, waardoor dus een indirecte adressering wordt verkregen. De indirecte adressen zijn alle in de laatste 64 bytes van pagina 0 (8K bytes groot) gelokaliseerd. Wegens de indirecte adressering doet het voor de gebruiker niet ter zake waar de subroutines in feite in het geheugen aanwezig zijn. De subroutines maken veelal gebruik van een aantal registers; soms wordt ook vereist, dat deze registers,

voordat de subroutine wordt aangeroepen, gevuld zijn met bepaalde informatie. Daar bij de aanroep van een subroutine op de adresstapel een adres wordt geplaatst, dient men van te voren zorgvuldig te overwegen of het nog mogelijk is, deze subroutine aan te roepen. Als er niet voldoende plaats meer op de stapel aanwezig is om het vertrekadres te redden, mag de subroutine niet worden aangeroepen. Het aantal niveau's dat de subroutine zelf oproept, is per subroutine aangegeven. Hiermee is dus niet bedoeld het niveau dat de subroutine zelf nodig heeft om aangeroepen te worden, doch uitsluitend het aantal malen dat de subroutine zelf weer gebruik maakt van andere subroutines. Bij de aanroep van een subroutine dient dus ten minste voor één adres plaats te zijn op de stapel, en bovendien dient men nog zo veel plaats in reserve te hebben als het aantal aanroepen die bij de subroutine zelf aangegeven zijn. De maximale stapeling is acht; overschrijdt men dit getal, dan gaat het vertrekadres verloren.

## MOVE-subroutine

### Functie

MOVE verplaatst een bericht ter grootte van acht bytes van een gebruikersprogramma naar dat deel van het geheugen dat het scherm van informatie voorziet. Door deze subroutine te combineren met de subroutine DISPLAY, kunnen berichten op het scherm zichtbaar worden gemaakt.

### Werking

Voordat deze subroutine wordt aangeroepen, dient een bericht van acht bytes in het programma aanwezig te zijn. De plaats waar dit bericht begint, dient in de registers R1 en R2 vóór het oproepen van de subroutine te worden geregistreerd. Het adres bestaat uit twee bytes, waarvan de eerste drie bits zeker bestaan uit nullen (32K geheugen). Het meest significante byte dient in register R1 geregistreerd te worden, het minst significante in register R2.

### Aanroep-instructie:

<i>mnemonic</i>	<i>hexadecimale waarde</i>
ZBSR *MOV	BB, FE

*Gebruikte registers:* R1 = meest significante byte van het adres -1  
R2 = minst significante byte van het adres -1

### Vereiste aantal stapelplaatsen na de oproep: 0

Wegens de beperkte omvang van het scherm kunnen niet alle 256 mogelijkheden van het bericht van een byte worden gebruikt; er is slechts een beperkte weergave mogelijk. In onderstaande tabel zijn de hexadecimale waarden met de erbij behorende tekens op het scherm weergegeven.

<i>karakter</i>	<i>waarde</i>	<i>karakter</i>	<i>waarde</i>
*0 of O	H'00'	*P	H'10'
*1 of I	H'01'	*L	H'11'
*2	H'02'	*U	H'12'
*3	H'03'	*R	H'13'
*4	H'04'	*H	H'14'
*5 of S	H'05'	*O	H'15'
*6 of G	H'06'	*=	H'16'
*7	H'07'	*spatie	H'17'
*8	H'08'	*J	H'18'
*9	H'09'	* -	H'19'
*A	H'0A'	*	H'1A'
*B	H'0B'	*Y	H'1B'
*C	H'0C'	*N	H'1C'
*D	H'0D'		
*E	H'0E'		
*F	H'0F'		

### Voorbeeld van een MOVE-subroutine aanroep

<i>adres</i>	<i>data</i>	<i>instructie mnemonic</i>	<i>commentaar</i>
0010	05,00	LODI,R1 H'00'	Laad adreswijzer
0012	06,FF	LODI,R2 H'FF'	-1 in R1 en R2
0014	BB,FE	ZBSR *MOV	(H'0100'-1 = H'00FF') Spring naar MOVE' Het bericht op 0100-0107 wordt verplaatst naar de buffer van het scherm
0100	17		= spatie
0101	14		= H
0102	0E		= E
0103	11		= L
0104	11		= L
0105	00		= O
0106	17		= spatie
0107	17		= spatie

## DISPLAY-subroutine

### Functie

Als deze subroutine wordt gebruikt in samenhang met de MOVE-subroutine, is het mogelijk op het scherm van de INSTRUCTOR 50 acht karakters weer te geven. DISPLAY leest het bericht dat is geregistreerd in de buffer van het scherm, b.v. door middel van de subroutine MOVE. Vervolgens plaatst deze subroutine dit bericht in dat deel van de INSTRUCTOR 50 dat de gegevens voor het scherm verzorgt. Deze routine verzorgt ook de weergave die plaats heeft na het indrukken van de besturingstoets of een hexadecimale toets.

### Werking

De subroutine DISPLAY heeft drie operatiemodes. Deze worden geselecteerd door vóór het aanroepen van de subroutine in register R0 een opdracht-byte te schrijven. De waarde van deze byte en de erbij behorende functie zijn hieronder weergegeven.

### waarde functie

H'00' Geef het bericht door aan het scherm tot een data-toets of een funktietoets wordt ingedrukt. Plaats de waarde van de ingedrukte toets in register R0.

H'01' Voer de DISPLAY-subroutine één maal uit. Bepaal niet de waarde van een eventueel ingedrukte toets. Daar het eenmalig passeren van deze subroutine niet in zichtbare tekens op het scherm resulteert, moet de subroutine deel uitmaken van een lus waarin de DISPLAY subroutine een voldoende aantal malen wordt aangeroepen om het bericht zichtbaar te maken.

H'80' Deze opdracht is identiek aan die van H'00', met die uitzondering dat nu uiterst links op het scherm een punt verschijnt.

*Aanroep-instructie:*

*mnemonic*                      *hexadecimale waarde*  
ZBSR \*DISPLY                  BB, EC

*Vereiste aantal stapelplaatsen na de oproep:* 0

*Gebruikte registers:* R0, R1, R2 en R3.

R0= DISPLAY-opdracht. Na het uitvoeren van de subroutine is R0= toetswaarde. Hieronder zijn de toetsen en de bijbehorende waarde in het register R0 na afloop van de subroutine weergegeven.

<i>toets</i>	<i>hexadecimale waarde</i>	<i>toets</i>	<i>hex. waarde</i>
0	H'00'	C	H'0C'
1	H'01'	D	H'0D'
2	H'02'	E	H'0E'
3	H'03'	F	H'0F'
4	H'04'	WCAS	H'80'
5	H'05'	BKPT	H'81'
6	H'06'	RCAS	H'82'
7	H'07'	REG	H'83'
8	H'08'	STEP	H'84'
9	H'09'	MEM	H'85'
A	H'0A'	RUN	H'86'
B	H'0B'	ENT/NXT	H'87'

0100	17
0101	14
0102	0E
0103	11
0104	11
0105	00
0106	17
0107	17

Eerste byte van het bericht is een spatie  
= H  
= E  
= L  
= L  
= O  
= spatie  
Laatste byte van het bericht is een spatie.

**USER DISPLAY-subroutine**

**Functie**

De USER DISPLAY-subroutine combineert de functies van de twee besproken subroutines MOVE en DISPLAY. Dit betekent dat USER DISPLAY een bericht van acht bytes verplaatst uit het programma van de gebruiker naar het gedeelte voor de buffer van het scherm en vervolgens dit bericht op het scherm zichtbaar maakt.

**Werking**

Voordat de USER DISPLAY wordt opgeroepen, dient het adres van het eerste byte van het bericht -1 gegeven te worden in de registers R1 en R2, zoals dit ook gebeurt bij de subroutine MOVE. Bovendien moet in register R3 het verlangde commando worden geladen, zoals dat gebeurt in de DISPLAY-subroutine in register R0. Na afloop van de subroutine treft men in R0 wederom de waarde van de ingedrukte toets aan, zoals dat ook gebeurt bij de DISPLAY-subroutine.

*Aanroep-instructie:*

*mnemonic*                      *hexadecimale waarde*  
ZBSR \*USRDSP                  BB,E6

*Gebruikte registers:* R0, R1, R2, R3.

- R3= DISPLAY-commando.
- R1= meest significante byte van het beginadres -1.
- R2= minst significante byte van het beginadres -1.
- R0= toetswaarde na afloop van de subroutine.

*Vereiste aantal plaatsen op de adresstapel:* 2

Hieronder volgt een voorbeeld voor het gebruik van de USER DISPLAY-subroutine.

*Voorbeeld van een MOVE en DISPLAY subroutine aanroep*

<i>adres</i>	<i>data</i>	<i>instructie mnemonic</i>	<i>commentaar</i>
0010	05,00	LODI,R1 H'00'	Plaats berichtwijzer -1 in R1
0012	06,FF	LODI,R2 H'FF'	en R2 ('0100'-1 = '00FF')
0014	BB,FE	ZBSR *MOV	Roep de MOVE-subroutine aan.
0016	04,00	LODI,R0 H'00'	Plaats de opdracht in R0.
0018	BB,EC	ZBSR *DISPLY	Roep de DISPLAY-subroutine aan
001A	E4,86	COMI,R0 H'86'	Vergelijk teruggegeven toetscode met RUN-code.
001C	1C,XX,XX	BCTA,EQ H'XXXX'	Indien gelijk, spring naar dit adres: H'XXXX'
001F	1F,00,16	BCTA,UN H'0016'	Indien ongelijk, ga naar H'0016' en wacht op volgende toets

*Voorbeeld van een USER DISPLAY subroutine aanroep*

adres	data	instructie mnemonic	commentaar
0010	05,00	LODI,R1 H'00'	Plaats berichten-tabelwijzer -1 en R1 en R2 ('0100' -1 = '00FF')
0012	06,FF	LODI,R2 H'FF'	
0014	07,00	LODI,R3 H'00'	Plaats opdracht in R3
0016	BB,E6	ZBSR *USRDSP	Roep USER DISPLAY aan
0018	E4,86	COMI,R0 H'86'	Vergelijk teruggegeven toetswaarde met RUN-waarde
001A	1C,XX,XX	BCTA,EQ H'XXXX'	Indien gelijk, spring naar H'XXXX'
001D	1F,00,10	BCTA,UN H'0010'	Indien ongelijk, ga terug naar H'0010' en druk andere toets
0100	17		Eerste byte van bericht is een spatie = H
0101	14		= E
0102	0E		= L
0103	11		= L
0104	11		= O
0105	00		= spatie
0106	17		Laatste byte is een spatie
0107	17		

**NIBBLE-subroutine**

**Functie**

Zoals reeds vermeld, kunnen niet alle 256 mogelijkheden, die een byte biedt, zichtbaar worden gemaakt op het scherm. Kunnen echter de 2x4 bits van een byte in afzonderlijke bytes worden geplaatst, zodat één byte in wezen twee bytes voortbrengt (elke byte met 4 leidende nullen), dan zou het mogelijk zijn deze byte met behulp van twee opeenvolgende plaatsen op het scherm zichtbaar te maken. De subroutine NIBBLE snijdt de byte daartoe in twee delen en plaatst deze in afzonderlijke bytes. Daar elke byte twee andere bytes produceert, kan men op deze wijze op het scherm maximaal vier bytes zichtbaar maken.

**Werking**

*Aanroep-instructie:*

mnemonic	hexadecimale waarde
ZBSR *DISLSD	BB,F4

*Gebruikte registers:* R0 en R1.

- R0= byte, die gesplitst dient te worden
- R0 na de subroutine = meest significante nibble
- R1 na de subroutine = minst significante nibble

*Vereiste aantal plaatsen op de stapel:* 1

In het onderstaande programma is het gebruik van de NIBBLE-subroutine weergegeven. Het is duidelijk dat na afloop van de subroutine de inhoud van de registers R1 en R0 naar het geheugen moet worden overgebracht om te voorkomen dat deze inhoud verloren gaat.

*Voorbeeld van NIBBLE*

Bij betreden: R0 = F3  
 Bij vertrek: R0 = 0F  
 R1 = 03

*Voorbeeld van het gebruik van het NIBBLE-programma*

adres	data	instructie mnemonic	commentaar
0000	70	REDD,R0	Lees I/O poort (non-extended in R0)
0001	BB,F4	ZBSR *DISLSD	Roep NIBBLE subroutine op
0003	CD,01,07	STRA,R1 H'0107'	Store low-order nibble in de berichtentabel
0006	CC,01,06	STRA,R0 H'0106'	Store high-order nibble in de berichtentabel
0009	05,00	LODI,R1 H'00'	Plaats berichtenwijzer -1
000B	06,FF	LODI,R2 H'FF'	in R1 en R2
000D	04,00	LODI, R0 H'00'	Plaats opdracht in R0
000F	BB,E6	ZBSR *USRDSP	Roep USER DISPLAY-subroutine op
			Laat vorige poort D-waarde zien. Nieuwe I/O-waarde wordt op schakelaars gegeven en verschijnt op het scherm indien een van de toetsen wordt ingedrukt
0011	1B,6D	BCTR,UN H'6D'	Keer terug naar 0000 en verkrijg I/O waarde = "R" (1e byte van bericht)
0100	13		= "E"
0101	0E		= "D"
0102	0D		= "D"
0103	0D		= "="
0104	16		= "spatie"
0105	17		= "spatie" (H-order nibble wordt hier opgeslagen.
0106	17		= "spatie" (L-order nibble wordt hier opgeslagen
0107	17		

H'80' Deze opdracht is identiek aan die van H'00', met die uitzondering dat nu uiterst links op het scherm een punt verschijnt.

*Aanroep-instructie:*

*mnemonic*                      *hexadecimale waarde*  
ZBSR \*DISPLY                  BB, EC

*Vereiste aantal stapelplaatsen na de oproep:* 0

*Gebruikte registers:* R0, R1, R2 en R3.

R0= DISPLAY-opdracht. Na het uitvoeren van de subroutine is R0= toetswaarde. Hieronder zijn de toetsen en de bijbehorende waarde in het register R0 na afloop van de subroutine weergegeven.

<i>toets</i>	<i>hexadecimale waarde</i>	<i>toets</i>	<i>hex. waarde</i>
0	H'00'	C	H'0C'
1	H'01'	D	H'0D'
2	H'02'	E	H'0E'
3	H'03'	F	H'0F'
4	H'04'	WCAS	H'80'
5	H'05'	BKPT	H'81'
6	H'06'	RCAS	H'82'
7	H'07'	REG	H'83'
8	H'08'	STEP	H'84'
9	H'09'	MEM	H'85'
A	H'0A'	RUN	H'86'
B	H'0B'	ENT/NXT	H'87'

*Voorbeeld van een MOVE en DISPLAY subroutine aanroep*

<i>adres</i>	<i>data</i>	<i>instructie mnemonic</i>	<i>commentaar</i>
0010	05,00	LODI,R1 H'00'	Plaats berichtwijzer -1 in R1
0012	06,FF	LODI,R2 H'FF'	en R2 ('0100' -1 = '00FF')
0014	BB,FE	ZBSR *MOV	Roep de MOVE-subroutine aan.
0016	04,00	LODI,R0 H'00'	Plaats de opdracht in R0.
0018	BB,EC	ZBSR *DISPLY	Roep de DISPLAY-subroutine aan
001A	E4,86	COMI,R0 H'86'	Vergelijk teruggegeven toetscode met RUN-code.
001C	1C,XX,XX	BCTA,EQ H'XXXX'	Indien gelijk, spring naar dit adres: H'XXXX'
001F	1F,00,16	BCTA,UN H'0016'	Indien ongelijk, ga naar H'0016' en wacht op volgende toets

0100	17
0101	14
0102	0E
0103	11
0104	11
0105	00
0106	17
0107	17

Eerste byte van het bericht is een spatie  
= H  
= E  
= L  
= L  
= O  
= spatie  
Laatste byte van het bericht is een spatie.

### USER DISPLAY-subroutine

#### Functie

De USER DISPLAY-subroutine combineert de functies van de twee besproken subroutines MOVE en DISPLAY. Dit betekent dat USER DISPLAY een bericht van acht bytes verplaatst uit het programma van de gebruiker naar het gedeelte voor de buffer van het scherm en vervolgens dit bericht op het scherm zichtbaar maakt.

#### Werking

Voordat de USER DISPLAY wordt opgeroepen, dient het adres van het eerste byte van het bericht -1 gegeven te worden in de registers R1 en R2, zoals dit ook gebeurt bij de subroutine MOVE. Bovendien moet in register R3 het verlangde commando worden geladen, zoals dat gebeurt in de DISPLAY-subroutine in register R0. Na afloop van de subroutine treft men in R0 wederom de waarde van de ingedrukte toets aan, zoals dat ook gebeurt bij de DISPLAY-subroutine.

*Aanroep-instructie:*

*mnemonic*                      *hexadecimale waarde*  
ZBSR \*USRDSP                  BB,E6

*Gebruikte registers:* R0, R1, R2, R3.

R3= DISPLAY-commando.

R1= meest significante byte van het beginadres -1.

R2= minst significante byte van het beginadres -1.

R0= toetswaarde na afloop van de subroutine.

*Vereiste aantal plaatsen op de adresstapel:* 2

Hieronder volgt een voorbeeld voor het gebruik van de USER DISPLAY-subroutine.

003A	04,01	LODI,R0 H'01	Plaats opdracht in R0 (H'01' = 2 digits)	0107	01	= "1" (voorgaande ingevoerde data waarden)
003C	BB,FC	ZBSR *GNPA	Roep MODIFY DATA-subroutine aan.			
0100	17		1e byte van het bericht = "spatie"			
0101	18		= "J"			
0102	15		= "O"			
0103	0B		= "B"			
0104	16		= "E"			
0105	17		= "spatie"			
0106	00		= "0" (voorgaande ingevoerde data waarden)			

Registerinhoud bij een terugkeer uit de MODIFY DATA subroutine

register	inhoud	commentaar
R0	H'02'	Ingevoerde data is geplaatst in R0
R1	H'XX'	Data in R1 is betekenisloos
R2	H'86'	Waarde van de RUN-toets
R3	H'00'	Geeft aan dat de waarde in R0 geldig is

## SYSTEEM-EXPANSIE

De INSTRUCTOR 50 biedt aan de achterzijde, zoals vermeld, de mogelijkheid om een contrasteker aan te brengen. Hierdoor kan de S100-interface worden benut. Dit schept de mogelijkheid om het systeem uit te breiden met extra geheugen en bovendien met een aantal in- en uitvoer-apparaten. Het aantal mogelijkheden is zo groot dat dit nauwelijks te beschrijven is. Behalve in de S100-bus zijn nog een aantal pennen nieuw benoemd. Deze reservepennen kunnen gebruikt worden om specifieke 2650 signalen, zoals b.v. OPREQ, R/W en M/IO uit te voeren. Hierdoor wordt aan de gebruiker de mogelijkheid geboden, alle speciale faciliteiten van de 2650 te gebruiken. De interface-signalen zijn onderstaand beschreven.

pen	mnemonic	signaalbeschrijving
1	+8	+8 V niet geregeld. Deze ingang kan een +8 V voorziening geven voor de INSTRUCTOR 50, afhankelijk van de positie van de verbinding voor de stroomvoorziening aan de onderzijde.
2	+16	Positieve 16 V spanningsvoorziening. Deze ingang wordt gereserveerd voor +16 V, eventueel nodig voor een S100 periferie printed circuit. De +16 V is niet noodzakelijk voor de INSTRUCTOR 50 en wordt ook niet door deze opgewekt.

3	XRDY	External Ready. XRDY is een signaal dat een I/O-apparaat geeft, als de data-overdracht met de 2650 is geschied. De INSTRUCTOR beschouwt dit als een OPACK voor de 2650.
4	VIO	Vectored Interrupt 0, VIO, is een externe INTERRUPT-lijn. Is hiervoor de verbinding aangebracht aan de onderzijde van de Instructor 50 voor externe INTERRUPTS, dan wordt de VIO geregistreerd; deze veroorzaakt een directe of indirecte interrupt, afhankelijk van de stand van de schakelaar DIRECT/INDIRECT.
5 t/m 11		Niet gebruikt!
12	R/W*	Read/Write. Dit is een besturingssignaal van de microprocessor 2650 dat aangeeft of deze een lees- dan wel een schrijfoperatie met de externe apparatuur wenst uit te voeren. Dit signaal is uitsluitend geldig indien OPREQ "hoog" is.
13	WRP*	Write Pulse. De microprocessor 2650 wekt een signaal op gedurende een geheugen- of een I/O-schrijfoperatie.

\* Dit is geen standaard S100-interface-signaal

	WRP kan worden gebruikt om data te laten overnemen door de externe apparatuur.	43	DI7	Data IN bit 7.
14	M/IO*			Niet gebruikt!
	Memory/Input-Output. Dit is een signaal van de microprocessor 2650 dat aangeeft, of de adresbus een geheugen of een I/O-adres bevat gedurende de dataoverdracht.	44		
15	RESET*		45	SOUT
	Reset. Als het signaal op deze pen hoog is wordt er een RESET-operatie uitgevoerd gelijk aan het indrukken van de RST-toets op het paneel van de INSTRUCTOR 50. Dit heeft tot gevolg dat de uitvoering van het programma op de plaats H'0000' begint.			SOUT geeft aan dat de adresbus een adres bevat voor een uitgangsaparaat. Het desbetreffende apparaat kan de waarde van de databus accepteren indien PWR (pen 77) actief is.
16	RUN/WAIT*		46	SINP
	Dit signaal van de microprocessor 2650 geeft aan of deze wacht dan wel een programma uitvoert.			SINP geeft aan dat de adresbus een adres bevat voor een ingangsaparaat. Het geselecteerde apparaat kan zijn data overdragen indien PDBIN (pen 78) actief is.
17	PAUSE*		47	SMEMR
	Dit is een 2650 controle-ingangssignaal, dat gebruikt kan worden voor een Direct Memory Access (DMA). Als het signaal "hoog" wordt gemaakt, stopt de microprocessor 2650 met de uitvoering van instructies, direct na het uitvoeren van de instructie waarmee hij bezig is.			Memory Read. Dit signaal geeft aan dat de adresbus het adres van een geheugenlocatie bevat en dat de microprocessor 2650 een geheugenleesoperatie uitvoert.
18 t/m 23			48	
24 $\Phi$ 1	Niet gebruikt!		49	CLOCK
	Fase 1-klok, kan worden bestuurd door het 895 kHz kloksignaal of het OPREQ-signaal, afhankelijk van de configuratie aan de onderzijde van de Instructor 50.			Systeemklok. Afhankelijk van de bedrading aan de onderkant van de Instructor 50, wordt deze pen aangedreven door de 895 kHz systeemklok dan wel door de 2650 OPREQ-output.
25 $\Phi$ 2			50	GND
	Fase 2-klok, kan worden bestuurd door de systeemklok of door het OPREQ-signaal, afhankelijk van de configuratie, aangebracht aan de onderzijde van de Instructor 50.		51	+8 Volt
26 t/m 28				Systeemaarde
29	A5		52	-16 Volt
30	A4			Een positieve 8 V-lijn, via welke +8 V aan de INSTRUCTOR 50 kan worden verstrekt als aan de onderzijde daartoe voorzieningen zijn getroffen.
31	A3		53	
32	A15		54	
	Adres-bit 15. Aangezien de microprocessor 2650 slechts 32K kan adresseren, is deze verbonden met de aarde.		55	D0*
33	A12		56	D1*
34	A9		57	D2*
35	DO1		58	D3*
36	DO0		59	D4*
37	A10		60	
38	DO4		61	D5*
39	DO5		62	D6*
40	DO6		63	D7*
41	DI2		64	UOPREQ*
42	DI3			UOPREQ (User Operation Request) is een besturingssignaal voor de microprocessor 2650, dat aangeeft dat de adres- en de data-bus en andere controllijnen geldig zijn. UOPREQ kan worden gebruikt om de data-bus te kopiëren voor de schrijfoperaties en activeert ingangsbusschrijvers voor leesoperaties.

65	INTACK*	Interrupt Acknowledge. De microprocessor 2650 geeft een bevestiging in de vorm van INTACK als een interrupt is herkend en verwerkt. Na dit signaal kan een interrumperend apparaat via de data-bus een relatief adres sturen, waardoor m.b.v. de ZBSR-instructie een sprong kan worden gemaakt in de eerste 64 bytes van pagina 0.	74 t/m 76 77	PWR	de 0-stand staat. PINT kan worden gebruikt om de microprocessor 2650 uit de HALT-toestand te brengen. Niet in gebruik! Processor Write (PWR) geeft aan dat de signalen op de data-bus geldig zijn en kunnen worden geaccepteerd door het geadresseerde geheugen of uitgangssignaal.
66	FLAG*	Deze lijn presenteert het FLAG-signaal van de microprocessor 2650.			Processor Data Bus In (PDBIN) geeft aan dat de microprocessor 2650 data leest van het geadresseerde geheugen óf van een INPUT-apparaat. PDBIN kan worden gebruikt om de busdrivers van een geselecteerd apparaat te activeren.
67	USENSE*	User Sense. Deze pen is verbonden met de 2650 SENSE-pen en kan gebruikt worden voor het invoeren van seriële data. FLAG en SENSE maken deel uit van het PSW.			
68	MWRITE	Het Memory Write (MWRITE)-signaal geeft aan dat data in het geheugen kan worden geschreven op die plaats die de adresbus aangeeft.	79	A0	Adres-bit 0.
69 t/m 71		Niet in gebruik!	80	A1	Adres-bit 1.
72	PRDY	Het Processor Ready (PRDY)-signaal is met XRDY in een OF-schakeling samengevoegd, waarvan het resultaat het statussignaal OPACK (Operation Acknowledge) van de microprocessor 2650 is. PRDY wordt teruggezonden door een geadresseerd periferie-apparaat (hetzij geheugen, dan wel I/O), óf door een randapparaat, nadat de data-overdracht na een interruptie is geschied.	81	A2	Adres-bit 2.
			82	A6	Adres-bit 6.
			83	A7	Adres-bit 7.
			84	A8	Adres-bit 8.
			85	A13	Adres-bit 13.
			86	A14	Adres-bit 14.
			87	A11	Adres-bit 11.
			88	DO2	Data OUT bit 2.
			89	DO3	Data OUT bit 3.
			90	DO7	Data OUT bit 7.
			91	DI4	Data IN bit 4.
			92	DI5	Data IN bit 5.
			93	DI6	Data IN bit 6.
			94	DI1	Data IN bit 1.
			95	DI0	Data IN bit 0.
73	PINT	Processor Interrupt (PINT) is een S100 interface signaal dat overeenkomt met het Interrupt Request signaal van de microprocessor 2650. Deze bevestigt PINT, als de instructie, waarmee hij bezig was toen het PINT signaal "laag" werd, is voltooid. De microprocessor 2650 herkent PINT niet indien hij in de WACHT-toestand is óf indien er het INTERRUPT INHIBIT-bit (II) in het PSW in	96 t/m 98 99	POR	Niet in gebruik! Power On Reset (POR) is een uitgangssignaal dat aangeeft dat de stroomvoorziening is aangesloten op de INSTRUCTOR 50, en dat het systeem in de 0-stand is gebracht. Het uitgangssignaal POR kan worden gebruikt om de periferie-apparatuur desgewenst eveneens in de 0-stand te brengen.
			100	GND	Ground. Systeemaarde.



## DE INTERFACE NAAR DE CASSETTE-RECORDER

Op de INSTRUCTOR 50 kan een cassette-recorder aangesloten worden om de in het geheugen geregistreerde data vast te leggen op een magnetische cassetteband. Deze mogelijkheid is van groot belang als men een wat groter programma heeft gemaakt en het programmeren op een bepaald moment wenst te beëindigen. Het nadeel is dan dat bij het wegvallen van de stroomvoorziening, ook de informatie uit de RAM geheugens verdwijnt. Indien men de informatie vóór het uitschakelen van de INSTRUCTOR 50 zou kunnen registreren, dan gaat deze niet verloren zodat ze dan later opnieuw aan de machine kan worden toevertrouwd. Voor dat doel kan een normale audio-cassette-recorder gebruikt worden.

Hiertoe worden twee snoeren meegeleverd, die gebruikt kunnen worden bij de in- en uitvoer van gegevens naar en van de cassette-recorder.

Op één cassette kunnen verscheiden programma's geregistreerd worden omdat de programma's ieder een eigen nummer krijgen, het z.g. "FILE" nummer. Voordat men begint met het registreren, resp. lezen, moet de cassetteband ongeveer op de juiste plaats worden gebracht. Men dient te voorkomen dat bij een reeds eerder geregistreerde file de data overschreven wordt, zodat men er verstandig aan doet, tussen de diverse files een behoorlijke ruimte open te laten.

Als men het ingangssignaal kan regelen d.m.v. een volumeregelaar, dan dient deze ongeveer de stand in te nemen waarbij men normaal een audio-signaal registreert. Er zijn enkele richtlijnen waarmee men rekening moet houden bij het schrijven, het afregelen en het lezen van de cassette.

### *Belangrijke opmerking*

Sommige typen cassette-recorders hebben een zeer snel werkende automatische volumeregeling. Deze zijn minder geschikt om met de INSTRUCTOR 50 te worden gebruikt. Bij de afregelprocedure komt dit tot uiting door het uitblijven van een stabiele "-" indicatie op het scherm en zal men uitsluitend een "d" of een "U" op het scherm zien.

Voorts kan het voorkomen dat, bij gebruik van slechte soorten cassetteband, voortdurend problemen optreden bij het inlezen van programma's in de INSTRUCTOR 50. Doorgaans ziet men dan "ERROR 6" of "ERROR 4" op het scherm verschijnen. De oplossing voor dit probleem is erg eenvoudig; gebruik uitsluitend cassettes van een goede kwaliteit!

### Schrijven op de cassette

De WCAS-toets (Write Cassette) biedt de mogelijkheid om een programma klaar te maken voor het schrijven op een cassetteband. Voor dat doel wordt eerst deze toets ingedrukt, zie fig. 14 a), zodat op het scherm L.Ad.= ver-

schijnt. De INSTRUCTOR 50 wacht nu op het adres van het eerste byte dat moet worden geregistreerd. In vrijwel alle gevallen zal dit byte H'0000' zijn, zodat men kan volstaan met het indrukken van de hexadecimale toets 0, zie b). Vervolgens drukt men ENT/NXT in, zie c), om te bevestigen dat dit het adres is waar zich het eerste geregistreerde byte bevindt. Vervolgens moet men het adres aangeven van het laatste byte dat geregistreerd moet worden. Bij het indrukken van ENT/NXT, zie c), vermeldt het scherm U.Ad.=, d.w.z. dat de INSTRUCTOR 50 wacht op het adres van het laatste byte dat geregistreerd moet worden. Stel dat dit laatste byte zich op het adres H'76' bevindt. Men kan dit adres nu invoeren, zie d) en e), en bevestigen door ENT/NXT in te drukken. Op het scherm verschijnt dan S.Ad.=. Men kan nu aan de INSTRUCTOR 50 het startadres van het programma toevertrouwen. Dit startadres behoeft niet hetzelfde te zijn als het adres van het eerste byte van de te registreren informatie. Het kan immers voorkomen dat men in de eerste 64 bytes van het geheugen een aantal adressen heeft geplaatst in verband met de instructie ZBSR, die gebruik maakt van de eerste 64 bytes van het geheugen voor subroutines, resp. voor verwijzing naar subroutines. Het zou dus zeer wel mogelijk zijn dat H'10' (= adres 16 decimaal) het adres is waar zich de eerste instructie van het gekozen programma bevindt. Door dit adres aan de cassette-recorder toe te vertrouwen, wordt straks het instructie-adresregister automatisch in de juiste stand geplaatst. Na het invoeren van het startadres, zie g) en h), drukt men de toets ENT/NXT in, zie i), waarna .F= op het scherm verschijnt. Nu kan men een filenummer invoeren, zodat de geregistreerde informatie wordt voorzien van een nummer en het mogelijk wordt meer dan één file op één magnetische band te plaatsen. In ons geval nemen we aan, dat dit filenummer 1 is en we slaan dit cijfer dan ook aan, zie j).

*Nu wordt de cassette-recorder gestart en de toets ENT/NXT ingedrukt, zie k).* Het duurt dan nog ongeveer 5 seconden vóórdat het programma op de cassette wordt geschreven. Gedurende deze tijd gaat de FLAG LED aan en uit, en wel ongeveer eens per seconde. Is de registratie uit het geheugen volledig op de band overgebracht, dan geeft de INSTRUCTOR 50 dit aan met het woord "HELLO", zie l). Men kan de cassette-recorder nu stoppen. Het is verstandig om het nummer te noteren van de stand van het telwerk van de cassette-recorder, zodat men later bij het lezen van informatie, deze plaats bij benadering kent. Aldus wordt ook voorkomen dat de volgende file de nu geschreven informatie verdringt.

Het opnemen van de informatie uit het geheugen op de band kan worden waargenomen op de acht lampjes links op het bedieningspaneel. Hiertoe moet de 3-standen schakelaar in de stand "EXTENDED" worden geplaatst.

## Afregelen van de cassette

Wil men de cassette gaan lezen, dan is het van belang dat het ontvangen signaal het juiste niveau heeft. Om de informatie uit de cassette wederom naar het geheugen over te dragen, moet men de cassette doorspoelen tot de stand bereikt is waar de over te dragen informatie staat. De uitgang van de cassette-recorder moet d.m.v. het bijgeleverde snoer op de "phone"-bus worden aangesloten. Om de cassette af te regelen, drukt men nu eerst op de toets REG, zie fig. 15 a). De INSTRUCTOR 50 wacht nu op het invoeren van nadere informatie. De opdracht tot het afregelen wordt gegeven door het indrukken van de toets A van het hexadecimale toetsenveld, zie b). Nu verschijnt bij het afspelen van de cassetterecorder op het scherm een letter. De letter U betekent dat het niveau te laag is en men het volume dus dient op te voeren. Verschijnt een d, dan wijst dat op een te hoog niveau van het signaal; men dient dat dan te verlagen. Het teken — betekent dat het juiste niveau is bereikt. Zo nodig moet men enkele keren terug naar het begin van de file om deze afregeling te herhalen, vooral ingeval betrekkelijk weinig informatie op de cassette is geplaatst. Nadat men het afregelen heeft verricht, d.w.z. dat een streepje op het scherm verschijnt, kan men op de knop MON drukken, zie e). Naast de informatie die op het scherm verschijnt, kan men ook nog d.m.v. de LED's van de I/O-poort waarnemen dat het niveau de juiste waarde heeft. Daartoe zet men de I/O-schakelaar in de middenstand (EXTENDED). De LED's hebben nu de volgende betekenis:

<i>Alle LED's uit:</i>	het niveau is <i>goed</i>
<i>De meest linkse LED aan en enkele andere uit:</i>	het niveau is te <i>laag</i>
<i>De linkse LED uit en enkele LED's aan:</i>	het niveau is te <i>hoog</i>

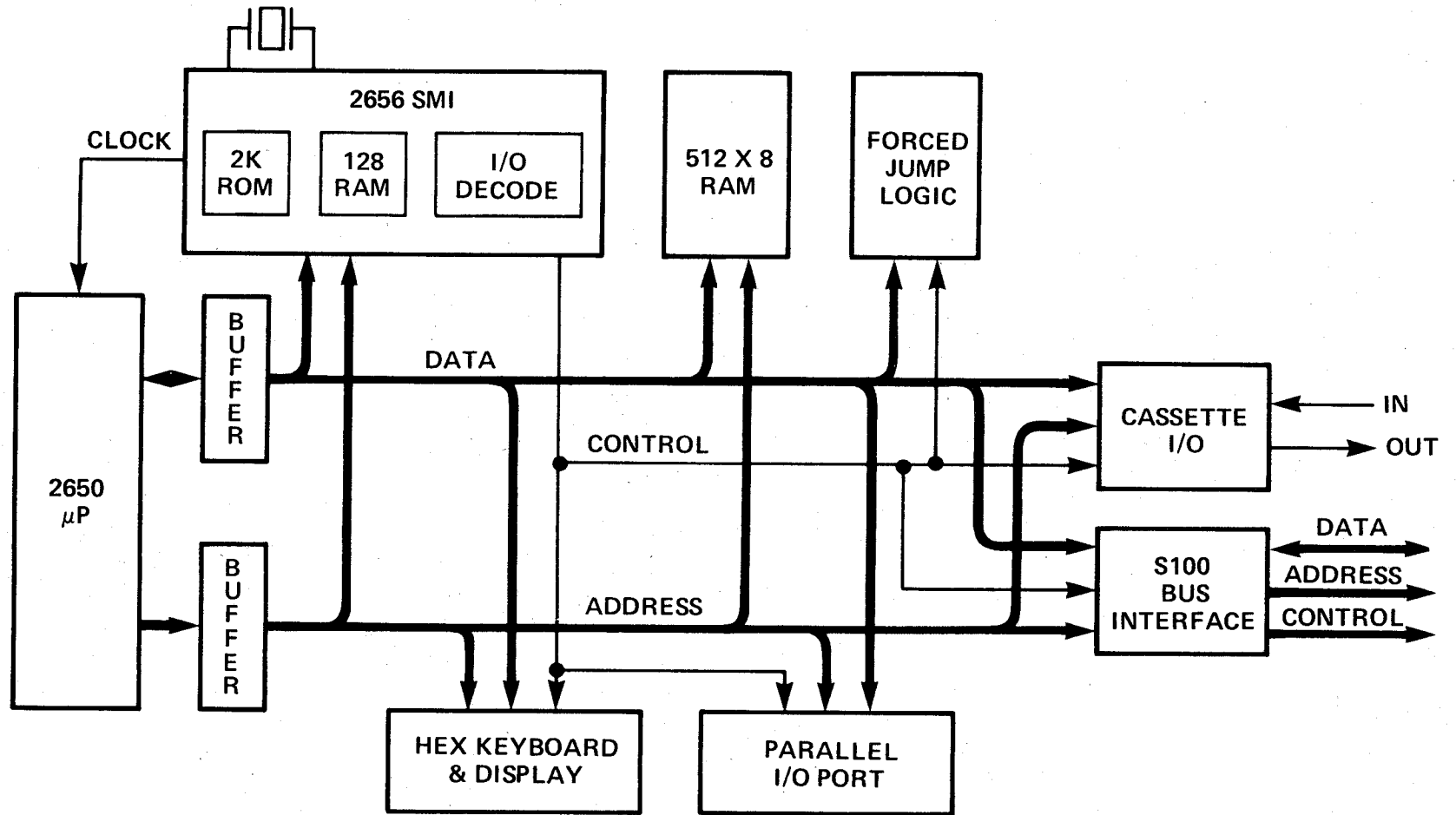
## Lezen van de cassette

Om de informatie van de cassette weer over te dragen naar het geheugen, moet ze gelezen worden. Het lezen is al voorbereid door het afregelen van de cassette, en we nemen aan dat de juiste file vlak voor de leeskop aanwezig is.

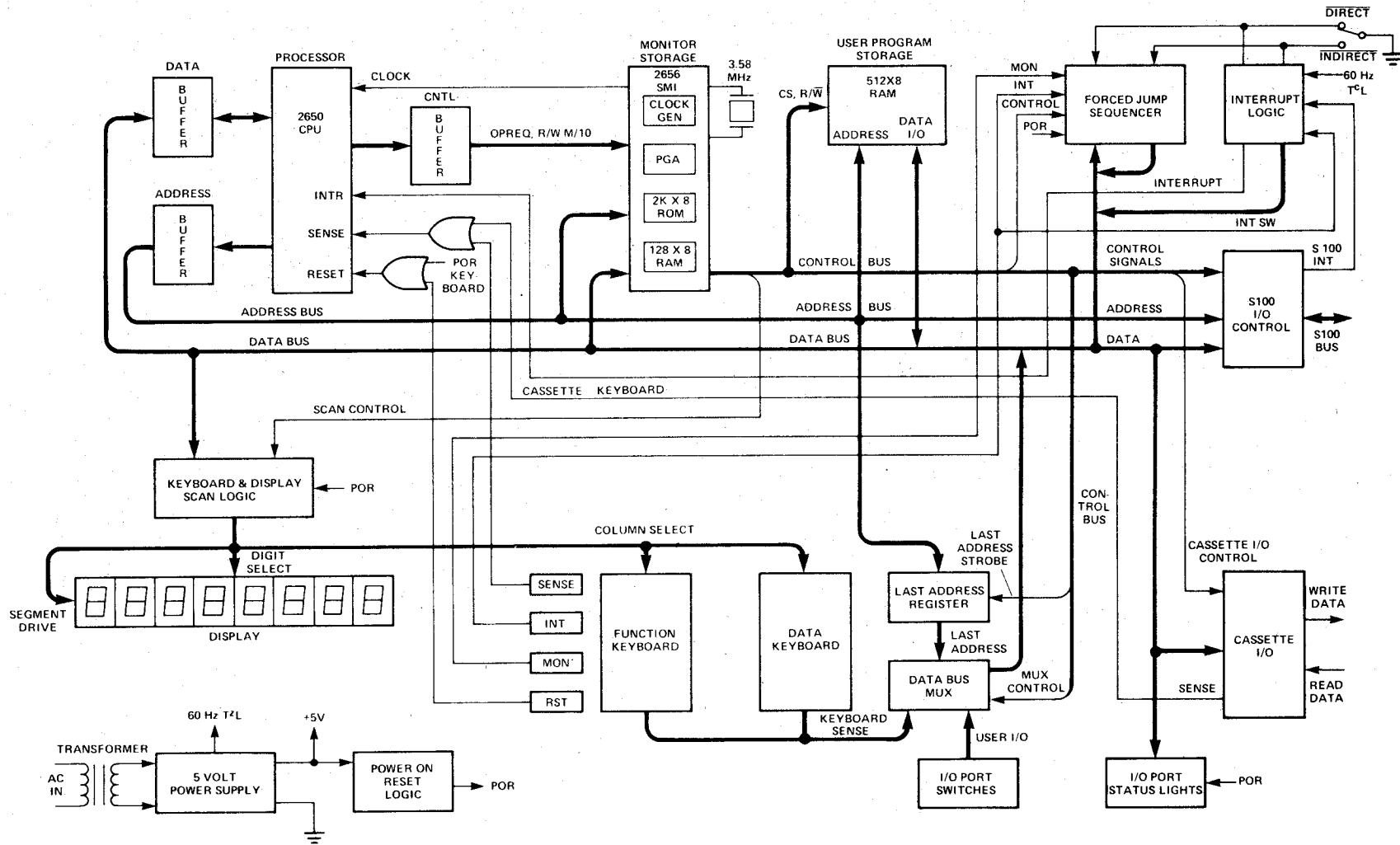
Nu wordt, zie fig. 16, de toets RCAS ingedrukt, zie a). Op het scherm verschijnt nu .F=, hetgeen betekent dat de INSTRUCTOR 50 op het nummer van de file wacht. Men kan nu het filenummer aanslaan. In ons voorbeeld is dit 1, zie b). Op het scherm verschijnt dan .F=1. Men kan het filenummer bevestigen door de ENT/NXT-toets in te drukken, zie c).

Nu kan de cassette-recorder gestart worden. Na afloop van het lezen verschijnt op het scherm het woord "HELLO". De cassette-recorder kan dan gestopt worden.

Het lezen van informatie van de band naar het geheugen kan worden waargenomen op de lampjes links op het bedieningspaneel indien de 3-standen schakelaar in de middenstand "EXTENDED" wordt gezet.



Opbouw van de INSTRUCTOR 50.



Gedetailleerde opbouw van de INSTRUCTOR 50.

# GEHEUGEN EN I/O-ORGANISATIE

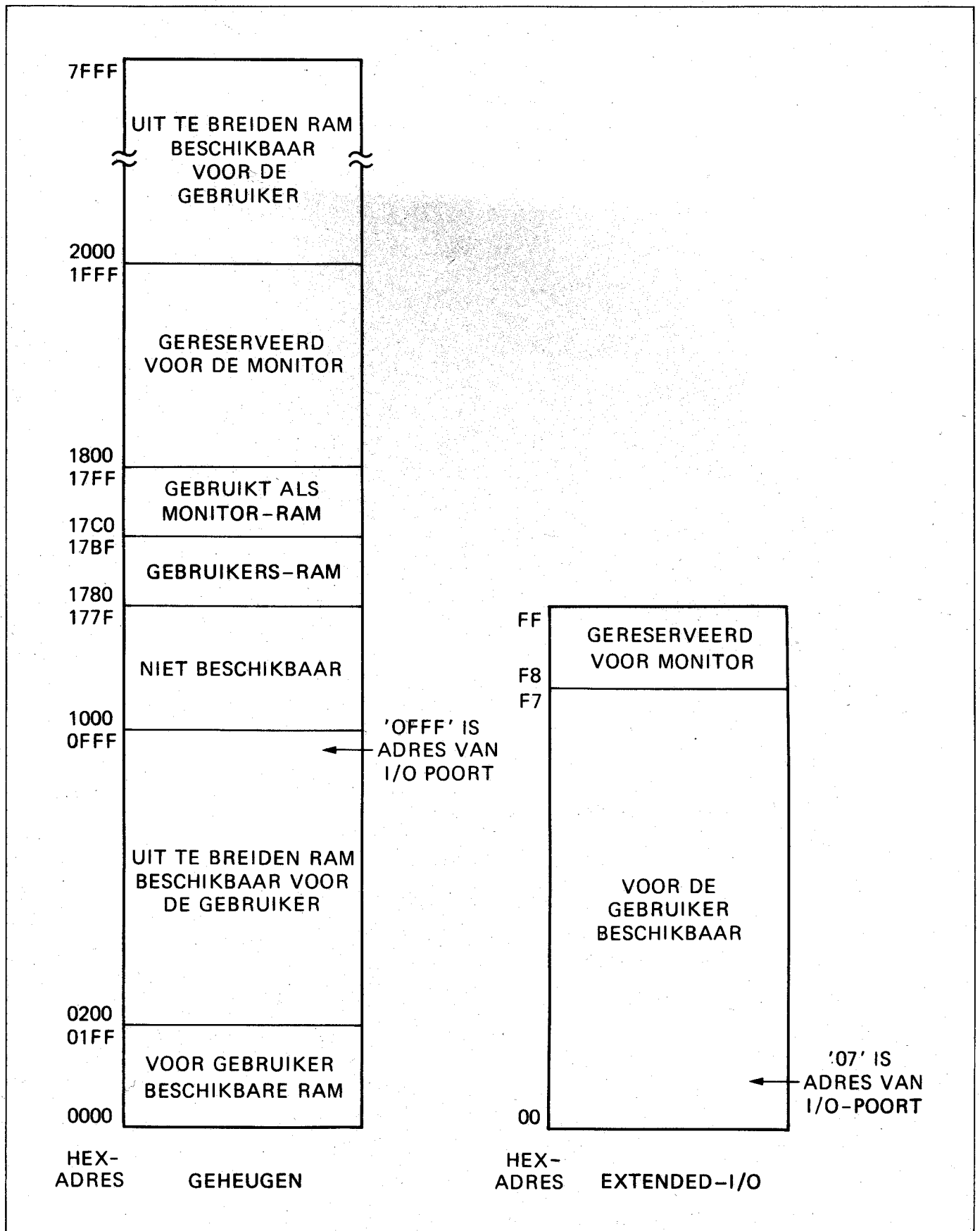


Fig. 4 Geheugen en I/O organisatie.

# LEZEN VAN CPU-REGISTERS EN PSW

(Program Status Word)

	TOETS	SCHERM	COMMENTAAR
a)	REG	r =	DE INSTRUCTOR WACHT OP EEN NUMMER
b)	4	.r4 = A8	DE INHOUD VAN REGISTER 1 BANK 1 IS H'A8'
c)	ENT NXT	.r5 = B6	DE INHOUD VAN REGISTER 2 BANK 1 IS H'B6'
d)	ENT NXT	.r6 = A4	DE INHOUD VAN REGISTER 3 BANK 1 IS H'A4'
e)	ENT NXT	.PU = 0A	DE INHOUD VAN HET PROGRAM STATUS WORD UPPER IS H'0A'
f)	ENT NXT	.PL = 21	DE INHOUD VAN HET PROGRAM STATUS WORD LOWER IS H'21'

DE PSW-INHOUD IS DUS H'0A21'

Fig. 5 Lezen van CPU-registers en PSW.

# LEZEN EN VERANDEREN VAN CPU-REGISTERS EN PSW

(Program Status Word)

	TOETS	SCHERM	COMMENTAAR
a)	REG	r =	DE INSTRUCTOR WACHT OP EEN NUMMER
b)	4	.r4 = A8	DE INHOUD VAN REGISTER 1 BANK 1 IS H'A8' DEZE A8 IS TOEVALLIG
c)	E	.r4 = E	DE INHOUD VAN REGISTER 1 BANK 1 WORDT GEWIJZIGD IN H'OE'
d)	2	.r4 = E2	DE INHOUD VAN REGISTER 1 BANK 1 WORDT GEWIJZIGD IN H'E2'
e)	ENT NXT	.r5 = B6	DE INHOUD VAN REGISTER 1 BANK 1 IS NU DEFINITIEF H'E2' GEWORDEN DE INHOUD VAN REGISTER 2 BANK 1 WORDT NU GETOOND EN ZOU TOEVALLIG H'B6' KUNNEN ZIJN

Fig. 6 Lezen en veranderen van CPU-registers en PSW.

## LEZEN VAN GEHEUGENCELLEN

	TOETS	SCHERM	COMMENTAAR
a)	MEM	.Ad. =	DE INSTRUCTOR VRAAGT OM EEN ADRES VOOR DE SELECTIE VAN EEN GEHEUGENCEL
b)	4	.Ad. = 4	HET MEEST SIGNIFICANTE CIJFER VAN HET ADRES WORDT AANGESLAGEN
c)	2	.Ad. = 42	HET VOLGENDE CIJFER WORDT AANGESLAGEN
d)	ENT NXT	.0042 A5	DE INHOUD VAN DE GEHEUGENCEL H'0042' IS H'A5'
e)	ENT NXT	.0043 B7	DE INHOUD VAN DE VOLGENDE GEHEUGENCEL (HIER H'0043') IS H'B7'
f)	ENT NXT	.0044 05	DE INHOUD VAN DE VOLGENDE GEHEUGENCEL (HIER H'0044') IS H'05'

Fig. 7 Lezen van geheugencellen.



## LEZEN EN VERANDEREN VAN GEHEUGENCELLEN

	TOETS	SCHERM	COMMENTAAR
a)	MEM	.Ad. =	DE INSTRUCTOR VRAAGT OM EEN ADRES VOOR DE SELECTIE VAN EEN GEHEUGENCEL
b)	4	.Ad. = 4	HET MEEST SIGNIFICANTE CIJFER VAN HET ADRES WORDT AANGESLAGEN EN IS 4
c)	2	.Ad. = 42	HET VOLGENDE CIJFER WORDT INGEVOERD
d)	ENT NXT	.0042 A5	DE INHOUD VAN GEHEUGENCEL H'0042' IS H'A5'
e)	6	.0042 6	OP DE INHOUD VAN GEHEUGENCEL H'0042' WORDT EEN CORRECTIE UITGEVOERD
f)	4	.0042 64	ER WORDT NOG EEN CORRECTIE UITGEVOERD
g)	ENT NXT	.0043 B7	DE INHOUD VAN H' 0042' IS NU DEFINITIEF H'64'. DE INHOUD VAN CEL H'0043' WORDT GETOOND EN IS H'B7'

Fig. 8 Lezen en veranderen van geheugencellen.

# SNEL LADEN VAN HET GEHEUGEN

	TOETS	SCHERM	COMMENTAAR
a)	REG	r =	DE INSTRUCTOR WACHT OP EEN NUMMER
b)	F	.Ad. =	H'F' WORDT AANGESLAGEN. NU VRAAGT DE INSTRUCTOR OM EEN ADRES
c)	1	.Ad. = 1	HET MEEST SIGNIFICANTE ADRES-CIJFER (H'1') WORDT AANGESLAGEN
d)	2	.Ad. = 12	HET VOLGENDE ADRES-CIJFER WORDT AANGESLAGEN
e)	ENT NXT	.0012	DE GEKOZEN GEHEUGENCEL IS H'0012'
f)	A	.0012 A	HET EERSTE CIJFER WORDT IN DE H'0012'-CEL AANGESLAGEN
g)	1	.0012 A1	HET TWEDE CIJFER WORDT AANGESLAGEN
h)	F	.0013 F	HET EERSTE CIJFER VAN DE VOLGENDE CEL (H'0013') WORDT AANGESLAGEN
i)	E	.0013 FE	HET TWEDE CIJFER WORDT AANGESLAGEN IN CEL H'0013'
		enz.	
j)	MEM	.Ad. =	DE SNELLE INVOER WORDT BEEINDIGD EN DE STAP-VOOR-STAP INVOER (IN FIG. 8) WORDT NU UITGEVOERD

Fig. 9 Snel laden van het geheugen.

# TONEN EN VERANDEREN VAN HET IAR

(Instructie Adres Register)

	TOETS	SCHERM	COMMENTAAR
a)	REG	r =	DE INSTRUCTOR 50 VRAAGT OM EEN NUMMER
b)	C	.PC = 0017	DE INHOUD VAN HET IAR (PC) IS H'0017'
c)	2	.PC = 2	HET IAR WORDT GEWIJZIGD
d)	8	.PC = 28	HET VOLGENDE CIJFER VOOR HET IAR WORDT AANGESLAGEN
e)	ENT NXT	r =	HET IAR IS DEFINITIEF GELADEN MET H'0028'

Fig. 10 Tonen en veranderen van het IAR.

## STAP-VOOR-STAP FUNKTIE

	TOETS	SCHERM	COMMENTAAR
a)	REG	r =	DE INSTRUCTOR VRAAGT OM EEN NUMMER
b)	C	.PC = 0017	DE INHOUD VAN HET IAR WORDT GETOOND (H'0017')
c)	1	.PC = 1	HET IAR WORDT GEWIJZIGD
d)	2	.PC = 12	HET VOLGENDE CIJFER VAN HET IAR
e)	2	.PC = 122	HET VOLGENDE CIJFER VOOR HET IAR
f)	ENT NXT	r =	HET IAR IS DEFINITIEF MET H'0122' GELADEN
g)	STEP	0124 CA	DE INSTRUCTIE OP H'0122' WORDT UITGEVOERD EN IS 2 BYTES GROOT; DE VOLGENDE INSTRUCTIE DIE UITGEVOERD GAAT WORDEN IS OP H'0124' EN IS H'CA' (STRR,R2)
h)	STEP	0126 CC	DE INSTRUCTIE OP H'0124' WORDT UITGEVOERD EN IS 2 BYTES GROOT; DE VOLGENDE INSTRUCTIE BEGINT IN CEL H'0126' EN LUIDT H'CC' (STRA,R0)
		⋮	
i)	REG	r =	DE STAP-VOOR-STAP FUNCTIE WORDT BEËINDIGD

Fig. 11 Stap-voor-stap funktie.

# HET AANBRENGEN VAN EEN BREEKPUNT IN HET PROGRAMMA

	TOETS	SCHERM	COMMENTAAR
a)	BKPT.	.b.P =	DE INSTRUCTOR 50 VRAAGT OM HET EERSTE CIJFER VAN HET BREEKPUNT
b)	1	.b.P = 1	HET EERSTE CIJFER WORDT AANGESLAGEN
c)	5	.b.P = 15	HET TWEEDE CIJFER WORDT AANGESLAGEN
d)	4	.b.P = 154	HET DERDE CIJFER WORDT AANGESLAGEN
e)	ENT NXT	b.P = 154	HET BREEKPUNT WORDT DEFINITIEF VASTGESTELD OP ADRES H'0154'
f)	BKPT.	.b.P = 0154	DOOR HET INDRUKKEN VAN DE BKPT-TOETS WORDT HET BREEKPUNT GETOOND
g)	BKPT.	.b.P =	HET BREEKPUNT WORDT OPGEHEVEN
h)		-0154 C0	HET PROGRAMMA IS OP HET BREEKPUNT OP ADRES H'0154' AANGEKOMEN EN LAAT DE VOLGENDE INSTRUCTIE H'C0' (NOP) ZIEN

Fig. 12 Het aanbrengen van een breekpunt in het programma.

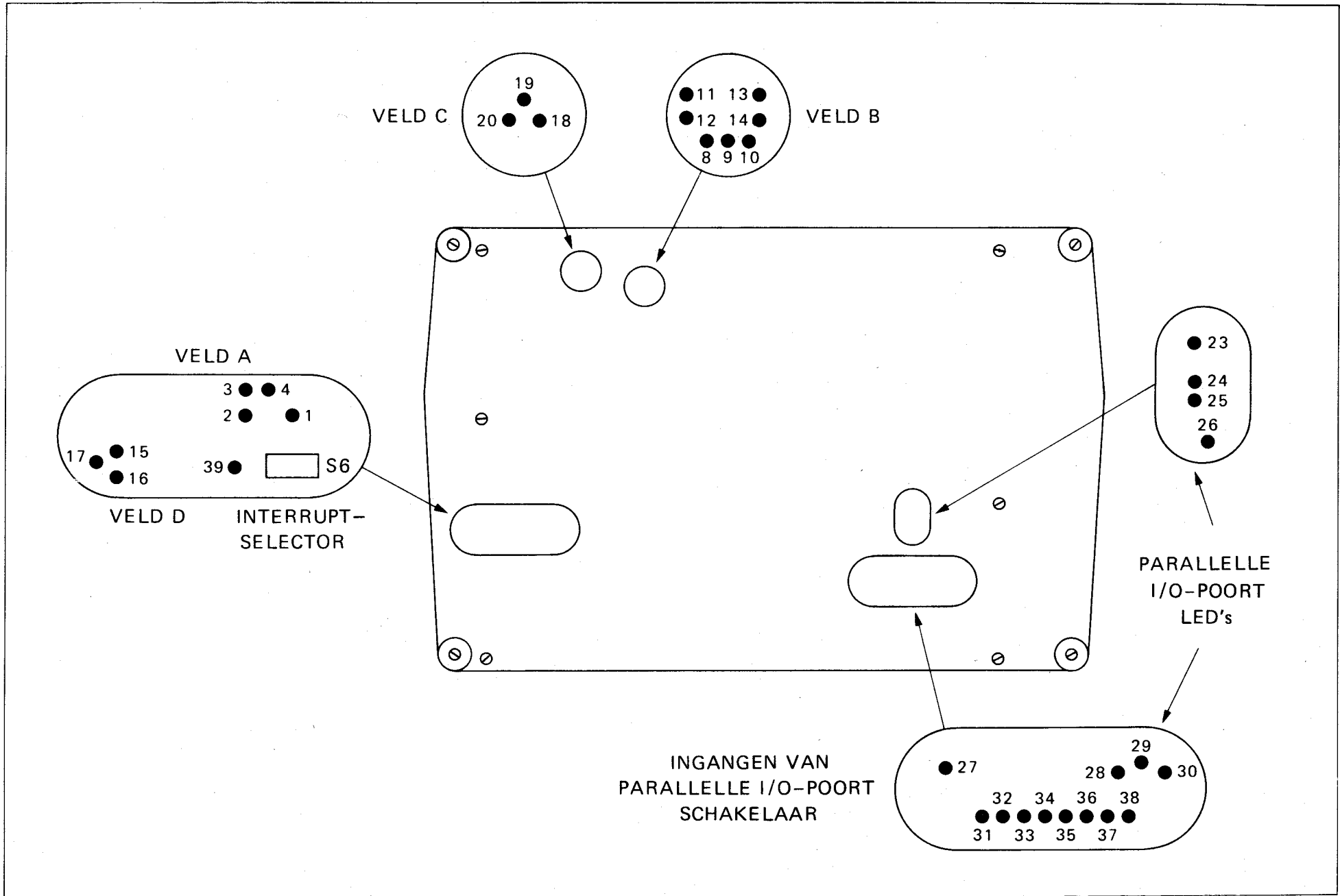


Fig. 13 Onderzijde van de INSTRUCTOR 50.

# SCHRIJVEN OP DE CASSETTE

TOETS	SCHERM	COMMENTAAR
a) <input type="text" value="WCAS"/>	<input type="text" value="L.Ad. ="/>	DE INSTRUCTOR WACHT OP HET EERSTE CIJFER VAN HET MEEST SIGNIFICANTE DEEL VAN HET LAAGSTE ADRES VAN HET PROGRAMMA
b) <input type="text" value="0"/>	<input type="text" value="L.Ad. = 0"/>	LAAGSTE ADRES WORDT AANGESLAGEN H'0000'
c) <input type="text" value="ENT&lt;br/&gt;NXT"/>	<input type="text" value="U.Ad. ="/>	DE INSTRUCTOR WACHT OP HET EERSTE CIJFER VAN HET MEEST SIGNIFICANTE DEEL VAN HET HOOGSTE ADRES VAN HET PROGRAMMA
d) <input type="text" value="7"/>	<input type="text" value="U.Ad. = 7"/>	} HET HOOGSTE ADRES WORDT AANGESLAGEN H'0076'
e) <input type="text" value="6"/>	<input type="text" value="U.Ad. = 76"/>	
f) <input type="text" value="ENT&lt;br/&gt;NXT"/>	<input type="text" value="S.Ad. ="/>	DE INSTRUCTOR WACHT OP HET START-ADRES VAN HET PROGRAMMA
g) <input type="text" value="1"/>	<input type="text" value="S.Ad. = 1"/>	} HET START-ADRES WORDT AANGESLAGEN H'0010'
h) <input type="text" value="0"/>	<input type="text" value="S.Ad. = 10"/>	
i) <input type="text" value="ENT&lt;br/&gt;NXT"/>	<input type="text" value=".F ="/>	DE INSTRUCTOR WACHT OP EEN FILE-NUMMER
j) <input type="text" value="1"/>	<input type="text" value=".F = 1"/>	FILE-NUMMER WORDT AANGESLAGEN H'01'
k) <input type="text" value="ENT&lt;br/&gt;NXT"/>		DATA WORDT OVERGEDRAGEN OP DE CASSETTEBAND
l) <input type="text" value="HELLO"/>		DATA IS OVERGEDRAGEN OP DE CASSETTEBAND

Fig. 14 Schrijven op de cassette.

## AFREGELLEN VAN HET SIGNAAL-NIVEAU VAN DE CASSETTE

	TOETS	SCHERM	COMMENTAAR
a)	REG	r =	DE INSTRUCTOR WACHT OP EEN NADERE SPECIFICATIE
b)	A	U	VERHOOG TERUGSPEEL-NIVEAU (VOLUMEREGELAAR OP CASSETTE-RECORDER)
c)		d.	VERMINDER TERUGSPEEL-NIVEAU (VOLUMEREGELAAR OP CASSETTE-RECORDER)
d)		-	HET AFSPEELNIVEAU IS JUIST; LAAT DE VOLUMEREGELAAR IN DEZE STAND STAAN
e)	MON	HELLO	BEËINDIG HET AFREGELLEN

Fig. 15 Afregelen van het signaal-niveau van de cassette.



## LEZEN VAN DE CASSETTE

	TOETS	SCHERM	COMMENTAAR
a)	RCAS	.F =	DE INSTRUCTOR WACHT OP EEN NADERE OPDRACHT
b)	1	.F = 1	HET FILE-NUMMER WORDT AANGESLAGEN H'01'
c)	ENT NXT		HET PROGRAMMA WORDT IN HET GEHEUGEN GELADEN
d)		HELLO	HET PROGRAMMA IS IN HET GEHEUGEN GELADEN

Fig. 16 Lezen van de cassette.