

MICRO-PROFESSOR

MPF-I HANDLEIDING



Copyright © 1982 Sciento b.v.

Dit boek is vertaald en gedrukt met voorafgaande toestemming van Multitech Industrial Corporation. Geen enkel deel van deze publicatie mag gereproduceerd worden op geen enkele manier : elektronisch, mechanisch, gefotokopieerd, opgeslagen, overgestuurd of anders zonder toestemming van Multitech Industrial Corporation en Sciento b.v.

2e druk
1 mei 1984

Sciento b.v.

Speldenmakerstraat 10c
5232 BG 's-Hertogenbosch, The Netherlands
Tel 073-424055 Telex 50766
KvK 's-Hertogenbosch 43161

VOORWOORD.

Deze handleiding is bedoeld om de gebruiker wegwijs te maken in het gebruik van de MICRO-PROFESSOR.

Het is vrij vertaald uit het Engels door ir.W.B.Koekkoek, die met zijn veeljarige en veelzijdige onderwijservaring het oorspronkelijke boek zodanig aangepast heeft dat het nu geschikt is voor een brede kring van gebruikers. Dit is bereikt door o.a. in te gaan op de verschillende talstelsels waardoor de gebruikte machinetaal veel van zijn mysteries verloren heeft. Daarnaast zijn interessante oefeningen in de tekst verwerkt waardoor de gebruiker al spelenderwijs vertrouwd raakt met het gebruik van de machinetaal, de Z-80[®] microprocessor en de MICRO-PROFESSOR.

Samen met de vele voorbeelden is deze handleiding uitgegroeid tot een didactisch verantwoord geheel waar menigeen plezier aan kan beleven.

De illustraties zijn van H. van Doorn en zullen ongetwijfeld helpen Uw studie te veraangenamen.

U wordt veel studiegenot en gebruiksplezier gewenst.

GEFELICITEERD!

MICRO-PROFESSOR zal U binnenleiden in de wereld van de microprocessor.

Wist U dat er voor Uw MICRO-PROFESSOR een grote reeks uitbreidingen te verkrijgen zijn :

1. SPEECH SYNTHESIS BOARD :

"Good morning, its now 7'o clock am."

Het nieuwste! Gebaseerd op de spraakchip van TI. Voor het genereren van spraakcommando's of complete zinnen. Standaard 35 woorden met sprekende klokprogramma en spraakroutine. Uitbreiding mogelijk tot 150 Engelse woorden, keuze uit 400 woorden.

2. EPROM PROGRAMMER BOARD :

Om zelf Uw EPROMS te programmeren voor de MICRO-PROFESSOR of andere computers. Voor de EPROM 2508, 2516, 2532 resp. 2758, 2716, 2732. Read, copy, list en verify mogelijkheid. Met 4K RAM, direkt adresseerbaar.

3. 2K TINY BASIC INTERPRETER :

Leer nu Basic met deze prijstopper. Inclusief handboek 30blz met voorbeeldprogramma's, masker voor basiccommando's op toetsenbord en handige sticker voor het display format. Mogelijkheid om I/O te adresseren en subroutines in machinaal taal aan te roepen.

Andere opties zijn :

1. Z-80 CTC + Z-80 PIO :

De counter timer circuit voor tel, interrupt of synchronisatie doeleinden en de parallel input/output om twee 8 bits poorten te programmeren.

2. MPF-BBD : experimenteerbord :

Het basisbord als hulp bij het zelf maken van kleine schakelingen.

3. 2K RAM :

Voor het uitbreiden van het MICRO-PROFESSOR geheugen. De socket is aanwezig.

4. 2K EPROM - 4K EPROM :

Standaard programma's kunnen nu direkt beschikbaar zijn b.v. voor besturingen.

Voor Uw verder ondersteuning zijn nog een aantal interessante informatica boeken verkrijgbaar :

- Praktische Inleiding tot de Microprocessor.
K. de Wever/H. Loobuyck
- Begincursus Z-80 Assembleertaal.
K. de Wever

De MICRO-PROFESSOR wordt voortdurend uitgebreid.
Vraag Uw dealer om informatie.

MPF-1 HANDLEIDING.

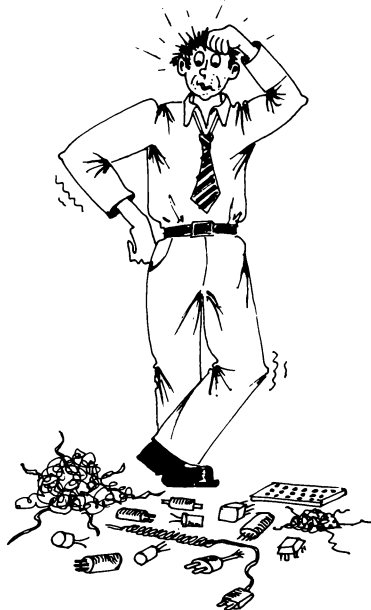
Inhoudsopgave :

<u>1. MPF-1 Specificaties.</u>	6
1.1. Wat hebt U aan onderdelen gekocht bij de aanschaf van Uw MICRO-PROFESSOR?	6
1.2. Monitorprogramma : beschrijving (kort).	10
<u>2. Algemene beschrijving.</u>	12
2.1. Functie van het monitorprogramma	12
2.2. Notatievormen gebruikt in het boek.	13
2.3. Foutmeldingen.	15
2.4. RAM adressering.	15
<u>3. Gebruiksaanwijzingen.</u>	16
<u>3.1. Basis instructies.</u>	16
3.1.1. Gebruik van de toetsen ADDR en DATA.	17
3.1.2. Uw eerste programma.	21
3.1.3. Programmeren van Uw eigen naam.	24
3.1.4. Tellen en talstelsels.	26
3.1.5. Uw tweede programma.	34
3.1.6. Registers.	35
3.1.7. Programmateller : PC-toets.	40
<u>3.2. Programma onderzoek.</u>	41
3.2.1. Starten van een programma : GO-toets.	41
3.2.2. Stap voor stap instructie : STEP-toets.	43
3.2.3. Het plaatsen van een breekpunt in het ge- bruikersprogramma : SBR-toets.	46
3.2.4. Verwijder breekpunt in het programma CBR-toets.	50
3.2.5. Onmiddellijk stoppen : MONI-toets.	51

<u>3.3.Hulpfuncties.</u>	55
3.3.1. Blokverplaatsing : MOVE-toets.	55
3.3.2. Data deletion : DEL-toets.	60
3.3.3. Data insertion : INS-toets.	62
3.3.4. Berekening van een relatief adres RELA-toets.	65
3.3.5. Opslag van programma's op cassettebandjes : TAPE WR-toets.	70
3.3.6. Inlezen data van de cassetteband TAPE RD-toets.	73
<u>4. Software en hardware beschrijving.</u>	75
4.1. Opbouw geheugen adressen.	76
4.2. Ingangs- en uitgangadressen.	78
4.3. Programma interrupt.	81
4.4. De software stop instructie.	82
4.5. Het hulpgeheugen (de STACK).	83
4.6. De RESET functie.	84
4.7. Opbouw van de data op de cassetteband.	85
<u>5. Beschrijving van de subroutines van het monitor- programma.</u>	87
5.1. Overzicht van de te gebruiken hulpfuncties.	87
5.2. Beschrijving van de hulpfunctie SCAN 1.	88
5.3. Beschrijving van de hulpfunctie SCAN.	89
5.4. Beschrijving van de hulpfunctie HEX7.	89
5.5. Beschrijving van de hulpfunctie HEX7SG.	90
5.6. Beschrijving van de hulpfunctie RAMCHK.	90
5.7. Beschrijving van de hulpfunctie TONE.	91
5.8. Beschrijving van de hulpfunctie TONE 1K.	91
5.9. Beschrijving van de hulpfunctie TONE 2K.	92

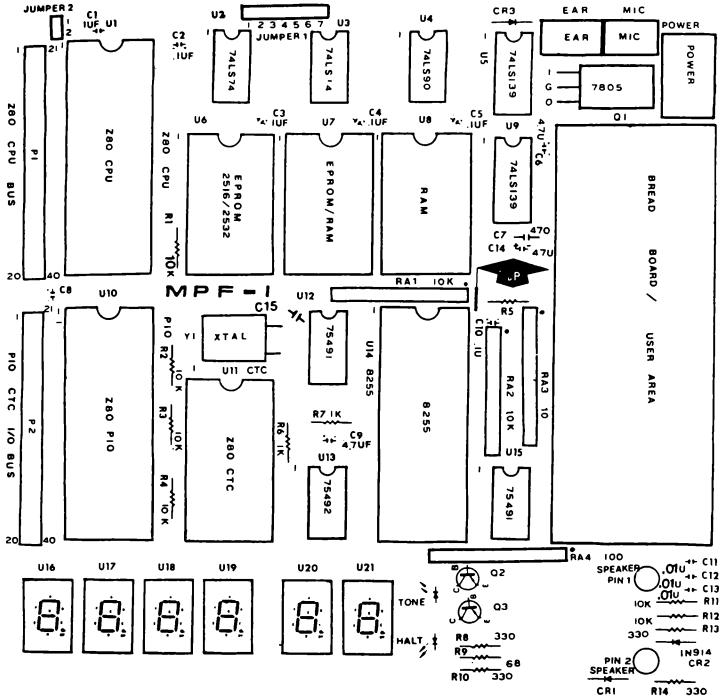
1. MPF-1 SPECIFICATIES.

1.1 Wat hebt U aan onderdelen gekocht bij de
aanschaf van Uw MICRO-PROFESSOR?



- a) Een printkaart waarop de onderstaande hoofdonderdelen zijn aangebracht (zie plattegrond fig.1.1).
- b) Een centrale processor (CPU), de Zilog Z-80 microprocessor met 158 ingebouwde instructies. (U1 in fig.1.1)
De maximale snelheid van deze processor is 2,5MHz. In de MICRO-PROFESSOR is deze snelheid op 1,79MHz ingesteld door middel van de systeemklok.
- c) Een geheugeneenheid met vooraf ingestelde informatie waaruit alleen gelezen kan worden (ROM, U6 in fig.1.1).

Figuur 1.1. : Plattegrond Computerkaart



Dit vaste geheugen bevat het monitor programma dat nodig is voor het gebruik van de MICRO-PROFESSOR. Dit monitor programma staat op de adressen 0000 tot 07FF

- d) Een willekeurig te benaderen geheugeneenheid waarin geschreven en uitgelezen kan worden door U als gebruiker (RAM, U8 in fig.1.1). Dit geheugen is bereikbaar op de adressen 1800 tot 1FFF.
- e) Tevens is aanwezig een uitbreidingsplaats (U7 in fig.1.1) waarmee het geheugen, onder punt d genoemd, is uit te breiden. De adressen voor deze geheugenuitbreiding zijn al reeds vastgelegd op de plaatsen 2000 tot 2FFF.
- f) Een in- en uitgangseenheid met in totaal 24 in en/of uitgangslijnen (U14 in fig 1.1).

Als U straks het systeem uit wilt breiden dan kan op de plaats U10 in fig.1.1 een extra, programmeerbare ingang-uitgangseenheid worden geplaatst (de PIO).

Het aantal in-uitgangen neemt daarmee toe met 16.

Naderhand is ook als extra te plaatsen een programmeerbare teller op de plaats (U11 in fig.1.1) Dit is de CTC.

Opgemerkt wordt nog dat deze twee laatst genoemde extra's niet nodig zijn voor de normale werking van Uw MICRO-PROFESSOR.

- g) Een uitlezing bestaande uit 6 cijfers welke rood oplichten (U16-U21 in fig.1.1).
- h) Een toetsenbord met daarop 36 toetsen. Hiervan worden er 19 gebruikt als functietoetsen en 16 als hexa-decimaal toetsen. De functie van één toets kan door U als gebruiker zelf worden bepaald.

- i) Een luidspreker welke ook in de door U zelf gemaakte programma's kan worden gebruikt.
- j) Een gebruikersgebied (user area). Hierop kunt U zelf naar behoefte elektronische componenten plaatsen voor speciale schakelingen.
- k) Een koppelschakeling welke de MICRO-PROFESSOR kan koppelen aan een gewone cassette recorder voor het opnemen/weergeven van door U ontwikkelde programma's (via pluggen ear, mic)
- l) Een systeemklok met een kristal oscillator welke een frequentie opwekt van 3,58MHz na deling door 2 ontstaat de frequentie van 1,79MHz die de snelheid van de MICRO-PROFESSOR bepaalt.
- m) Een netvoedingsadapter voor 220V wisselspanning die uitgaand 9V afgeeft bij een stroom van 600mA.
Het systeem dat voor U ligt gebruikt zonder uitbreidingen ongeveer 500mA.

1.2. Monitorprogramma beschrijving (kort)

Zoals reeds vermeld zit er in Uw MICRO-PROFESSOR een eenheid met vooringestelde programmatuur. Dit is het zogenaamde monitorprogramma en is speciaal ontworpen om een eenvoudig gebruik van Uw MICRO-PROFESSOR te bevorderen.

Nadat U de netspanning hebt ingeschakeld begint Uw MICRO-PROFESSOR direkt te werken in zijn monitorprogramma en de invoertoetsen zijn nu direkt te gebruiken.

Hieronder volgt een korte beschrijving van de toets functies :

- (1) RS : systeem reset functie, dit brengt het systeem in zijn begintoestand
- (2) ADDR : het bepalen van het geheugen-adres
- (3) REG : het bepalen van de registernaam
- (4) DATA : het invoeren van gegevens naar geheugenplaats of register
- (5) PC : het oproepen van de programmataeller
- (6) + : het verhogen van het adres van geheugen of register met één
- (7) - : het verlagen van het adres van geheugen of register met één
- (8) STEP : het stap voor stap doorlopen van een door U zelf ontwikkeld programma
- (9) SBR : het inbrengen van het stopadres in een door U zelf ontwikkeld programma
- (10) CBR : het verwijderen van het stopadres in een door U zelf ontwikkeld programma
- (11) MONI : het onmiddellijk stoppen in een door U zelf ontwikkeld programma

- (12)

GO

 : het starten van een door U zelf ontwikkeld programma of het uitvoeren van een van de monitorfuncties

- (13)

INS

 : het toevoegen van een woord in een geheugenplaats

- (14)

DEL

 : het verwijderen van een woord uit een geheugenplaats

- (15)

MOVE

 : het verschuiven van een programmablok in zijn geheel van een bepaald geheugengebied naar een ander geheugengebied

- (16)

RELA

 : het berekenen van de relatieve adressering

- (17)

TAPE WR

 : het overbrengen van een programma naar de cassetterecorder

- (18)

TAPE RD

 : het ontvangen van een programma van de cassetterecorder

- (19)

INTR

 : het onderbreken van een programma dat loopt

- (20)

USER KFY

 : toetsfunctie te definiëren door gebruiker

- (21) : 16 hexa-decimale cijfers voor invoer of registernaam toetsen :

AF 0	BC 1	DE 2	HL 3	AF' 4	BC' 5	DE' 6	HL' 7	IX 8
IX 9	SP A	I. IF B	SZ.H C	.PNC D	SZ.H' E	.PNC' F		

2. ALGEMENE BESCHRIJVING.

2.1. Functie van het monitorprogramma.

Het monitorprogramma is ingebracht om U als gebruiker de gelegenheid te geven programma's te ontwikkelen, zonder al teveel problemen.

Deze mogelijkheden houden het volgende in :

- a) Het invoeren van een door U ontwikkeld programma in het RAM geheugengedeelte
- b) Het laten werken van een door U ingevoerd programma
- c) Het gebruik van de stap voor stapfunctie of het plaatsnemen van een stopplaats in een door Uzelf ontwikkeld programma.

Door deze functies kunt U als gebruiker het werkende programma stap voor stap of deel voor deel doorlopen. Na elke stap worden al de te controleren grootheden overgebracht naar het monitorgedeelte en al de toestanden welke in de CPU bewaard moeten worden, worden ook in het monitorgedeelte opgeslagen. U als gebruiker kunt al de registers en de onderhavige geheugenplaatsen onderzoeken op hun inhoud en indien gewenst veranderen voordat U de volgende programmastap ingaat.

Dit is een krachtig hulpmiddel om in programma's fouten op te zoeken en te verbeteren.

- d) Andere hulpfuncties, zoals cassetterecorder controle en relatieve adresberekening. Dit laatste is een prettige hulpfunctie in het monitorprogramma bij het programmeren.

2.2. Notatievormen gebruikt in dit boek



De voorbeelden en de uitleg in dit boek zijn voor het leesgemak en de eenvoud op een bepaalde wijze opgezet. Hiertoe maken we de volgende afspraken :

- a) Het symbool voor de 6-cijferige uitlezing is in figuur 2-1 te zien.

1	2	3	4	5	6
---	---	---	---	---	---

figuur 2-1

- b) Wanneer de waarde van één van de 6 cijfers er niets toe doet is dit aangegeven met een "X".
- c) Een vierkantje met daarin wat tekst staat voor een toets welke ingedrukt moet worden, zie figuur 2-2.

+	9
---	---

figuur 2-2

- d) De notatie <adres> staat voor een geheugen-plaats dat bestaat uit 4 hexadecimale cijfers. Dat adres wordt door U als gebruiker ingevoerd. Worden door U meer dan 4 cijfers ingevoerd dan neemt de MICRO-PROFESSOR de laatste 4 cijfers op. Worden er minder dan 4 cijfers ingevoerd dan wordt voor de voorste cijfers een 0 genomen.
- e) De notatie <data> staat voor een 2-cijferige code welke U als gebruiker dient in te voeren. De spelregels zijn dezelfde als genoemd onder punt d.
- f) Als een toets <adres> of <data> ingesloten is door rechte haken "[]" bijvoorbeeld [<adres>] dan mag dit overgeslagen worden.

2.3. Foutmeldingen

Als een verboden toets is ingedrukt dan blokkeert het monitorprogramma de uitlezing en is er dientengevolge niets op de uitlezing te zien : dit is een foutmelding.

Sommige programmeerfouten geven een foutmelding in het programma met op de uitlezing de tekst -ERR . Als dit laatste voorkomt moet de fout in het programma worden opgezocht en de juiste gegevens worden ingebracht.

2.4. RAM adressering

De adressen van de geheugenplaatsen van het RAM geheugen dat in de MICRO-PROFESSOR is meegeleverd lopen van 1800 tot en met 1FFF.

De geheugenplaatsen 1F93-1FF3 worden door het monitorprogramma gebruikt. Als U deze plaatsen wilt gebruiken in een door Uzelf ontwikkeld programma dient U eerst zorgvuldig hoofdstuk 4 te lezen van deze inleiding.

Als men de geheugencapaciteit wil opvoeren dan is een extra RAM geheugen zo bij te plaatsen in Uw MICRO-PROFESSOR op de plaats U7. De bij deze uitbreiding behorende geheugen adressen lopen van 2000 tot en met 27FF.

3. GEBRUIKSIINSTRUCTIES.

3.1. Basisinstructies.



Na deze inleiding gaan we praktisch oefenen met onze MICRO-PROFESSOR. Wat eenvoudige vinger-oefeningen zijn nu eenmaal nodig. Laten we eerst kijken hoe dat gaat.

Sluit de meegeleverde adapter eerst aan op het lichtnet en steek daarna de plug in de contraplug op Uw MICRO-PROFESSOR. Als alles goed is staat er nu UPF--1 op de uitlezing.

Staat dit er niet?

Druk dan de toets RS in, de uitlezing laat dan UPF--1 zien.

Door deze RS toets wordt het gehele systeem in zijn rusttoestand gebracht.

3.1.1. Gebruik van de toetsen ADDR en DATA.

De gebruiksvolgorde van deze toetsen is :

<adres> [<data>] [<data>]

Aan de hand van de nu volgende voorbeelden zullen we eens kijken wat er allemaal gebeurt en wat we er allemaal mee kunnen doen.

toets in	uitlezing	commentaar
<input type="text" value="ADDR"/>	X.X.X.X. X X <div style="text-align: center;"> </div>	de vier punten in het adresveld geven aan dat U een adres moet opgeven
<input type="text" value="A0F"/>	0.0.0.0. 0 6	de inhoud van adres 0 is 06
<input type="text" value="+"/>	0 0 0 1 0.0.	merk op dat door het indrukken van de <input type="text" value="+"/> toets het volgende adres, 0001, in het adresveld wordt geplaatst met daar achter de data die in dit adres staan.
<input type="text" value="+"/>	0 0 0 2 1.0.	inhoud adres 0002 is 10
<input type="text" value="+"/>	0 0 0 3 F.E.	inhoud adres 0003 is FE
<input type="text" value="+"/>	0 0 0 4 3.E.	inhoud adres 0004 is 3E

Door de wordt telkens het volgende hogere adres genomen, wat te doen als we eens terug willen naar het vorige lagere adres?

Nu, dat gaat net zo eenvoudig : druk de in.

<input type="text" value="-"/>	neem hier zelf de uitlezing over
<input type="text" value="-"/>	- - - - -	neem ook hier zelf de uitlezing over
<input type="text" value="-"/>	- - - - -	als boven
<input type="text" value="-"/>	- - - - -	als boven

Als U nu de overgenomen uitlezingen vergelijkt met de eerste uitlezing dan ziet U dat nu de adressen in aflopende nummering worden uitgelezen.

Op bovenstaande wijze kunnen we alle adressen bereiken, maar willen we b.v. naar een hoog adres dan is het een heel werk om steeds maar de plus-toets in te drukken.

Het monitor programma gaat ons nu handig helpen.

Het volgende voorbeeld laat U zien hoe U een willekeurig adres kunt oproepen en zijn inhoud wijzigen.

toets in	uitlezing	commentaar
<input type="text" value="ADDR"/>	X.X.X.X. X X	de 4 punten in het adresveld geven aan dat U een adres kunt intoetsen
<input type="text" value="1800"/>	1.8.0.0. X X	voer adres 1800 in door de toetsen <input type="text" value="1"/> <input type="text" value="8"/> enz. op volgorde in te drukken
<input type="text" value="DATA"/>	1 8 0 0 X.X.	door indrukken DATA toets gaan de punten vanuit het adresveld naar het dataveld en geven daardoor aan dat U nu de inhoud van dit adres kunt wijzigen.
<input type="text" value="3"/> <input type="text" value="A"/>	1 8 0 0 3.A.	voer de data 3A in op adres 1800
<input type="text" value="B"/>	1 8 0 0 A.B.	voer nog een B in. Op het adres 1800 staan nu de data AB. De 3 is nu uit de adresplaats verdwenen. Data groter dan 2 karakters worden niet onthouden.

[+]	1 8 0 1 X.X.	door de [+] toets wordt het volgende adres uit het geheugen opgehaald en uitgelezen. De twee punten in het dataveld geven aan dat U nieuwe data kunt invoeren.
[C][D]	1 8 0 1 C.D.	voer de data CD in op geheugenadres 1801

Op deze wijze kunnen we al de geheugen plaatsen tussen 1800 en 1FFF vullen en wijzigen.

Proberen we op dezelfde manier de inhoud van adres 0000 te veranderen dan lukt dat niet. Probeer maar.

toets in	uitlezing	commentaar
[ADDR][0]	0.0.0.0. 0 6	inhoudadres 0000 is 06
[DATA]	0 0 0 0 0.6.	twee punten in dataveld geven aan dat U de inhoud hiervan kunt veranderen
[5]	DONKER	zolang U nu een willekeurige data toets indrukt laat de uitlezing niets zien. Laat u de toets los dan komt de oude uitlezing weer terug
	0 0 0 0 0.6.	dit laatste betekent dat U in het geheugendeel waar het monitorprogramma staat niets kunt veranderen.

Samenvatting :

De afkorting ADDR staat voor adres. Na het indrukken van deze toets komt de uitlezing in zijn standaard formaat. Dit houdt in : de linkse 4 cijfers geven het adres aan en de 2 rechtse cijfers geven de data aan welke op de desbetreffende geheugenplaats staat. Het adresveld is geïndexeerd met 4 punten.

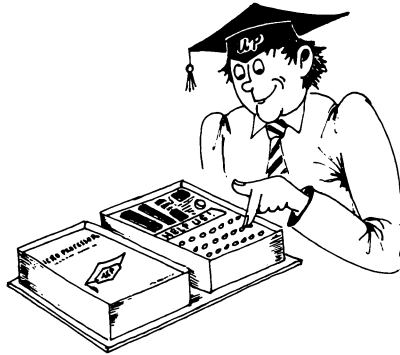
Worden meer dan 4 cijfers ingevoerd in het adresveld dan worden uitsluitend de laatste 4 geaccepteerd. Voert U minder dan 4 cijfers in dan is het uitgelezen adres het werkelijke adres.

Wordt de datatoets ingedrukt dan worden de indexpunten vanuit het adresveld naar het dataveld geschoven. Hiermee wordt aangegeven dat U nieuwe data kunt invoeren.

De inhoud van het aangegeven RAM geheugenadres kan nu vervangen worden door de door U gewenste data. Het gebruik van de +/- toetsen verhoogt of verlaagt het adres in het adresveld. Staan de indexpunten al in het dataveld dan hoeft U de data toets niet meer in te drukken alvorens U de inhoud van het desbetreffende adres wilt veranderen. Na het gebruik van de adres toets mag U direkt de +/- toets gebruiken.

Als U als gebruiker de inhoud van de ROM wilt veranderen dan dooft de uitlezing zolang U een data toets ingedrukt houdt. Na het loslaten van de toets komt de uitlezing normaal terug. Misschien ten overvloede wordt hier opgemerkt dat in het ROM geheugen het monitor programma staat dat vast is ingebracht en niet meer is te veranderen.

3.1.2. Uw eerste programma



Met voorgaande kennis gewapend kunnen we al een klein programmaatje intoetsen en laten werken.

In dit programma willen we op de uitlezing de tekst HELP US laten verschijnen.

toets in	uitlezing	kommentaar
RS	U P F - - 1	systeem in ruststand
ADDR	X.X.X.X. X X	voer startadres in
1 8 0 0	1.8.0.0. X X	
DATA	1 8 0 0 X.X.	voer data in
D D	1 8 0 0 D.D.	DD ingevoerd
+	1 8 0 1 X.X.	volgend adres
2 1	1 8 0 1 2.1.	21 ingevoerd
+	1 8 0 2 X.X.	volgend adres
2 0	1 8 0 2 2.0.	20 ingevoerd
+	1 8 0 3 X.X.	volgend adres
1 8	1 8 0 3 1.8.	18 ingevoerd

Wat we tot nu toe voor dit programma hebben ingevoerd wordt in het algemeen veel korter geschreven als :

1800 DATA DD + 21 + 20 + 18 +

U moet dit als volgt lezen : data DD komt op adres 1800; data 21 komt op adres 1801; data 20 komt op adres 1802 en data 18 komt op adres 1803 te staan. Het gehele hoofdprogramma ziet er verkort geschreven als volgt uit :

RS	ADDR	
1800	DATA DD + 21 + 20 + 18 +	deze regel was al reeds ingevoerd dus ga verder op adres 1804
1804	CD + FE + 05 +	
1807	FE + 13 +	
1809	20 + F9 +	
180B	76 + +	
180C t/m 181F		doet er niet toe. Door herhaald de <input type="checkbox"/> in te drukken is geheugenplaats 1820 te bereiken.
1820	AE +	
1821	B5 +	
1822	1F +	
1823	85 +	
1824	8F +	
1825	37	

Het programma is nu ingevoerd. Nu moeten we het programma nog laten werken. We doen dit als volgt :
toets in uitlezing commentaar

ADDR 1.8.2.5. 3 7 de 4 punten geven aan dat een adres gekozen kan worden

8 0 0 1.8.0.0. D D start adres programma
 GO H E L P U S uitlezing van de tekst

Hoe stoppen we nu dit programma ?

Wel, druk op **STEP** en de uitlezing wordt donker.

Hoe kunnen we het ingebrachte programma nu weer laten lopen?

Druk op **RS** daarna op **ADDR** en weer op **GO** . Het programma loopt nu weer.

Bij het invoeren van dit programma is U misschien iets vreemds opgevallen i.v.b. de telling van de geheugenadressen. De telling verloopt niet zoals U dat bij normaal tellen gewend bent nl van 1800 1801 1802 1803 1804 1805 1806 1807 1808 1809 1810 1811 1812 enz. Maar de telling verliep als volgt 1800 1801 en verder normaal tot 1809 maar dan is de volgende geheugenplaats geen 1810 maar 180A en vervolgens 180B 180C 180D 180E 180F 1820 1821 enz. Wees gerust er is niets met U en met Uw MICRO-PROFESSOR aan de hand maar de adressen en de data gegevens worden in de hexadecimale code aangegeven. Hoe dit werkt wordt U nog uitgelegd.

3.1.3. Programmeren van Uw eigen naam

Maar voordat we dat gaan doen gaan we het programma zo veranderen dat het inplaats van HELP US Uw eigen VOORNAAM kan uitlezen.

Let op Uw naam of voornaam mag niet meer dan 6 letters bevatten.

Hiervoor moeten we de inhoud van de geheugenplaatsen 1820 t/m 1825 aanpassen aan Uw persoonlijke wens.

Stel dat Uw naam PIET is. Dit zijn minder dan 6 letters. We willen op de opengebleven plaatsen streepjes. De uitlezing komt er nu zo uit te zien.

- P I E T -

De data voor de geheugenplaatsen 1820 t/M 1825 wordt dan

1820	02	;	-
1821	87	;	t ..
1822	8F	;	E
1823	30	;	I
1824	1F	;	P
1825	02	;	-

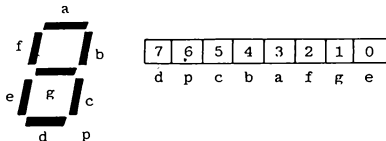
Breng de bovenstaande veranderingen aan in de gegeven geheugenplaatsen en laat het programma weer lopen.

Maak nu Uw eigen naam met behulp van de onderstaande karakertabel in figuur A (blz. 25).

Merk op dat in deze tabel de uitgelezen karakters niet zo fraai gevormd zijn. De oorzaak hiervan is de eenvoud van de toegepaste uitlezing.

Is het programmeren van Uw eigen naam gelukt dan kunt U verder gaan met de bestudering van dit boek zo niet dan moet U het voorgaande nog eens goed doornemen totdat dit goed gaat.

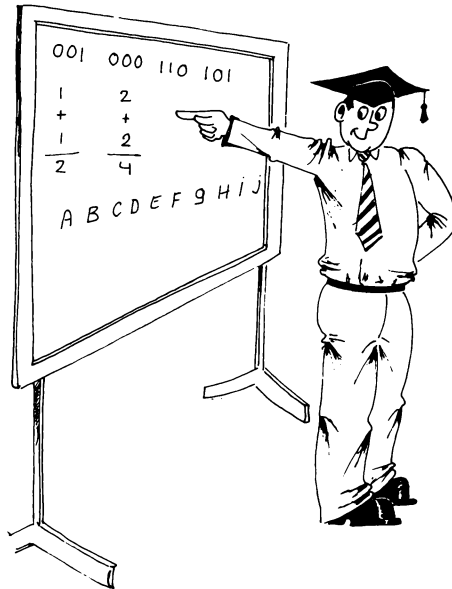
Figuur A : Display format, positiecode en interne code.



DISPLAY FORMAT :

COED	BD 30 9B BA 36 AE AF 38 BF BE 3F A7 8D B3
DATA	0 1 2 3 4 5 6 7 8 9 A B C D
DISP	0 1 2 3 4 5 6 7 8 9 A b c d
CODE	8F 0F AD 37 89 B1 97 85 2B 23 A3 1F 3E 03
DATA	E F C H I J K L M N O P Q R
DISP	E F G H I J K L ñ ñ o P q r
CODE	A6 87 B5 B7 A9 07 B6 8A 83 A2 32 02 C0 00
DATA	S T u v w X Y Z () + - .
DISP	S t U u ū t y ÷ c d 4 - ..

3.1.4. Tellen en talstelsels



In het normale leven tellen we 10-talig met de cijfers 0,1,2,3,4,5,6,7,8,9. Met deze cijfers kunnen we getallen vormen b.v. 713, 317, 173, 371. Voor het vormen van de vier getallen zijn hier telkens dezelfde drie cijfers gebruikt.

De waarde van deze vier getallen is echter niet aan elkaar gelijk. Hun gewicht is elke keer anders. Het getal vindt zijn gewicht dan ook door de cijfers en de plaats die deze in het getal innemen. We noemen dit dan ook een gewogen talstelsel of een positie stelsel.

Als er 713 staat wat betekent dit dan?

Wel er zijn in dit getal :

$$\begin{array}{rcll} 7 \text{ honderden} & = & 7 \times 100 & = 7 \times 10 \times 10 = 7 \cdot 10^2 \\ 1 \text{ tiental} & = & 1 \times 10 & = 1 \times 10 = 1 \cdot 10^1 \\ 3 \text{ eenheden} & = & 3 \times 1 & = 3 \times 1 = 3 \cdot 10^0 \\ + \frac{\quad}{713} & + & \frac{\quad}{713} & + \frac{\quad}{713} & + \frac{\quad}{713} \end{array}$$

We zeggen dan ook 7 maal 10 tot de macht 2, plus 1 maal 10 tot de macht 1, plus 3 maal 10 tot de macht 0 is gelijk aan 713.

Het getal 317 bevat :

$$\begin{array}{rcll} 3 \text{ maal honderd} & = & 3 \times 100 & = 3 \times 10^2 \\ 1 \text{ maal tien} & = & 1 \times 10 & = 1 \times 10^1 \\ 7 \text{ maal één} & = & 7 \times 1 & = 7 \times 10^0 \end{array}$$

Uw MICRO-PROFESSOR en alle andere microprocessoren werken niet met het grondtal 10, maar met het grondtal 2 en de bijhorende karakterset bestaat uit 1 en 0.

Het is ook een gewogen of positiestelsel.

Zo betekent het getal 11001_2 het volgende :

$$\begin{array}{rcll} 11001_2 : 1 \times 2^0 & = & 1 \times 1 & = 1 \\ & 0 \times 2^1 & = & 0 \times 2 = 0 \\ & 0 \times 2^2 & = & 0 \times 2 \times 2 = 0 \\ & 1 \times 2^3 & = & 1 \times 2 \times 2 \times 2 = 8 \\ & 1 \times 2^4 & = & 1 \times 2 \times 2 \times 2 \times 2 = \underline{16} \\ & & & 25_{10} \end{array}$$

Het tweetje achter het getal geeft aan dat het getal gelezen moet worden in het grondtal twee.

Wat is nu het gewicht van dit getal op de basis van het grondtal 10?

Dat is eenvoudig uit te rekenen. Bereken daarvoor elk gewicht en tel het geheel op. De som is dan 25.

Dus : 11001_2 is evenveel als 25_{10}

zo is : $101101_2 = 45_{10}$

en : $1001_2 = 9_{10}$

Reken dit zelf maar eens na.

Gewogen stelsels met andere grondtallen zijn natuurlijk ook mogelijk.

Zie volgend voorbeeld voor het grondgetal 4 met de karakterset 0;1;2;3.

$$\begin{aligned} 2310_4 &: 0 \times 4^0 = 0 \times 1 &&= 0 \\ &1 \times 4^1 = 1 \times 4 &&= 4 \\ &3 \times 4^2 = 3 \times 4 \times 4 &&= 48 \\ &2 \times 4^3 = 2 \times 4 \times 4 \times 4 &&= \underline{128} \\ &&&180_{10} \end{aligned}$$

Bereken zelf eens :

$$123_4 = 27_{10}$$

$$1021_4 = 73_{10}$$

Grondtallen voor zo'n gewogen svsteeem kunnen ook groter zijn dan 10. Zo wordt in de computerwereld heel dikwijls het grondtal 16 gebruikt. De reden voor het grondtal 16 is de volgende : alle getallen welke data of adressen voorstellen worden in het tweetallige stelsel verwerkt in de processor. In dit tweetallig stelsel alles invoeren is een vreselijke karwei en door het gebruik van het 16-tallige stelsel wordt dit werk veel eenvoudiger met veel minder kans op fouten. U zult nu zeggen "Maar waarom dan niet het grondtal 10 gekozen inplaats van dat moeilijke grondtal 16?".

Hierop is het antwoord zeer snel te geven. De omzetting van het tweetallige stelsel naar het 16-tallige stelsel is zeer eenvoudig te realiseren in de electronica en gaat veel eenvoudiger dan van grondtal 2 naar 10 of omgekeerd van 10 naar 2. We zullen dit nog laten zien aan de hand van wat voorbeelden.

Voor het grondtal 16 hebben we 16 karakters nodig en deze zijn : 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F.

De letters A,B,C,D,E en F hebben hier dus respectievelijk de tiental-gewichten 10,11,12,13,14 en 15.

Om aan te geven dat een getal in het 16-tallig stelsel wordt weergegeven, worden in de literatuur de volgende notaties gebruikt :

15_{16} of 15H (de H van Hexadecimaal)

Wat is nu het tientallige gewicht van een getal in het 16-tallige stelsel genoteerd?

Voorbeeld : $2A_{16} = ?$

$$\begin{aligned} A \times 16^0 &= 10 \times 1 = 10 \\ 2 \times 16^1 &= 2 \times 16 = \underline{32} \\ &42_{10} \end{aligned}$$

$BC_{16} = ?$

$$\begin{aligned} C \times 16^0 &= 12 \times 1 = 12 \\ B \times 16^1 &= 11 \times 16 = \underline{176} \\ &188_{10} \end{aligned}$$

Probeer zelf eens het gewicht op het grondtal 10 te becijferen van de volgende zestientallig genoteerde getallen :

$$3B_{16} = 59_{10}$$

$$1FFF_{16} = 8191_{10}$$

$$1800_{16} = 6144_{10}$$

De laatste twee getallen zijn niet willekeurig gekozen. Ze komen overeen met het laatste en eerste adres van de RAM geheugenplaatsen. Het verschil van deze twee waarden bepalen dan ook het aantal te gebruiken geheugenplaatsen. Het verschil is 2047. Om het juiste aantal te bepalen moet het eerste adres er ook bijgeteld worden. Dit brengt het totale aantal te gebruiken geheugenplaatsen op $2047+1 = 2048$. Dit is 2×1024 . Het getal 1024 wordt in de microprocessorwereld 1K plaatsen genoemd.

In Uw MICRO-PROFESSOR is dus standaard meegeleverd een RAM geheugen met een inhoud van 2K.

Zoals al is gezegd wordt de 16-tallige notatie gebruikt om een adres eenvoudig aan te geven. De data welke in het dataveld moet worden ingevoerd wordt dan ook 16-tallig genoteerd en wel telkens met twee karakters. Kijk hiervoor eventueel nog maar eens terug naar het programma wat Uw eigen naam kon uitlezen. Zo'n data blok bestaat altijd uit twee karakters en deze twee karakters worden op één geheugenplaats opgeslagen. Zo'n data blok wordt in de MICRO-PROFESSOR als een tweetallig getal opgeslagen.

Hoeveel plaatsen zijn daar nu voor nodig in een geheugenplaats? Daartoe dienen we het getal in het 16-tallige stelsel om te zetten naar het tweetallige stelsel. Hiervoor gebruiken we de onderstaande omzettingstabel.

10-tallig	16-tallig	2-tallig
00	00	0000
01	01	0001
02	02	0010
03	03	0011
04	04	0100
05	05	0101
06	06	0110
07	07	0111
08	08	1000
09	09	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Het getal $2A_{16}$ laat zich als volgt tweetallig noteren :

Zoek in de tabel onder 16-tallig de 2 op. In het tweetallig stelsel komt dit overeen met 0010. Zoek evenzo de A op, hiervoor staat tweetallig 1010.

$$2A_{16} = \underbrace{0010}_2 \underbrace{1010}_A$$

Dit is te controleren door $2A_{16}$ om te zetten naar het grondtal 10 en $0010\ 1010_2$ om te zetten naar het grondtal 10. Beide antwoorden geven dan 42_{10} .

Let op, deze truc werkt alleen tussen het grondtal 16 en het grondtal 2 en niet tussen de getallen van het grondgetal 10 naar 16 en 10 naar 2 en omgekeerd.

nogmaals de truc.

$$3B_{16} = \underbrace{0011}_3 \underbrace{1011}_B_2$$

Reken maar na.

Het werkt ook voor grotere getallen op het grondtal 16.

$$1FFF_{16} = \underbrace{0001}_1 \underbrace{1111}_F \underbrace{1111}_F \underbrace{1111}_F_2$$

Dit laatste is ook na te rekenen al vergt het wel wat tijd.

We dienen hier op te merken dat een geheugenplaats door 16 enen en nullen op een rijtje in de MICRO-PROFESSOR wordt verwerkt, en dat één geheugenplaats gevuld wordt met 8 enen en nullen. Zo'n groep van 8 enen en nullen wordt een woord of byte genoemd.

$$\begin{array}{ccc} & & \text{bit} \\ & & \downarrow \\ \underline{2A}_{16} & = & \underline{00101010} \\ \text{byte} & & \text{één byte} \end{array}$$

Nu is ook duidelijk geworden wat wil zeggen het RAM geheugen is 2K bytes groot.

Elk element uit een byte wordt een "bit" genoemd. Het woord "bit" is een samentrekking van de eerste letters van de woorden binair informatie teken.

In de microprocessorwereld wordt nooit gesproken over een tientallig stelsel maar van een decimaal stelsel. Zo ook voor het zestientallig stelsel. Dit noemt men hexadecimaal stelsel. Voor het tweetailig stelsel gebruikt men de naam binair stelsel.

Deze namen zijn in het voorgaande betoog niet gebruikt voor de eenvoud maar U moet wel aan de wat lastigere namen wennen zodat ze U in het vervolg bekend in de oren klinken.

Tot slot wordt opgemerkt dat de wiskundige argumentatie voor de overzetting van een hexadecimale notatie naar de binaire notatie volgens de gegeven "truc" buiten het bestek van deze handleiding valt.

Genoeg wiskunde gedaan. Terug naar de werkelijkheid.

3.1.5. Uw tweede programma : laat HELP US knippen

Met het tweede voorbeeld willen we het woord HELP US laten knippen.

RS	ADDR	
1800	212618	
1803	E5	
1804	DD212018	
1808	DDE3	
180A	0632	
180C	CD2406	
180F	10FB	
1811	18F5	
1820	AE	S
1821	B5	U
1822	1F	P
1823	85	L
1824	8F	E
1825	37	H
1826	00	
1827	00	
1828	00	
1829	00	
182A	00	
182B	00	

ADDR 1 8 0 0 GO

Om het lopende programma te stoppen moet U RS indrukken. Het opnieuw starten via :

ADDR 1 8 0 0 GO

De snelheid waarmee het woordje HELP US knippert is te variëren door de inhoud van adres 180B te veranderen.

Nog een verandering in het programma is de volgende :

Zet op de adressen 1826 tot en met 182B de informatie -teip- $\hat{=}$ "-piet-" en laat het gewijzigde programma weer lopen. De uitlezing knippert nu met de tekst PIET HELP US.

Natuurlijk kunt U ook andere teksten met behulp van figuur A invoeren welke dan knipperend worden uitgelezen.

3.1.6. Registers



In de CPU zelf zijn ook nog enkele geheugenplaatsen opgenomen. Deze geheugens worden "registers" genoemd.

Deze registers kunnen net als adressen van het RAM geheugendeel onderzocht worden op hun inhoud en eventueel van nieuwe data worden voorzien.

De toetsen **REG** en **DATA** worden hiervoor gebruikt met het voorschrift :

REG <registernaam> [**DATA** [<data>]] + [<data>] + . . .

Vergelijk dit invoervoorschrift met dat wat in hoofdstuk 3.1.1. vooraan vermeld wordt.

Voorbeeld:

Kontroleer de inhoud van de registers SP, HL, IY
en zet in register A;12 en in register F;34.

toets in	uitlezing	kommentaar
REG	R E G -	zet monitorprogramma in registerstand

S A	<u>X X X X</u> <u>S P</u>	
	register- register-	
	inhoud naam	de inhoud en de naams- afkorting van het register worden uitge- lezen

H₃L	X X X X H L
--	-------------

I₉Y	X X X X I Y
--	-------------

A₀F	X X X X A F
--	-------------

DATA	X X X.X. A F	de twee indexpunten in het dataveld geven aan dat het F register klaar staat om data op te nemen
--	--------------	--

3 4	X X 3.4. A F	register F geladen met data 3 4
---	--------------	------------------------------------

+	X.X.3 4 A F	de indexpunten zijn naar het dataveld van register A geschoven
---	-------------	--

1 2	1.2.3.4 A F	register A is gevuld met de data 1 2
---	-------------	---

Samenvatting :

<Register naam>staat voor de naam van een register. Elk register wordt geadresseerd door slechts één toets. Als is ingedrukt geeft de uitlezing rEG- aan, dit om aan te geven dat een register-naam moet worden ingevoerd.

Na het indrukken van een registernaam verschijnt op de twee rechtse plaatsen van de uitlezing de afkorting van het desbetreffende register, in de vier linkse plaatsen wordt de inhoud van het register zichtbaar gemaakt.

Bijvoorbeeld geeft als uitlezing 1 F 9 F S P . Dit betekent : de stackpoint heeft een inhoud van 1 F 9 F .

Als U als gebruiker verschillende registers wilt onderzoeken dan is het intoetsen van <registernaam> voldoende, natuurlijk na de instructie REG .

Om de inhoud van een register te veranderen moet de worden ingedrukt gevolgd door of . De inhoud van een register is een byte lang, dus twee hexadecimale karakters breed. Nadat al of niet gevolgd door en of is ingetoetst verschijnen er in het dataveld twee indexpunten om aan te geven welk deel van het register klaar staat om data in te voeren. Door het indrukken van de en of toets verschuiven de indexpunten zoals onderstaande figuur aangeeft.

	Naam	Uitlezing
<input type="text" value="+"/>	Vlaggen	X X X.X. A F
↓	Reg A	X.X.X X A F
	Reg C	X X X.X. B C
↑	Reg B	X.X.X X B C
<input type="text" value="-"/>		

Let hier speciaal op de beweging van de indexpunten.

In Uw MICRO-PROFESSOR komen nog enkele speciale register afkortingen voor.

1° AF', BC', DE', HL', het teken ' wordt in de uitlezing aangegeven door een decimaal punt rechts in het adresveld.

A F' X X X X A F.

2° register IX en IY worden aangegeven door respectievelijk I_t en I_Y.

3° Register I en interruptvector IFF 2 worden uitgelezen als IF.

3.1.7. Programmateller (Program Counter) : PC-toets.

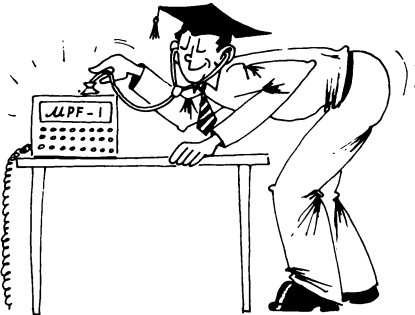
Door middel van toets PC kan de waarde van de programmateller zichtbaar worden gemaakt.

Als het systeem is ge-"reset" dan zet het monitor-programma de programmateller op het laagste RAM adres (1800).

Wordt PC ingetoetst nadat de RS toets is gebruikt dan laat de uitlezing het laagste RAM adres zien.

toets in	uitlezing	kommentaar
<input type="checkbox"/> RS	U P F - - 1	terug in het monitor-programma
<input type="checkbox"/> PC	1 8 0 0 X X	het laagste RAM adres

3.2. Programma onderzoek.



3.2.1. Starten van een programma : GO-toets.

Deze toets is alleen te gebruiken als de uitlezing een adres toont met bepaalde data.

Na het indrukken van de toets springt de CPU naar het adres dat op de uitlezing stond.

Voordat het programma gaat lopen worden al de registers welke U als gebruiker hebt geladen, veilig opgeborgen.

Gebruikers registers kunnen vooraf worden geladen (zie hoofdstuk 3.1.6.).

Voorbeeld : starten gebruikersprogramma

toets in	uitlezing	commentaar
<input type="text" value="REG"/>	R E G -	
<input type="text" value="SF"/>	X X X X S P	de vier linkse karakters zijn geen adres.
<input type="text" value="GO"/>	DONKER	uitlezing is weg : dit is een foutmelding
	X X X X S P	na loslaten van toets <input type="text" value="GO"/> komt de uitlezing weer normaal terug
<input type="text" value="PC"/>	1 8 0 0 X X	het adresveld is nu gevuld met een adres
<input type="text" value="GO"/>	X X X X X X	CPU start met het uit- voeren van het inge- brachte programma

3.2.2. Stap voor stap instructie : STEP-toets

De STEP toets is vergelijkbaar met de GO toets en alleen te gebruiken als de uitlezing een adres toont (de adres-data vorm).

Elke STEP actie heeft tot gevolg dat de CPU één instructie uit het programma uitvoert, dit in overeenstemming met de stand van de programma teller.

Na uitvoering neemt het monitorprogramma de controle weer over en laat op de uitlezing de nieuwe stand van de PC zien met zijn programma inhoud (PC = programma counter).

U als gebruiker mag de inhoud van de registers en de geheugens onderzoeken en wijzigen naar behoefte na elke stap.

Voorbeeld :

We zullen een programma in het RAM geheugen ingebracht, laten lopen door middel van de STEP toets.

toets in uitlezing		kommentaar
RS	U P F - - 1	stelsysteem reset
PC	1 8 0 0 X.X.	begin adresprogramma
3 E	1 8 0 0 3.E.	vul register A met 0
+ 0	1 8 0 1 0.0	
+ 3 C	1 8 0 2 3.C.	verhoog A met 1
+ 4 7	1 8 0 3 4.7.	vul reg B met inhoud van reg A
PC	1 8 0 0 3.E.	terug naar beginadres, uitlezing is in de adresvorm.
STEP	1 8 0 2 3.C.	eerste stap uitgevoerd, PC staat nu op 1802
REG A F	0 0 X X A F	het A register is 0
PC STEP	1 8 0 3 4.7.	volgende stap, PC staat nu op 1803
REG A F	0 1 X X A F	register A is nu 1 geworden.
PC STEP	1 8 0 4 X.X.	volgende stap, PC staat nu op 1804

Theoretische achtergrond van de step instructie :

Wanneer het programma doorgelopen wordt op de stap voor stap manier dan gebruikt het monitorprogramma de gebruikersstack om het retouradres op te slaan. De gebruikersstack is een intern geheugen bankje waar tijdelijk programma informatie wordt opgeslagen.

De opgeslagen adressen in dit geheugenbankje moeten wijzen naar de RAM geheugenplaatsen. Is dit niet goed dan laat de uitlezing ERR - SP zien.

Als de adressen in dit geheugenbankje naar de geheugenadressen van het systeem zelf wijzen dan laat de uitlezing SYS - SP zien.

Als de retourinstructie RET is afgewerkt in het programma dan kan dit geheugenbankje overlopen en geeft dan ook een foutmelding. In bovenstaande gevallen moet de stackpointer (dit is de wijzer in het kleine geheugenbankje) worden veranderd of de RS toets wordt ingedrukt.

Na reset wordt de gebruikers stackpointer naar de foutwaarde gezet door het monitor programma. U als gebruiker hoeft zich geen zorgen te maken over de stackpointer van het systeem zelf.

3.2.3. Het plaatsen van een breekpunt in het gebruikers programma : SBR-toets.



Als een programma erg lang is en er zit een fout in het programma dan kan het punt waar het fout gaat opgezocht worden met de stap voor stap methode maar dit vraagt zeer veel tijd. Handiger is het dan om het programmagedeelte dat goed is snel te doorlopen om te laten stoppen voor de plaats waar de vermoedelijke fout zit en dan stap voor stap verder te zoeken met de STEP toets.

We plaatsen dan een breekpunt (Breakpoint) in het programma. Dit heeft het gewenste effect. Onder controle van het monitorprogramma loopt het programma door tot het gewenste stop adres.

Als het programma op het breekpunt is aangekomen dan mag U als gebruiker de register- en geheugenplaatsen onderzoeken op hun inhoud en deze eventueel wijzigen.

SBR betekent : plaats breekpunt. Als de uitlezing in de adresvorm staat en het adres wijst naar een adres in het RAM geheugen dan wordt door het indrukken van de SBR toets het breekpunt geplaatst.

Voorbeeld van het plaatsen van een breekpunt :

toets in	.uitlezing	kommentaar
<input type="text" value="RS"/>	U P F - - 1	reset
<input type="text" value="ADDR"/> <input type="text" value="1"/> <input type="text" value="8"/> <input type="text" value="0"/> <input type="text" value="0"/>	1.8.0.0. X X	plaats van het start- adres
<input type="text" value="DATA"/> <input type="text" value="3"/> <input type="text" value="E"/>	1 8 0 0 3.E.	laad reg A met 0
<input type="text" value="+"/> <input type="text" value="0"/>	1 8 0 1 0.0.	
<input type="text" value="+"/> <input type="text" value="3"/> <input type="text" value="C"/>	1 8 0 2 3.C.	verhoog reg A met 1
<input type="text" value="+"/> <input type="text" value="4"/> <input type="text" value="7"/>	1 8 0 3 4.7.	laad reg B met in- houd reg A
<input type="text" value="+"/> <input type="text" value="0"/> <input type="text" value="4"/>	1 8 0 4 0.4.	verhoog reg B met 1
<input type="text" value="+"/> <input type="text" value="4"/> <input type="text" value="8"/>	1 8 0 5 4.8.	laad reg C met in- houd van reg B
<input type="text" value="+"/> <input type="text" value="F"/> <input type="text" value="B"/>	1 8 0 6 F.B.	mogelijkheid tot interrupt, de inter- rupt flip flop is gezet
<input type="text" value="SBR"/>	1.8.0.6 F.B.	plaats breekpunt op adres 1806
<input type="text" value="PC"/>	1 8 0 0 3.E.	startadres van pro- gramma

GO 1 8 0 7 X.X. het programma is uit-
gevoerd en is ge-
stopt op 1807. Dit
is het adres van de
volgende instructie

REG R E G - controle van de
registers

A₀F 0 1 X X A F inhoud reg A is 1 en
is dus juist

B₁C 0 2 0 2 B C inhoud reg B en C is
2 en dat is ook juist

I_{-B}IF 0 0 0 1 I F de interrupt flip
flop is gezet. Dit
is het resultaat van
de instructie EI

We veranderen nu de instructie EI in DI op de geheugenplaats 1806. DI betekent reset interrupt flip flop. De betekenis van deze instructies komen aan de orde in het "Experiment manual".

PC - 1.8.0.6. F.B. veranderen EI in DI
(F.B. in F.3.)

F 3 1.8.0.6. F.3.

ADDR 1 8 0 0 1.8.0.0. 3 E plaats startadres
van het programma

GO 1 8 0 7 X.X. laat programma lopen

REG I_{-B}IF 0 0 0 0 I F controleer de inter-
rupt flip flop

Laat dit programma nog even staan en zie wat het effect op het programma is als we het breekpunt weer uit het programma hebben verwijderd (zie hiervoor hoofdstuk 3.2.4).

Samenvatting van de eigenschappen van het monitor-
programma tijdens het gebruik van SBR.

- 1° Het is verboden een breekpunt te plaatsen in het ROM geheugendeel. Doet U dit toch dan dooft de uitlezing.
 - 2° Als een instructie langer is dan één byte dan moet het breekpunt geplaatst worden bij het eerste byte. Doet U dit niet dan komt er een foutmelding.
 - 3° Als de uitlezing in de adres-datavorm staat en het adresveld is het breekpunt dan staan de zes indexpunten in de uitlezing aan om te laten zien dat dit adres het breekadres is.
 - 4° De inhoud van een breekadres kan veranderd worden na het gebruik van de adrestoets.
 - 5° Als het gebruikersprogramma op het breekpunt staat dan is de uitlezing in de adres-data vorm. In het adresveld staat de waarde van de gebruikers programma teller.
 - 6° Als het gebruikersprogramma op zijn breekpunt staat zijn al de toestanden en registers beschermd tegen verlies.
- (7° Zie hoofdstuk 3.2.2 voor de stackregels.)

3.2.4. Verwijder breekpunt in het programma :
CBR-toets.

Als in het programma al of niet een fout is gevonden en verbeterd dan moet het breekpunt weer uit het programma verwijderd worden.

Dit gaat eenvoudig door de toets CBR (clear breakpoint) te gebruiken op het adres waar het breekpunt is ingesteld.

Op elk tijdstip mag U de toets CBR gebruiken en de uitlezing laat dan F.F.F.F.F. zien.

Het breekpunt is dan op adres FFFF gezet.

Verwijder nu het breekpunt van het programma dat in hoofdstuk 3.2.3 ingetoetst is en laat dan het programma weer lopen. Wat ziet U nu? Juist, het programma loopt nu door naar een onbekend adres.

3.2.5 Onmiddellijk stoppen : MONI-toets.

Als in een programma veel fouten zitten of het programma is in een eindeloze lus gekomen dan kan niets het lopende programma stoppen. De voedingsspanning wegnemen is dan het laatste redmiddel, maar dan zijn we het ingebrachte programma ook kwijt. De toets MONI geeft ons nu de mogelijkheid om zo'n programma te stoppen. Het indrukken van de toets MONI werkt volgens het zelfde mechanisme als dat van de STEP toets en brengt het programma weer onder monitorcontrole.

De inhoud en het adres van de gebruikersprogramma teller worden dan uitgelezen. Als bijvoorbeeld in het gebruikersprogramma de halt instructie is uitgevoerd dan licht de LED "Halt" op naast de uitlezing. De uitlezing zelf is dan donker. Door nu de toets MONI in te drukken komt het gebruikersprogramma weer onder controle van het monitorprogramma. De uitlezing laat nu het adres en de inhoud van de programma teller zien op één adresplaats verder dan de halt instructie stond.

Nadat de toets MONI is gebruikt, controleert het monitorprogramma de gebruikers stackpointer.

De gebruiksregels zijn hier hetzelfde als bij de toetsen stap voor stap en breekpunt.

Voorbeeld : gebruik van de instructie "Halt" en terug naar het monitorprogramma.

Verander het programma van het voorbeeld gegeven in hoofdstuk 3.2.3 als volgt :

	toets in uitlezing	commentaar
<input type="checkbox"/>	PC 1 8 0 0 3.E.	zet programmateller aan begin van het programma
<input type="checkbox"/>	+ 1 8 0 1 0.0.	
<input type="checkbox"/>	+ 1 8 0 2 3.C.	
<input type="checkbox"/>	+ 1 8 0 3 4.7.	
<input type="checkbox"/>	+ 1 8 0 4 0.4.	
<input type="checkbox"/>	+ 1 8 0 5 4.0.	
<input type="checkbox"/>	+ 1 8 0 6 F.b.	
<input type="checkbox"/>	+ 1 8 0 7 7.6.	instructie halt op adres 1807 geplaatst.
<input type="checkbox"/>	PC 1 8 0 0 3.E.	programma aan begin gezet
<input type="checkbox"/>	GO D O N K E R	<input checked="" type="checkbox"/> rode LED aan
<input type="checkbox"/>	MONI 1 8 0 8 X.X.	de uitlezing laat nu het eerstvolgende adres zien met zijn inhoud.

Voorbeeld :

Als U de MONI-toets gebruikt als het monitorprogramma loopt dan geeft de uitlezing SYS - SP

toets in uitlezing

kommentaar

RS

U P F - - 1

het monitorprogramma scant het toetsenbord.

MONI

S Y S - S P

de gebruikers stack pointer staat in de systeem stack

Na deze droge stof zullen we eens een programma inbrengen waarbij de kleine luidspreker in Uw MICRO-PROFESSOR op een leuke manier wordt gebruikt. We gaan een politie sirene nabootsen.

Toets daarvoor het onderstaande programma in.

toets in	uitlezing	commentaar
RS	U P F - - 1	

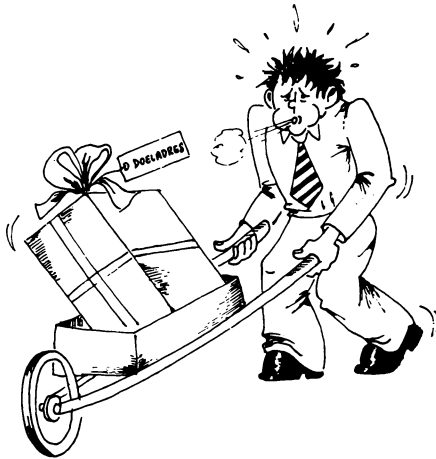
ADDR	1	8	0	0	
DATA	O	E			1 8 0 0 O.E.
+	O				1 8 0 1 O.O.
+	2	1			1 8 0 2 2.1.
+	C	0			1 8 0 3 C.O.
+	O				1 8 0 4 O.O.
+	C	D			1 8 0 5 C.D.
+	E	4			1 8 0 6 E.4.
+	O	5			1 8 0 7 O.5.
+	O	E			1 8 0 8 O.E.
+	C	O			1 8 0 9 C.O.
+	2	1			1 8 0 A 2.1.
+	O				1 8 0 B O.O.
+	O	1			1 8 0 C O.1.
+	C	D			1 8 0 D C.D.
+	E	4			1 8 0 E E.4.
+	O	5			1 8 0 F O.5.
+	1	8			1 8 1 0 1.8.
+	E	E			1 8 1 1 E.E.
PC					1 8 0 0 O.E.
GO					D O N K E R (X) groene LED aan

De luidspreker laat nu een sirene horen. Ik hoop dat dit geluid U niet al te bekend in de oren klinkt.

3.3. Hulpfuncties.

In de inleiding is gezegd dat in het monitorprogramma een aantal handige hulpfuncties zijn opgenomen. In dit deel komen de uitvoeringen van de hulpfuncties aan de orde.

3.3.1. Blok verplaatsing : MOVE-toets.



De instructie ziet er als volgt uit :

`MOVE` <adres> `+` <adres> `+` <adres> `GO`

Voorbeeld :

Verschuif het datablok 1800-18FF naar 1810-190F.
toets in uitlezing commentaar

MOVE

 X.X.X.X. - S S is de afkorting van
startadres

1	8	0	0
---	---	---	---

 1.8.0.0. - S startadres 1800

+

 X.X.X.X. - E E is de afkorting van
eindadres

1	8	F	F
---	---	---	---

 1.8.F.F. - E eindadres 18FF

+

 X.X.X.X. - d D is de afkorting van het
doeladres

1	8	1	0
---	---	---	---

 1.8.1.0. - d doeladres is 1810

GO

 1 8 1 0 X.X. de verschuiving is klaar.
Het laatste verschoven
byte is dat van adres
1810.

Zijn nu inderdaad de data van het geheugenplaatsen
verschoven? Dat kunt U controleren.

Plaats daartoe in de geheugenplaatsen 1800 tot
1820 controleerbare data.

Neem bijvoorbeeld :

ADRES INHOUD

1800 00

1801 01

1802 02

1803 03

.

.

.

.

.

181C 1C

181D 1D

181E 1E

181F 1F

1820 20

1821 00

1822 00

1823 00

.

.

.

.

181F 00

1830 00

toets in	uitlezing	kommentaar
MOVE 1 8 0 0	1.8.0.0. - S	startadres 1800
+ 1 8 2 0	1.8.2.0. - E	eindadres 1820
+ 1 8 1 0	1.8.1.0. - D	doeladres is 1810
GO	1 8 1 0 0.0.	verschuiving is klaar

Onderzoek nu eens de inhoud van de geheugen-
plaatsen 1810 tot en met 1820.

U zult zien dat op de geheugenplaatsen 1810 tot en
met 1830 de data 00 tot en met 20 staat, die eerst
op de adressen 1800 tot en met 1820 waren inge-
toetst. Kijken we nog even naar wat er op de
geheugenplaatsen 1800 tot en met 180F staat na de
blokverschuiving, dan zien we dat de ingetoetste
data 00 tot en met 0F daar nog staat. We hebben
hiervoor het datablok naar boven geschoven (naar
hogere adressen). Naar beneden verschuiven kan ook.

Vul daartoe de adressen 1800 tot en met 180F
allemaal met de inhoud 0.

MOVE X.X.X.X. - S

1810 1.8.1.0. - S startadres 1810

+ X.X.X.X. - E eindadres

1830 1.8.3.0. - E eindadres 1830

+ X.X.X.X. - D

1800 1.8.0.0. - D doeladres 1800

GO 1 8 2 0 2.0. verschuiving is klaar

Kontroleer nu nogmaals de inhoud van de adressen
1800 tot en met 1820. Als alles goed gegaan is
staat nu op die adressen dat wat U met de hand aan
het begin hebt ingetoetst. U hebt dus het blok
eerst naar boven (naar hogere adressen) geschoven
en bij de tweede verschuiving is het blok weer
naar de oude plaats geschoven.

Samenvatting :

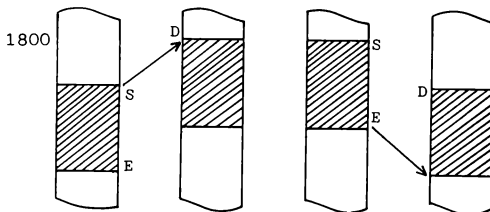
Na het indrukken van de MOVE-toets geeft de uitlezing X.X.X.X.- S.

De S slaat op start adres van de data die verschoven moet worden. Na het indrukken van de $\left[\begin{array}{c} - \\ + \end{array} \right]$ geeft de uitlezing X.X.X.X.- E.

De E slaat op het eind adres van het data blok dat verschoven moet worden.

Na nogmaals $\left[\begin{array}{c} - \\ + \end{array} \right]$ in te drukken geeft de uitlezing X.X.X.X.- D.

De D slaat op het eerste doel adres van de nieuwe datablok plaats. Na het indrukken van de GO-toets wordt de data verschoven en als het verschuiven voltooid is dan komt de uitlezing in de adres-data vorm terug met het adres van het laatste doel adres. Het schuiven van blokken kan naar boven en naar beneden. Bij verschuiven naar boven is het laatst verschoven adres het laagste adres en bij verschuiven naar beneden is het laatst verschoven adres het hoogste adres, zie fig. 3.3.



verschuiving naar boven verschuiving naar beneden

Fig. 3.3 beeld van move functie

Uw MICRO-PROFESSOR werkt hier zeer snel zodat na het indrukken van de GO-toets onmiddellijk de uitlezing het laatst verplaatste adres aangeeft.

Als het doelgebied, het gebied van het monitorprogramma overlapt, geeft de MICRO-PROFESSOR een foutmelding. U dient dan de RS-toets te gebruiken om terug te komen in het monitorprogramma.

3.3.2. Data deletion : DEL-toets.

Deletion betekent verwijderen.

De regel luidt : <data>

Deze toets is te gebruiken als de uitlezing in de adres-data vorm staat. Door het indrukken van deze toets wordt de data op het gewenste adres uit het programma verwijderd en alle hogere adres inhouden worden naar één adresplaats lager verschoven.

Voorbeeld:

Plaats in de adressen 1800 tot en met 1809 de data 00, 11, 22 tot en met 99.

Willen we nu op adres 1803 de data verwijderen dan gaat dit als volgt :

toets in	uitlezing	kommentaar
----------	-----------	------------

<input type="text" value="ADDR"/> <input type="text" value="1"/> <input type="text" value="8"/> <input type="text" value="0"/> <input type="text" value="3"/>	1 8 0 3	3.3
---	---------	-----

<input type="text" value="DEL"/>	1 8 0 3	4.4. verwijdering uitgevoerd en alle hogere plaatsen zijn één plaats opgeschoven.
----------------------------------	---------	---

Onderzoek dit maar door de adressen 1800 tot en met 1809 door te lopen.

<input type="text" value="ADDR"/> <input type="text" value="1"/> <input type="text" value="8"/> <input type="text" value="0"/> <input type="text" value="0"/>	1 8 0 0	0.0. beginadres
---	---------	-----------------

<input type="text" value="+"/>	1 8 0 1	1.1
--------------------------------	---------	-----

<input type="text" value="+"/>	1 8 0 2	2.2.
--------------------------------	---------	------

<input type="text" value="+"/>	1 8 0 3	4.4. data van oud adres 1804
--------------------------------	---------	------------------------------

<input type="text" value="+"/>	1 8 0 4	5.5. data van oud adres 1805
--------------------------------	---------	------------------------------

<input type="text" value="+"/>	1 8 0 5	6.6. data van oud adres 1806
--------------------------------	---------	------------------------------

enzovoort

Samenvatting :

De data in het ROM geheugen kan niet verwijderd worden, hier staat het monitorprogramma.

Het gebied waar data verwijderd kan worden ligt in het adresgebied van 1800 tot en met 1DFF. Wanneer het te verwijderen adres in het toegestane gebied ligt dan schuift de data op de achter het te verwijderen adres een plaats op en de laatste plaats (1DFF) wordt met een 0 gevuld.

3.3.3. Data insertion : INS-toets.

Insertion betekent tussenvoegen.

De regel luidt : INS <data>

Dit is een bijzonder prettige hulpfunctie bij het programmeren. Bent U bijvoorbeeld een instructie vergeten bij het programmeren of hebt U een twee byte's lange instructie als één byte's instructie in het programma opgenomen dan moet er een instructie of instructiedeel tussengevoegd worden. Hier is natuurlijk een adresplaats voor nodig. Alles wat boven dat tussengevoegde adres staat moet dus een plaats worden opgeschoven.

De INS toets is te gebruiken als de uitlezing in de adres-data vorm staat. De tussen te voegen data komen direkt na het adres dat op het adresveld uitgelezen wordt.

Voorbeeld :

Zet in het RAM geheugen de volgende data :

Adres	Data
1800	00
1801	11
1802	22
1803	44
1804	55
1805	66

Stel nu dat we op adres 1803 de data 33 willen
tussenvoegen dan doen we dit als volgt :

toets in	uitlezing	commentaar
ADDR 1 8 0 2	1.8.0.2. 2 2	adres tot waar het programma moet blijven staan
INS	1 8 0 3 0.0.	het adres waar data tussengeschoven moet worden is leeg ge- maakt
3 3	1 8 0 3 3.3.	data invoeren

Controle

ADDR 1 8 0 0	1.8.0.0. 0 0	startadres
+	1 8 0 1 1.1.	
+	1 8 0 2 2.2.	
+	1 8 0 3 3.3.	
+	1 8 0 4 4.4.	
+	1 8 0 5 5.5.	

U ziet dat het tussengevoegde adres 1803 nu gevuld
staat met 33 (was 44). De inhoud van adres 1803 is
naar adres 1804 gebracht, enzovoort.

Samenvatting :

Het werkgebied van deze hulpfunctie is gelijk aan dat van de delete (verwijder) hulpfunctie. Het laatste byte in het werkgebied valt weg.

Het tussen te voegen byte wordt één adres verder geplaatst dan het uitgelezen adres.

Door het indrukken van de INS-toets wordt de data één geheugenplaats opgeschoven. Dit gaat zeer snel zodat ogenblikkelijk na het loslaten van de INS-toets het in te voegen adres verschijnt en U als gebruiker mag elke willekeurige data direkt invoeren.

3.3.4. Berekening van een relatief adres :
RELA toets.

De springinstructie JR en DJNZ (deze instructie komen aan de orde in het "Experiment Manual") gebruiken relatieve adressen. Het berekenen met de hand van deze relatieve adressen is geen eenvoudige klus. Voor het gemak is in het monitorprogramma een hulpfunctie opgenomen waarmee een relatief adres eenvoudig kan worden berekend. Het gebruiksrecept van de RELA - toets is als volgt :

<adres> <adres>

Voorbeeld :

Neem aan dat U in een programma de springinstructie JR of DJNZ opneemt op het adres 1810 en dat U wilt springen naar adres 1819. Wat is nu het relatief adres?

In dit eenvoudige voorbeeld is het relatief adres ook wel zo uit te rekenen volgens : doeladres - startadres is de relatieve adres afstand.

Aangezien we achter adres 1810 het relatief adres moeten opgeven (op adres 1811) en de sprong vóór adres 1819 moet inspringen moeten we van de relatieve adres afstand nog twee aftrekken om aan de juiste relatieve afstand te komen. Dus het relatief adres om van 1810 naar 1819 te springen is :

$$(1819-1810) - 2 = 9 - 2 = 7$$

Kijken of dit klopt met dat wat Uw MICRO-PROFESSOR er van maakt.

toets in	uitlezing	commentaar
<input type="checkbox"/>	RELA	X.X.X.X. - S monitorprogramma is klaar voor opname van startadres
<input type="checkbox"/>	1 8 1 0	1.8.1.0. - S startadres is opgegeven
<input type="checkbox"/>	+	X.X.X.X. - D monitorprogramma is klaar voor opname van het doeladres
<input type="checkbox"/>	1 8 1 9	1.8.1.9. - D doeladres is opgegeven
<input type="checkbox"/>	GO	1 8 1 1 0.7. het monitorprogramma heeft het relatieve adres berekend en geplaatst op adres 1811

Laten we nu maar meteen onderzoeken wat zo'n springinstructie doet in een programma. Maak daarvoor de adressen 1800 tot 1825 schoon door er allemaal nullen in te zetten.

Zet nu de programmateller op 1810 en vul dit adres met 18 als zijnde de data voor de instructie JR=Jump Relatief (spring relatief) en 1811 met 07 op het relatief adres. Vul nu nog de adressen 1818 met AA, 1819 met BB en 181A met CC en laat de overige gewoon leeg.

Onderzoek nu het programma met de step voor step
functie. Weet U het nog?

toets in	uitlezing	kommentaar
<input type="checkbox"/> RS	U P F - - 1	monitorprogramma.
<input type="checkbox"/> PC	1 8 0 0 0.0.	begin van het programma
<input type="checkbox"/> STEP	1 8 0 1 0.0.	stap door het geheugen- adres tot 1810
.	.	.
.	.	.
.	.	.
.	.	.
<input type="checkbox"/> STEP	1 8 1 0 1.8.	springinstructie
<input type="checkbox"/> STEP	1 8 1 9 B.B.	de programmateller heeft de sprong uitgevoerd en de uitlezing laat de in- houd van dit adres zien

Om terug te komen op regel 1810 moet er herhaalde
malen op de toets gedrukt worden. Doe dat maar
en stop op adres 1811.

<input type="checkbox"/> -	1 8 1 8 A.A.	
.	.	.
.	.	.
.	.	.
<input type="checkbox"/> -	1 8 1 1 0.7.	adres 1811 met relatief adres voor JR instructie

Als U nu op de STEP-toets zou drukken wat zou er dan gebeuren?

Kunt U dit voorspellen?

Ja juist, adres 1812 komt voor te staan en wordt uitgelezen.

STEP 1 8 1 2 0.0.

druk nu tweemaal op de toets.

 1 8 1 1 0.7.

 1 8 1 0 1.8.

Wat zal er gebeuren als U de STEP-toets indrukt?

Juist, nu wordt er wel gesprongen naar regel 1819.

Als we op adres 1819 de inhoud BB vervangen door 1 8 : de sprong instructie JR met het relatief adres F4; dan hebben we zelf een klein sprong programmaatje gemaakt.

Onderzoek de werking van dit programma met de STEP toets. Noteer daartoe de uitlezingen welke na elke step actie worden uitgelezen. Als U begrijpt wat hier gebeurt kunt U dan ook voorspellen wat er gebeurt als U op de GO-toets zou drukken?

Doe dat dan maar.

Opmerking : het relatief adres op adres 181A is F4. Hiermee springt het programma naar regel 180F. Probeer U eens dit relatief adres met potlood en papier te berekenen. Dat is zeker niet gemakkelijk. Met de MICRO-PROFESSOR gaat dat eenvoudig dat hebt U kunnen zien.

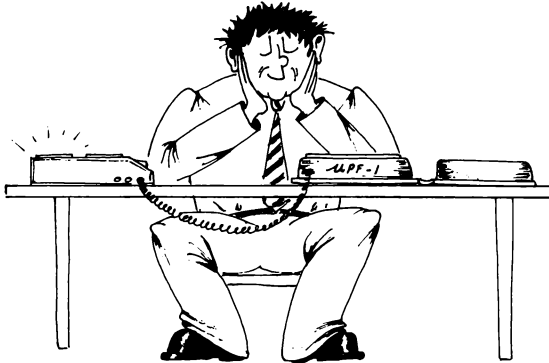
Samenvatting :

Na het indrukken van de toets RELA laat de uitlezing het adres zien waarop de programmateller stond. De S geeft aan dat dit het start adres is voor een springinstructie. Na het indrukken van de + vraagt de MICRO-PROFESSOR via de uitlezing om het doel- of bestemmingsadres.

Als dit adres is opgegeven en men drukt de GO toets in dan berekent de MICRO-PROFESSOR het relatief adres en zet dit op de plaats van de tweede byte van de operatie code. De uitlezing komt in de standaard vorm terug en het dataveld laat het relatieve adres zien.

Als de relatieve adresafstanden meer dan +127 of -128 (decimaal) zijn dan komt de uitlezing met een foutmelding -ERR terug.

3.3.5. Opslag van programma's op cassettebandjes :
TAPE WR - toets



Een cassettebandje is een geheugen medium dat een grote opslagcapaciteit heeft en niet vluchtig is. Uw MICRO-PROFESSOR bevat zowel de benodigde electronica als hulpprogrammatuur om door U zelf ontwikkelde programma's naar een cassettebandje te schrijven.

De gebruiksregel luidt :

<blad nummer> <adres> <adres>

Voorbeeld :

Sla de data op van adres 1800 tot 18FF op een cassettebandje. Om het terugzoeken op het bandje te vereenvoudigen als er meerdere programma's opstaan wordt er een codenummer vooraf mee gezonden dit is het bladnummer (File name). Laat dit bladnummer hier 1234 zijn.

toets in	uitlezing	kommentaar
<input type="checkbox"/> TAPE WR	X.X.X.X. - F	F is de afkorting van bladnummer

<input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4	1.2.3.4. - F	bladnummer is 1234
---	--------------	--------------------

<input type="checkbox"/> +	X.X.X.X. - S	S is de afkorting van startadres
----------------------------	--------------	----------------------------------

<input type="checkbox"/> 1 <input type="checkbox"/> 8 <input type="checkbox"/> 0 <input type="checkbox"/> 0	1.8.0.0. - S	startadres is 1800
---	--------------	--------------------

<input type="checkbox"/> +	X.X.X.X. - E	E is de afkorting van programma eind adres
----------------------------	--------------	--

<input type="checkbox"/> 1 <input type="checkbox"/> 8 <input type="checkbox"/> F <input type="checkbox"/> F	1.8.F.F. - E	eindadres is 18FF
---	--------------	-------------------

play en rec	verbind de microfoon-ingang van de cassette-recorder met de MIC-plug van de MICRO-PROFESSOR. Zet de recorder op opname en start de recorder.
-------------	--

<input type="checkbox"/> GO	D O N K E R	start uitvoer van data naar cassetterecorder. Tijdens de uitvoer is de uitlezing donker maar de LED tone out is aan.
-----------------------------	-------------	--

1 8 F F X.X.	De overdracht is klaar en het eindadres wordt uitgelezen.
--------------	---

Samenvatting :

Door het indrukken van de TAPE WR - toets komt de uitlezing in de vorm van X.X.X.X.- F.

De F is de afkorting van File name (bladnummer). De X.X.X.X. gaat het bladnummer vormen, wat het terugzoeken op een bandje met meerdere bladnummers mogelijk maakt.

Na het indrukken van de toets laat de uitlezing X.X.X.X.- S zien waarmee het monitorprogramma vraagt om het startadres waarmee het op te slaan programma mee begint. De S slaat op de afkorting startadres. Dit adres moet nu opgegeven worden. Door het nogmaals indrukken van de toets laat de uitlezing X.X.X.X.- E zien om aan te geven dat het eindadres van het programma wordt gevraagd. Voor het indrukken van de GO - toets moet de cassetterecorder via de MIC plug op de MICRO-PROFESSOR aangesloten worden op de microfoon ingang van de recorder en tevens moet de recorder gestart worden in de stand opname. Als de recorder niet loopt of niet op de stand opname staat en U drukt de GO - toets in dan wordt de data wel uitgezonden maar niet opgenomen waardoor U een stuk van de op te nemen data mist op Uw bandje. Tijdens het overbrengen van de data op het bandje is de uitlezing donker maar de groene LED is aan en de luidspreker in Uw MICRO-PROFESSOR laat een hoog pieptoontje horen.

Als de overdracht klaar is laat de uitlezing het eindadres zien met zijn inhoud. De groene LED is nu uitgegaan en de luidspreker is stil geworden:

3.3.6. Inlezen data van de cassetteband :
TAPE RD-toets

De gebruiksregel luidt : <bladnummer>

Voorbeeld :

Lees de data op het bandje met het bladnummer 1234 terug in Uw MICRO-PROFESSOR.

toets in uitlezing kommentaar

X.X.X.X. - F F is de afkorting van het
bladnummer

1.2.3.4. - F bladnummer opgegeven.

Verbindt de uitgang van de cassetterecorder met de EAR ingang van de MICRO-PROFESSOR. Gebruik de oortelefoon uitgang (EAR).

start uitvoering lees opdracht.

Start de cassetterecorder op de stand weergave met het hoogste uitgangsniveau.

Uw MICRO-PROFESSOR toont nu de langskomende bladnummers. Elk bladnummer wordt ongeveer 1,5 seconde op de uitlezing uitgelezen. Wanneer het gewenste bladnummer gevonden is dan gaat de uitlezing over in.

bladnummer - - - - - als het inlezen klaar is dan wordt het laatste adres met zijn inhoud op de uitlezing gezet.

Samenvatting :

Voordat U het inlezen start moet de oortelefoon-uitgang van de cassetterecorder verbonden worden met de ingang (EAR) van Uw MICRO-PROFESSOR. Zet de volumeregelaar van de recorder op maximaal. Druk nu de GO - toets op Uw MICRO-PROFESSOR in en start daarna de cassetterecorder in de stand weergave. Tijdens het zoeken laat de uitlezing zes punten zien totdat het gewenste bladnummer is gevonden. Is dit bladnummer gevonden dan geeft de uitlezing zes streepjes aan - - - - - . Start en eind adres staan al op de band zodat deze niet opnieuw opgegeven moeten worden.

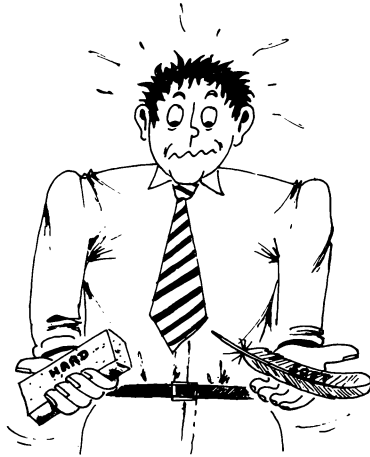
U als gebruiker hoeft alleen het bladnummer (File name) op te geven. Tijdens het opnemen is een controle woord mee opgenomen. Bij het uitlezen wordt dit codewoord vergeleken met het codewoord gevormd tijdens het inlezen. Zijn deze twee codewoorden niet aan elkaar gelijk dan is er iets fout gegaan bij de opname of weergave. De uitlezing geeft dan - ERR aan.

Zijn de twee codewoorden aan elkaar gelijk dan zijn er geen fouten ontstaan bij de opname en weergave operatie's en dan geeft de uitlezing het laatste adres met zijn inhoud weer in de standaard adres-data vorm.

Als de uitgelezen data op geheugenplaatsen worden uitgelezen, die tot het systeem geheugen behoren, dan wordt een foutmelding gegeven.

De ingelezen data wordt tijdens het inlezen door de luidspreker in Uw MICRO-PROFESSOR hoorbaar gemaakt, zodat heel eenvoudig is te controleren of de band leeg of met data beschreven is. Dit is praktisch bij het schrijven op de cassetteband. Door eerst te lezen van de cassetteband middels tape read kunt U nu horen tot waar er al programma's op de band staan zodat overschrijving op de band voorkomen kan worden.

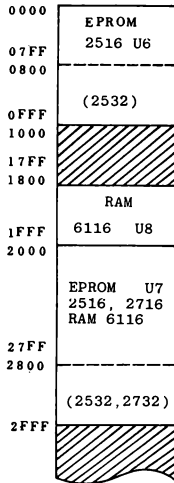
4. SOFTWARE EN HARDWARE BESCHRIJVING.



Opmerking :

"Software" is een programmatuur en "Hardware" is de electronica in zijn uitvoeringsvorm.

4.1. Opbouw geheugen adressen.

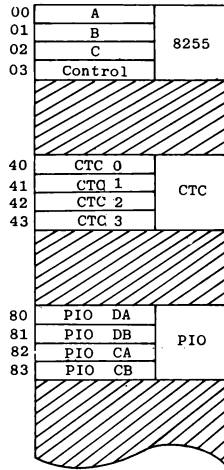


Figuur 4-1 : Plattegrond geheugenplaatsen.

Beschrijving geheugenplaatsen :

- 1° U6 is de Eprom met het monitorprogramma.
U7 is gereserveerd voor uitbreiding van het bestaande systeem met een RAM of een EPROM.
U8 is het RAM geheugen dat standaard in Uw MICRO-PROFESSOR is opgenomen. De geheugenplaatsen 1F9F tot en met 1FF3 worden door het monitorprogramma gebruikt. Let hier op bij het programmeren!
- 2° De adressen zijn volledig gedecodeerd in Uw MICRO-PROFESSOR en U hoeft geen enkele verbinding te verbreken of te maken als U op de plaats U7 een IC met het typenummer 2516-2716 of 2532 wenst bij te plaatsen.
- 3° Als U in U7 het IC met typenummer 2732 wenst op te nemen dan moet U de volgende verbindingen verbreken aan de onderzijde van de hoofdprint.
pin 1,2 van jumper 1
pin 3,4 van jumper 1
pin 5,6, van jumper 1
Doorverbonden moeten dan worden :
pin 2,3 van jumper 1
pin 4,5, van jumper 1
pin 6,7 van jumper 1
- 4° Wenst U het IC 6116 op te nemen als U7 dan moet U vooraf het volgende doen :
Verbreek aan de onderzijde van de hoofdprint de verbinding 3,4 van jumper 1 en verbindt pin 4 met 5 van jumper 1

4.2. Ingangs- en uitgangsadressen :
Input/Output adressen (I/O adressen).



Figuur 4-2
I/O adres plattegrond

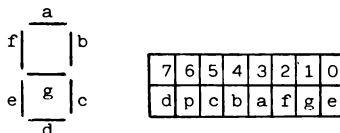
Beschrijving van de Input/Output poorten.

1° IC 8255 is een programmeerbare periferie interface met 24 parallele I/O lijnen. Deze 24 I/O lijnen zijn verdeeld in drie 8 bit brede poorten. (Zie de fabrieksgegevens voor verdere detaillering).

2° Het controlewoord van deze programmeerbare I/O interface is 90.H. Poortdeel A is dan een ingangspoort en de poorten B en C zijn dan uitgangspoorten, elk 8-bit breed.

a) Poort A heeft het adres 00 bit plaats 7 cassette band in bit plaats 6 is verbonden met de toets USER KEY en wordt actief als deze laag wordt gemaakt bit plaatsen 0 tot en met 5 zijn verbonden met de rijen van het toetsenbord. Hetingangssignaal wordt alleen laag als de toets in de aktieve kolom wordt ingedrukt.

b) Poort B heeft het adres 01 en stuurt de zeven segmenten en de decimaal punten in de uitlezing aan. Fig 4-3 geeft de naam van elk segment en het corresponderende uitgangsbite van poort B. Als een uitgangsbite hoog is dan licht het desbetreffende segment op.



Figuur 4-3 : de corresponderende uitgangsbite met hun desbetreffende segmenten.

c) Poort C heeft het adres 02. Bit 7 verzorgt de data uitgang naar de cassetterecorder en parallel hieraan is de luidspreker in Uw MICRO-PROFESSOR aangesloten. Ook wordt de LED tone out hiermee aangestuurd. De LED wordt uitgeschakeld als deze uitgang laag is.

Bit 6 monitor break control :

Onder geen voorwaarden mag dit bit veranderd worden.

De bits 0 tot en met 5 zijn verbonden met de 6 kolommen van de uitlezing en het toetsenbord.

Bit 0 is verbonden met de meest rechtse en bit 5 is verbonden met de meest linkse kolom van de uitlezing.

Deze bits zijn in hun actieve stand hoog.

3° De Z80 counter timer controller (CTC) is een programmeerbaar onderdeel met vier onafhankelijke kanalen die kunnen tellen en/of een tijd-functie kunnen genereren.

De I/O adressen voor deze CTC lopen van 40.H tot en met 43.H.

4° Het Z80 parallel I/O onderdeel is een programmeerbare ingangs/uitgangspoort. Deze PIO heeft twee 8-bits brede ingangs/uitgangs poorten welke op TTL niveau een koppeling kunnen maken tussen Uw MICRO-PROFESSOR en de omgeving.

Het I/O adres van de PIO loopt van 80.H tot en met 83.H.

5° Voor de onderdelen, besproken in dit hoofdstuk, zijn de adreslijnen niet volledig gedecodeerd. Alleen de adres lijnen A0, A1, A6 en A7 zijn gebruikt. De adreslijnen A2 tot en met A5 zijn ongebruikt.

4.3. Programma Interrupt (Programma Onderbreking).

In het monitorprogramma wordt de niet gemaskeerde interrupt gebruikt. Het is U als gebruiker niet toegestaan dit te gebruiken. Pin nummer 16 van de CPU is verbonden met de opsteek connector I links op de hoofdprint naar de pin INT. Het monitorprogramma maakt in zijn gebruik speciale verbindingen met het adres 0038. Wanneer het programmadeel volgend op adres 0038 wordt uitgevoerd dan wordt de controle overgedragen via de adressen 1FEE en 1FEF. Tijdens dit proces worden al de toestanden in de CPU niet beïnvloed. De normale waarde van deze twee adressen is 0066 en dit is het ingangspunt van de service routine.

Het programmadeel onder 0038 wordt uitgevoerd in de volgende situaties :

- 1° De 1^e manier : De interrupt wordt beantwoord.
- 2° De instructie RST 38.H is uitgevoerd (operatiecode FF).
- 3° De databus is hoog gezet. Als de manier "0" interrupt is beantwoord zonder het gebruik van de interruptvector. RST 38.H wordt dan uitgevoerd.
- 4° Als er een fout is in het lopende programma en er wordt naar een niet bestaand adres gesprongen dan wordt door de CPU de operatiecode FF uitgevoerd.

Als de inhoud van de adressen 1FEE en 1FEF niet veranderd is na het aansluiten van de voedingsspanning het effect van de uitvoering van 0038 hetzelfde als het indrukken van de monitortoets of het plaatsen van een breekpunt. U als gebruiker mag Uw eigen service programma adresseren door de inhoud van de adressen 1FEE en 1FEF te veranderen.

4.4. De software stop instructie :
RST 30.H (met operatiecode F7).

RST 30.H heeft hetzelfde effect op een lopend programma als een onderbreking. Deze instructie wordt software breekpunt genoemd omdat deze programma regel in het monitorprogramma (30.H) is opgenomen en voor deze onderbreking is geen kontakt aktie nodig.

Dit monitorprogrammadeel wordt gewoonlijk gebruikt om een gebruikersprogramma te beëindigen. Ook is deze RST op 30.H te gebruiken als een veelvoudig breekpunt bij programma onderzoek.

4.5. Het hulp geheugen (de STACK).

In fig. 4-4 is de plattegrond van het hulpgeheugen getekend. Het laagste adres van het gebruikersgedeelte is 1F9F. Elke keer als U Uw gebruikersprogramma onderbreekt dan controleert het monitorprogramma zijn eigen systeem - hulpgeheugen - aanwijzer (systeem stackpointer (SP)), en zal deze uitlezen als de gebruikers stackpointer naar het systeem hulpgeheugen wijst. Als er in het gebruikersprogramma een instructie is opgenomen die overeenkomt met die uit de systeem stack (b.v. RET) dan kan er een foutmelding komen als het gebruikers hulpgeheugen die van het systeem overlapt.

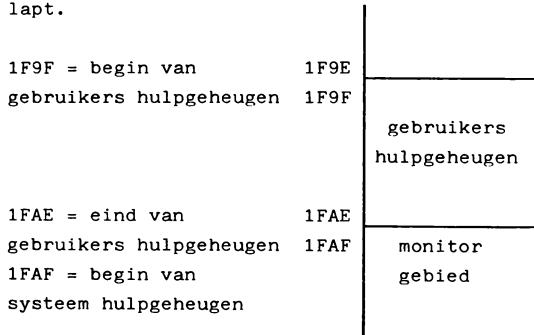


FIG. 4-4 : Ligging van de hulpgeheugens

De uitlezing SYS- SP kan als volgt worden uitgelezen :

- 1° Door het indrukken van de MONI - toets wanneer het monitorprogramma de CPU controleert. Deze opdracht vernietigt de inhoud van de gebruikersregisters en moet dan ook om deze reden zoveel mogelijk vermeden worden.
- 2° Door uitvoering van het monitorprogramma door er met de step functie door te willen lopen.

4.6. De RESET functie.

Op twee manieren is het systeem in zijn beginstand (reset stand) te brengen :

1° Door het aansluiten van de voedingsspanning.

De volgende interne akties vinden dan plaats :

- a) De interrupt flip flop wordt op 0 gezet (IFF set to 0)
- b) Het I register wordt op 0 gezet
- c) De interrupt manier wordt op de 0 manier gezet.
- d) De gebruikers programmateller wordt naar adres 1800 gezet.
- e) De gebruikers stackpointer wordt op adres 1F9F gezet.
- f) Alle breekpunten worden verwijderd.
- g) In adres 1FEE wordt 66 en in adres 1FEF wordt 00 geschreven.

Als nu het monitorprogramma regel 0038 uitgevoerd wordt door de CPU dan gaat deze naar regel 0066.

Deze aktie is gelijkwaardig aan het indrukken van de toets MONI

- h) De uitlezing wordt dan van rechts naar links volgeschoven met de letters MPF-1

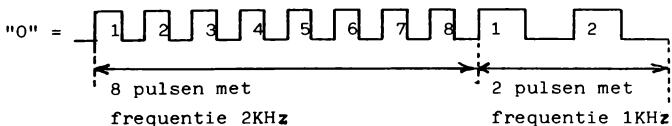
2° Indrukken van de toets RS :

- a) In deze situatie gebeurt bijna hetzelfde als onder punt (1) genoemd. De punten a tot en met e zijn hetzelfde. De akties genoemd onder de punten f en g blijven hier achterwege en de uitlezing komt nu ineens terug met UPF--1.

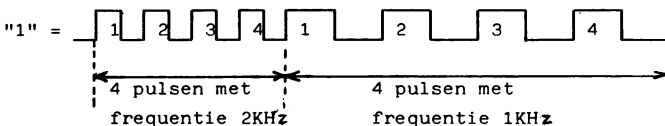
4.7. Opbouw van de data op de cassetteband.

1° Opbouw van een bit op de band :

Voor een "0" wordt er op de band het volgende patroon gezet :



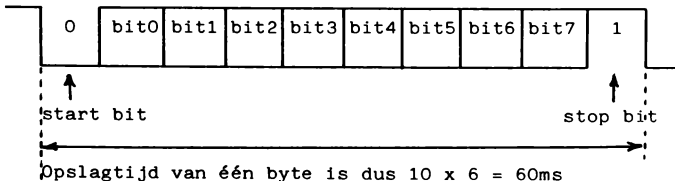
en voor een "1" wordt op de band gezet het signaal



De totale tijdsduur voor een bit is 6ms.

2° De opbouw van een byte met zijn synchronisatie pulsen is als volgt opgebouwd : een start byte dat altijd "0" is gevolgd door de 8 bits. van de data die opgeslagen moeten worden. Deze worden weer gevolgd door een stop bit dat altijd "1" is.

8 data bits = 1 byte



3° Een heel programma op de band gezet ziet er als volgt uit :

begin synch	blad naam	start adres	eind adres	contr getal	midden synch	data	na synch
1KHz	2	2	2	1	2KHz	data	2KHz
4 sec	byte	byte	byte	byte	2 sec		2 sec

5. BESCHRIJVING VAN DE SUBROUTINES VAN HET MONITOR-PROGRAMMA.

5.1. Overzicht van de te gebruiken hulpfuncties.

<u>Adres</u>	<u>Afkorting</u>	<u>Omschrijving functie</u>
0624	scan 1	zoek het toetsenbord af en lees een cyclus uit op de uitlezing
05FE	scan	zoek het toetsenbord af en lees uit totdat een nieuwe toets is ingedrukt
0689	HEX 7	vertaal één hexadecimaal cijfer in de 7 segments uitleeskode
0678	HEX 7SG	vertaal twee hexadecimale cijfers in de 7 segments uitleeskode
05F6	RAM CHK	onderzoek of het gegeven adres in het RAM veld valt
05E4	TONE	wek toon op
05DE	TONE 1K	wek toon van 1KHz op
05E2	TONE 2K	wek toon van 2KHz op

5.2. Beschrijving van de hulpfunctie SCAN 1.

Het startadres van de scan 1 hulpfunctie is 0624.
De functie van deze scan 1 hulpfunctie is het afzoeken van het toetsenbord of er ergens een toets is ingedrukt en het aansturen van de uitlezing. Eén zo'n zoekcyclus wordt doorlopen in 9,97ms en wel van rechts naar links.

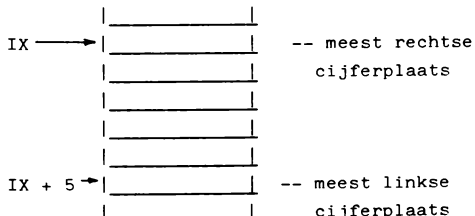
Het IX register wijst naar de uitlezing van het tussengeheugen. De aansturing van deze scan 1 hulpfunctie d.m.v. het toetsenbord geschiedt als volgt :

- 1° Als er geen toets is ingedrukt dan wordt het overdrachts bit op "1" gehouden.
- 2° Als er wel een toets is ingedrukt dan wordt het overdrachts bit op "0" gezet en de positie kode van de desbetreffende toets wordt in register A opgeslagen.

De inhoud van de registers AF; A'F'; B'C' en D'E' worden hierbij overschreven.

Over de uitlezing valt nog op te merken dat :

- 1° Er 6 bytes nodig zijn om de 6 cijferplaatsen in de uitlezing aan te sturen.
- 2° Het IX register wijst naar de uiterste linkse cijferplaats en het register IX + 5 wijst naar de meest rechtse cijferplaats (zie figuur).



- 3° Voor de relatie tussen de gezette bits in een byte en de oplichtende segmenten van een cijferplaats : zie fig.4-3 (blz. 79).

5.3. Beschrijving van de hulpfunctie SCAN

Het adres van deze functie is 05FE.

In functie is deze functie gelijk aan de functie van SCAN 1 met uitzondering van de volgende punten :

- 1° SCAN 1 zoekt eenmaal, maar SCAN zoekt of er een nieuwe toets is ingedrukt.
- 2° SCAN 1 komt na zijn zoekactie terug in de rustpositie terwijl SCAN antwoordt met de interne kode van de ingedrukte toets.

De ingang wijst naar het uitlezingsbuffer.

Via het A register wordt de uitgang verzorgd. Register A bevat de interne kode van de ingedrukte toets.

De registers AF, B, HL, AF', BC' en DE' verliezen bij deze aktie hun inhoud.

5.4. Beschrijving van de hulpfunctie HEX 7

Het adres van deze functie is 0687.

Zijn functie is het decoderen van één hexadecimaal cijfer in zijn bijhorende 7-segmentskode. Zijn ingangsgegevens zijn de laatste 4-bits van een byte welke in register A wordt opgeslagen. Dit register bevat dan ook het hexadecimale nummer (0-F).

Het resultaat van de omzetting wordt weer in het register A opgeslagen.

Bij deze aktie verliest alleen het AF register zijn inhoud.

5.5. Beschrijving van de hulpfunctie HEX7SG

Het startadres van deze functie is 0678.

Deze functie zet twee hexadecimale cijfers om in de 7-segments kode.

Van de benodigde ingangsgegevens wordt het eerste nummer opgeslagen in de rechtse 4 bits van het A register en het tweede cijfer wordt opgeslagen in de linkse 4 bits van datzelfde A register.

Het door deze functie bepaalde uitleespatroon wordt opgeslagen in het HL register. Het eerste patroon wordt opgeslagen in het HL register en het tweede in het HL + 1 register.

Bij deze aktie verliezen de registers AF en HL hun oorspronkelijke waarde.

5.6. Beschrijving van de hulpfunctie RAMCHK

Het adres van deze hulpfunctie is 05F6.

Zijn functie is het controleren van de inhoud van een adres in het RAM geheugenveld.

Zijn ingangsgegeven is het adres dat onderzocht moet worden. Dit adres wordt in het HL register geplaatst. Zijn uitgangswaarde kent twee toestanden, namelijk :

is het adres in het RAM veld dan wordt de nulvlag "1" gemaakt

in het andere geval wordt deze nulvlag op "0" gezet.

Bij deze aktie verliest het AF register zijn inhoud.

5.7. Beschrijving van de hulpfunctie TONE

Het adres van deze functie is 05E4.

De functie wekt een willekeurige toon op die door U zelf bepaald kan worden. Het ingangsgegeven voor deze hulpfunctie is de waarde van het getal C. Deze C waarde bepaald de opgewekte frequentie volgens de formule : $(44 + C \times 13) \times 2 \times 0,5$ microseconde. Daar hoort de frequentie, $200/(10 + 3 \times C)$ bij. Het HL register bevat de waarde van het getal C wat direkt de frequentie bepaald. De maximale waarde van C is 32768. Er is geen uitgangsgrootheid voor deze functie.

De registers AF, B, DE en HL verliezen bij de uitvoering van deze hulpfunctie hun inhoud.

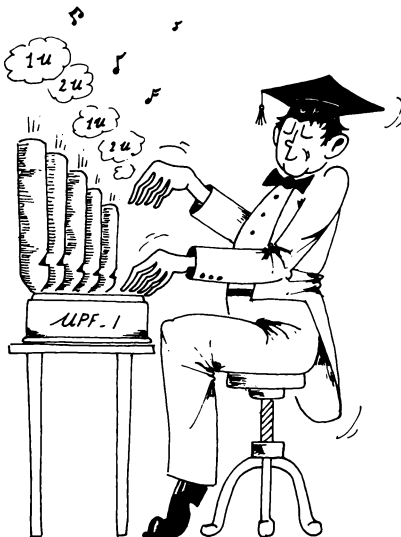
5.8. Beschrijving van de hulpfunctie TONE 1K

Zijn adres is 05DE en zijn functie is het opwekken van een toon van 1KHz.

Zijn benodigde ingangsgegeven worden in het HL register opgeslagen. Ook bij deze functie zijn er geen uitgangsgegevens. Bij de uitvoering van deze hulpfunctie verliezen de registers AF, BC, DE en HL hun oorspronkelijke inhoud.

5.9. Beschrijving van de hulpfunctie TONE 2K

Zijn adres is 05E2. De beschrijving is geheel gelijk aan die van TONE 1K met het enige verschil dat de inhoud van het HL register hoger is. Dit is nodig omdat hier een toon van 2KHz moet worden opgewekt.



OPMERKING :

Met de kennis die U nu bezit moet U al de voorbeeldprogramma's uit het experimentenboek (Experiment Manual) kunnen uitvoeren.

VEEL SUCCES!

Ir. W. B. Koekkoek.

SOFTWARE NOTITIES

SOFTWARE NOTITIES

|

SOFTWARE NOTITIES
