

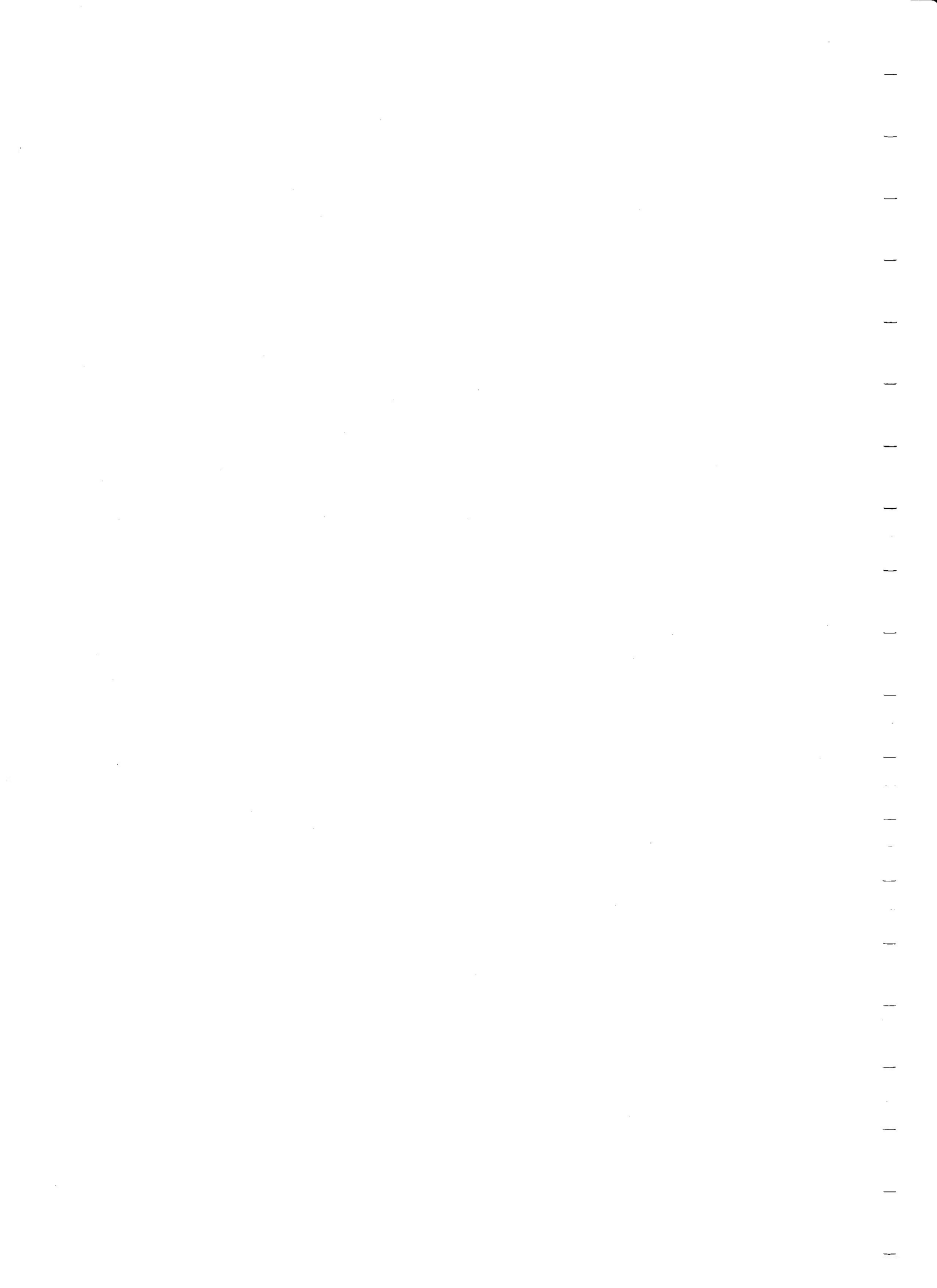


PHILIPS

**MICROPROCESSOR
ZELFSTUDIË-
CURSUS**

VRAAGSTUKKEN
EN OPLOSSINGEN

55



VRAAGSTUKKEN
EN OPLOSSINGEN

© 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 toestemming van N.V. Philips' Gloeilampenfabrieken, Eindhoven.

Vraagstuk 1

Start het MONITOR-programma en instrueer de INSTRUCTOR dat U de registers, het PSU en het PSL wenst te onderzoeken. Wat verschijnt er op de DP?

Vraagstuk 2

Nadat oefening 1 verricht is, laat dan de inhoud van register 4 zien. Deze keuze is foutief, want men wil de inhoud van register 0 zien. Voer de correctie uit!

Vraagstuk 3

Na het kiezen van register 0 noteert men de inhoud hiervan zoals dit op de DP verschijnt. Nu kiest men het volgende register door slechts één toets in te drukken.

- Welke toets is dat?
- Welke registerinhoud wordt nu getoond en van welk register?
- Hoe kan men nu de volgende registerinhouden bepalen evenals het PSU en PSL?

Vraagstuk 4

Kies weer register 0 en maak de inhoud gelijk aan H'00'. Vervolgens kieze men:

register 1 bank 0 en maak de inhoud H'10'
register 2 bank 0 en maak de inhoud H'20'
register 3 bank 0 en maak de inhoud H'30'
register 1 bank 1 en maak de inhoud H'11'
register 2 bank 1 en maak de inhoud H'12'
register 3 bank 1 en maak de inhoud H'13'
maak de inhoud van het PSU H'FF'
maak de inhoud van het PSL H'EE'

Controleer vervolgens de inhoud van deze registers en het PSU en PSL.

Vraagstuk 5

Onderzoek de geheugenplaatsen H'0000' t/m H'001F' en schrijf de inhoud op. Wijzig vervolgens de geheugencellen in hexadecimaal oplopende waarden, d.w.z. cel H'0000' krijgt de waarde H'00' en cel H'0000' de waarde H'01', enz.

Vraagstuk 6

Controleer na het verrichten van oefening 5 de inhoud van de geheugencellen H'0000' t/m H'001F' en voer eventuele correcties uit.

Vraagstuk 7

Kies een willekeurige cel uit de voorgaande oefeningen en controleer deze op de inhoud. Maak vervolgens de inhoud H'00'.

Vraagstuk 8

Als oefening 5, maar nu met snelladen.

Vraagstuk 9

Als oefening 6, maar nu met snel controleren.

Vraagstuk 10

- Hoe wordt de inhoud van het instructie-adresregister (IAR) ook wel genoemd?
- Maak de inhoud van het IAR H'01EF'.

Vraagstuk 11

Controleer na het verrichten van oefening 10 de inhoud van het IAR.

Vraagstuk 12

- Laad de geheugenplaatsen H'0000' en H'0001' met de NOP-instructie en geheugenplaats H'0002' met de HALT-instructie.
- Maak het PSW H'0000'.
- Controleer de waarde van het PSW en de waarden van de geheugenplaatsen H'0000' t/m H'0002'.
- Maak PC=0. Druk nu de STEP-toets verscheiden malen achtereen in en noteer de waarden op de DP. Welke waarden zijn dat?

Vraagstuk 13

- Laad d.m.v. het toetsenbord het register 1 met de waarde H'FF'. Laad register 0 met de waarde H'AA'.
- Maak een programma dat de inhoud van register 1 in register 0 laadt. Zorg voor een HALT-instructie. Onderzoek na afloop de registers 0 en 1. Het beginadres is 0.

Vraagstuk 14

- a) Maak een programma waardoor register 0 wordt geladen met H'00' en register 1 met H'AA'. Vervolgens moet de inhoud van register 1 worden overgedragen aan register 0. Daarna stopt het programma.
- b) N.B. Start het programma door RST in te drukken.
- c) Voer hetzelfde programma nog een keer uit in de STEP-by-STEP mode. Wat toont de DP?
- d) Onderzoek na afloop van het programma, register 0.

Vraagstuk 15

- a) Maak een programma dat register 2 laadt met data van een plaats die 24 bytes verder gelegen is dan het adres van deze instructie. Zorg dat op deze plaats H'EF' aanwezig is.
- b) Controleer na afloop de inhoud van het register 2.
- c) Maak vervolgens een programma dat van deze plaatsen (dus 24, 25 verder) het indirecte adres van de data (H'EF') betreft, die echter op H'100' aanwezig is. Zorg zelf voor de NOP- en HALT-instructies.

Vraagstuk 16

- a) Laad m.b.v. het toetsenbord de waarde H'FF' op geheugenplaats H'00AA'.
- b) Maak nu een programma dat gebruik maakt van een relatieve adressering en deze waarde aldus in register 0 laadt. Start met RST-toets.
- c) Onderzoek het register 0 vóór en na de uitvoering.
- d) Laad de gegeven waarde d.m.v. een instructie (indirect) in register 3. Gebruik als plaats voor de meest significante byte voor de indirecte adressering H'000F'.

Vraagstuk 17

- a) Maak een programma dat op H'0000' begint en vijf waarden in het geheugen verplaatst van de locaties H'0100' t/m H'0104' naar de locaties H'0110' t/m H'0114', en wel zodanig dat de data aanwezig op H'0100', geplaatst wordt op H'0110', die van H'0101' op H'0111', enz.
Het register 0 doet dienst als transportcentrum. Gebruik het register R1 als index-register.
Laad de geheugenplaatsen H'0100' t/m H'0104' met resp. H'0A', H'0B', H'0C', H'0D', H'0E' d.m.v. het toetsenbord.
Initieer het register R1 d.m.v. het toetsenbord.
Gebruik voor dit programma de "Load en Store Absolute" instructies. Start het programma d.m.v. de RUN-toets (N.B. zet eerst PC).
- b) Controleer of inderdaad de gewenste data op de juiste plaatsen staat.
- c) Controleer eveneens na afloop de inhoud van register 0 en 1. Wat zijn de inhoud?

Vraagstuk 18

- a) Maak een programma dat op H'0000' begint en dat vier waarden van een gebied dat "oorsprong" heet, naar een gebied dat "doel" heet overdraagt.
"Oorsprong" komt overeen met H'0100' en "doel" met H'010F'. De waarden die overgedragen moeten worden zijn H'02', H'06', H'05' en H'00'.
Gebruik de "Load en Store Absolute" instructies. Nu dient echter indirecte adressering gebruikt te worden.
Plaats d.m.v. het toetsenbord de adressen (dus H'0100', H'010F') op H'0040' en H'0042'.
- b) Controleer na afloop van het programma of inderdaad alle data op de gewenste plaatsen aanwezig is.

Vraagstuk 19

Voer oefening 17 uit, echter zodanig dat de data van H'0100', H'0101' enz. komt op de plaatsen H'0114', H'0113' enz.
Gebruik daartoe resp. register R1 als "index-register" voor de "Load"-instructies en register R2 voor de "Store"-instructies.

Vraagstuk 20

Voer vraagstuk 18 uit, echter zodanig dat op het gebied "doel" niet "2 6 5 0" komt te staan, maar "0 5 6 2". Gebruik daartoe register R2.

Vraagstuk 21

- a) Maak een programma voor het optellen van de inhoud van de registers R1 en R2 (zie voor de te laden waarden, a resp. b van vraagstuk 21b). Deze worden d.m.v. het toetsenbord (REG) geladen. De som dient in register R1 te komen. Het programma dient op geheugenplaats H'0000' te beginnen en te stoppen d.m.v. de HALT-instructie. De INSTRUCTOR moet daartoe op de juiste wijze geladen worden!
- b) Na het uitvoeren van het programma dient de inhoud van register R1 onderzocht te worden, evenals de PSL-bit 0 (C) en de PSL-bit 2 (OVF):

	R1	R2
a:	64	52
b:	- 64	-52
c:	- 64	52
d:	64	-52
e:	120	52
f:	120	-52
g:	-120	52
h:	-120	-52

N.B. dit zijn decimale waarden, dus eerst omrekenen in hexadecimale waarden!

Vraagstuk 22

- Maak een programma voor het optellen van drie waarden die op opeenvolgende geheugenplaatsen staan, en wel de eerste byte op H'0010', de tweede op H'0011' en de derde op H'0012'. Het programma begint op adres H'0000'.
De waarden zijn decimaal resp. 20, -80 en 60 (omrekenen in hex!). Het resultaat dient op plaats H'0010' geplaatst te worden. Er mag niet van een instructie gebruik gemaakt worden die nog niet behandeld werd.
- Controleer de inhoud van geheugenplaats H'0010' na afloop van het programma en ook PSL-bit 0 (C) en bit 2 (OVF).

Vraagstuk 23

- Laad de waarden H'2 6 5 0' van oefening 18 "stap-voor-stap" in register R0. Tel er telkens een vast getal bij op, en plaats de sommen terug op de oorspronkelijke plaatsen. Controleer dit!
- Voer dit programma uit in de "RUN"-mode.
- Voer dit programma uit in de "STEP"-mode.

Vraagstuk 24

Als vraagstuk 23, maar maak nu gebruik van "indirecte" adressering.

Vraagstuk 25

- In reg. R0 is een bepaalde logische vector aanwezig b.v. '11110101'. Van deze vector moeten de bits 0, 5 en 6 een logische "0" worden; de overige bits blijven ongewijzigd.
Maak een programma dat deze handeling laat verrichten (zie verder laatste zin onder b).
- Laad d.m.v. het toetsenbord register R0 resp. met de volgende waarden: H'A3', H'60', H'33', H'7A'.
Onderzoek telkens de CC en de inhoud van register R0.

Vraagstuk 26

- De logische "OF" wordt vaak gebruikt om "enen" te plaatsen op bepaalde posities in een byte.
Voor I/O-doeleinden worden "enen" gewenst op de bit-plaatsen 7, 5 en 4.
Stel een programma op, gebruik makend van de in vraagstuk 25 gegeven waarden in register R0.
- Onderzoek telkens het register R0 en de CC. Wat valt nu op van de CC t.o.v. van die van het vorige vraagstuk? Verklaar dit!

Vraagstuk 27

- De inhoud van register R0 dient op bepaalde plaatsen gewijzigd te worden. Deze plaatsen zijn de bits 0, 3, 5 en 7. Dit dient met behulp van een "immediate" instructie te geschieden. Maak een programma dat eerst H'F0' laadt in register R0.
- Controleer na afloop de inhoud van register R0 en de Conditie-Code (CC).

Vraagstuk 28

- Laad het register R3 met de waarde H'05'. Laad vervolgens register R0 met H'17'. Verschuif nu de inhoud van register R3 één plaats naar links. Tel vervolgens R0 en R3 bij elkaar op. Verschuif daarna R0 één plaats naar links en trek R3 ervan af.
- Wat is het resultaat? Verklaar dit!
(Controleer eventueel de inhoud van de registers R0 en R3 door het programma d.m.v. de STEP-toets stap-voor-stap uit te voeren).

Vraagstuk 29

- Maak een programma dat in register R0 de waarde H'40' plaatst. Draag deze waarde vervolgens over aan het PSU.
- Na afloop van het programma dient men de lamp FLAG te inspecteren en het PSU m.b.v. het monitor-programma te controleren. Hoe is het PSU en wat is er met de FLAG?

Vraagstuk 30

- Maak een programma dat de FLAG in- of uitschakelt telkens als het programma gestart wordt. Laad daartoe d.m.v. het toetsenbord het register R0 met een bepaalde vector.
- Controleer na afloop het PSU en het register R0.

Vraagstuk 31

- Maak een programma waarin m.b.v. een "immediate" instructie het register R3 van bank 0 wordt geladen met een waarde, b.v. H'33'.
- Vervolgens moet deze data in het register R2 van bank 1 geplaatst worden.
- Controleer na afloop van het programma de inhoud van alle registers die hierbij betrokken zijn geweest.

Vraagstuk 32

- a) Maak een programma dat het register R1 van bank 0 met H'80' laadt. Vervolgens wordt deze "1" over één plaats naar links geschoven in de Carry-bit van het PSL. Deze wordt vervolgens in register R2 van bank 1 geschoven (maak R2 eerst 0).
- b) Controleer na afloop het PSL, register R1 van bank 0 en register R2 van bank 1. Interpreteer het PSL.

Vraagstuk 33

- a) In het geheugen staan twee vectoren, n.l. H'97' en H'34' op de adressen H'0100' resp. H'0101'. Deze vectoren stellen logische waarden voor. Maak een programma, dat deze waarden vergelijkt.
- b) Onderzoek na afloop de CC-bits van het PSL om te zien welke vector "groter" was. Waarom?

Vraagstuk 34

Idem als oefening 33, maar met dien verstande dat de waarden H'97' en H'34' geen vectoren, maar getallen in het 2-complementaire binaire systeem voorstellen (vergelijk dus "aritmetisch").

Vraagstuk 35

- a) Maak een programma voor het onderzoeken van de "SENSE"-ingang. Als het programma gestart wordt, mag de "SENSE"-toets "wel" of "niet" ingedrukt worden gehouden. Na afloop van het programma (RUN-lamp dooft) mag de SENSE-toets eventueel worden losgelaten.
- b) Onderzoek daarna de CC-bits van het PSL.

Vraagstuk 36

- a) Op de adressen H'0101' en H'0102' staan de getallen H'80', H'44' en H'67'. Tel deze getallen op in register R0. Indien de som positief is, beëindig dan het programma. Is de som negatief, dan moet het register R0 met 0 geladen worden, waarna het programma beëindigd moet worden.
- b) Onderzoek daarna de inhoud van het register R0 en de CC-bits van het PSL.

Vraagstuk 37

Idem als oefening 36, maar op het adres H'0100' moet nu de waarde H'01' i.p.v. H'80' komen te staan.

Vraagstuk 38

- a) Er is een tabel die begint op adres H'0100' en in totaal tien waarden bevat, n.l. 1, 2, 3, 4, 5, 6, 7, 8, 9 en 10 (H'0A'). Deze tabel dient d.m.v. het toetsenbord in het geheugen geladen te worden. De tien waarden moeten bij elkaar worden geteld in register R0. In register R1 wordt bijgehouden hoeveel getallen al zijn opgeteld. Maak een programma waarin slechts één optel-instructie voorkomt.
- b) Controleer na uitvoering de inhoud van de registers R0 en R1.

Vraagstuk 39

- a) Reken uit hoeveel tijd bij benadering verloopt in het onderstaande programma tussen het "aan" en "uit" gaan van het FLAG-lampje:

PPSU	H'40'	FLAG "aan"
LODI, R1	H'C8'	Laad R1 met 200
TIJDLUS	BDRR, R1	TIJDLUS Delay-lus
CPSU	H'40'	FLAG "uit"
HALT		
- b) Controleer deze tijd aan de hand van het FLAG-lampje. Laad daartoe het programma in de INSTRUCTOR 50 van adres H'0100' af. Start het programma van dit adres af door het laden van het IAR ("zetten" van de PC).

Vraagstuk 40

Door het toevoegen van vier instructies in het programma van oefening 39 kan een "delay" van ca. 20 seconden verkregen worden. Maak dit nieuwe programma, laad het in de INSTRUCTOR en ga na of de tijd klopt.

Vraagstuk 41

Schrijf een programma dat na het indrukken van de SENSE-toets het FLAG-lampje "aan" zet en dit gedurende 20 seconden laat oplichten. Als men na het doven de SENSE-toets opnieuw indrukt, wordt het "aan" en "uit" gaan herhaald. Zorg ervoor dat bij het indrukken van RST het programma begint, maar dat dit onmiddellijk op adres H'0100' verder gaat, waar het programma van oefening 40 begint.

Vraagstuk 42

- a) Als in oefening 41 de SENSE-toets ingedrukt blijft, dan zal het FLAG-lampje niet uitgaan. Wijzig het programma zodanig dat het FLAG-lampje pas "aan" gaat nadat de SENSE-toets weer wordt losgelaten.

- b) Het kan zijn dat bij het indrukken z.g. “bouncing” (flikkeren) optreedt. Met behulp van een tijdslus van 20 ms kan dit vermeden worden (zogenaamd software debouncing). Voeg zo nodig deze korte tijdslus toe. Gebruik daarvoor de registers van bank 1.

Vraagstuk 43

- a) In het geheugen zijn op de plaatsen H'0100' t/m H'0109' getallen van twee bytes elk aanwezig. Deze getallen zijn decimaal resp. 326, 852, -230, 5 en -302 (dus omrekenen in 2-complementair hex.). Ze dienen d.m.v. het toetsenbord in het geheugen geladen te worden. De laatste byte bevat het minst significante deel van het getal -302. Zet het resultaat in de registers R1 (meest significante deel) en R2 (minst significante deel) van bank 0.
Schrijf een programma (waarin indirecte adressering wordt gebruikt voor deze data, gebruik hiervoor adres H'0040' e.v.) met niet meer dan twee absolute optel-instructies voor deze data. Gebruik register R3 als indexregister en maak gebruik van de instructie BRNR.
- b) Controleer na uitvoering de inhoud van register R0, van het gebruikte index-register R1 en de registers R2 en R3 van bank 0, het PSU en het PSL en de CC.

Vraagstuk 44

Schrijf een programma waarbij, na het indrukken en weer loslaten van de SENSE-toets, het FLAG-lampje elke 5 seconden “aan” en “uit” gaat. Gebruik voor dit probleem slechts één tijdprogramma, dat als subroutine gerealiseerd dient te worden.

Maak gebruik van de instructie RETC om van de subroutine terug te keren.

Vraagstuk 45

Als oefening 44, nu echter met 5 seconden “aan” en 10 seconden “uit”. Er mag slechts één tijdsroutine aanwezig zijn. Realiseer “uit” door indirecte adressering. Het adres op deze plaats mag gewijzigd worden.

Vraagstuk 46

De SENSE-toets wordt een even aantal malen ingedrukt. Na het indrukken wordt dit d.m.v. optellen in register R3 geregistreerd. Is men gereed met het indrukken van deze toets (max. 6 keer), dan kan men de toets “lang” ingedrukt houden. Dit signaleert het einde van het invoeren van impulsen. Nu wordt naar een van vier subroutines gesprongen, een en ander afhankelijk van de inhoud van register R3. Deze sprong dient m.b.v. indirecte adressering te worden uitgevoerd d.m.v. de instructie BSXA. Na het terugkeren

uit de subroutines wordt weer teruggekeerd naar het programma, zodat hetzelfde proces herhaald kan worden.

De subroutines hebben de volgende functies:

R3 = 2. De FLAG-lamp gaat “aan”. Daarna wacht de subroutine op het indrukken van de SENSE-toets, waarna de FLAG-lamp weer dooft en de subroutine beëindigd wordt.

R3 = 4. De FLAG-lamp gaat gedurende 5 seconden “aan”, dan 5 seconden “uit”, dan weer 5 seconden “aan” en vervolgens “uit”. Gebruik voor de tijdmeting een tweede subroutine.

R3 = 6. De FLAG-lamp gaat gedurende 5 seconden “aan” en dan “uit”, waarna de subroutine beëindigd is.

Wordt de SENSE-toets langdurig ingedrukt terwijl de inhoud van register R3 oneven is, dan dienen de beschouwde afgegeven impulsen als ongeldig beschouwd te worden en wordt opnieuw begonnen met het kijken naar SENSE-informatie.

Vraagstuk 47

Bij een INTERRUPT biedt de INSTRUCTOR 50 het adres H'0007' aan. Als indirecte adressering (stand schakelaar) gebruikt wordt, bevindt zich op deze plaats de eerste byte van het effectieve adres.

Schrijf een programma dat start door RST in te drukken en dat de FLAG-lamp elke 5 seconden “aan”, resp. “uit” laat gaan. Wordt de INT(errupt)-toets ingedrukt, dan stopt dit “aan” en “uit” schakelen. De FLAG-lamp blijft gedurende een tijd van 20 seconden in de toestand waarin hij zich op het moment van de interruptie bevond, waarna het oorspronkelijke programma herhaald wordt.

Vraagstuk 48

- a) Schrijf een programma dat data, die vóór het starten van het programma in de data-schakelaars werd geplaatst, uitleest. De waarde van deze data moet in register R2 geplaatst worden. Daarna stopt het programma. Lees de data in de “NON-EXTENDED I/O”-mode in. Zorg ervoor dat bij de start van het programma de CC = 00 is.
- b) Controleer de CC-bits van het PSL en de inhoud van register R3 na afloop van het programma.

Vraagstuk 49

Deze oefening is analoog aan oefening 48. Nu dient de data echter de data in de “MAPPED I/O”-mode ingelezen te worden.

Vraagstuk 50

Deze oefening is analoog aan de voorgaande twee. Nu dient echter in de "EXTENDED I/O"-mode ingelezen te worden, (N.B. Let goed op het adres; raadpleeg de handleiding van de INSTRUCTOR.)

Vraagstuk 51

Laad d.m.v. het toetsenbord de waarden H'02', H'06', H'05' en H'00' in vier opeenvolgende plaatsen.

Tel deze vier waarden bij elkaar op en geef de som weer via de "NON-EXTENDED I/O"-mode op de data-LED's.

Vraagstuk 52

Als oefening 51, maar geef nu de som in de "EXTENDED I/O"-mode weer.

Vraagstuk 53

Als oefening 51, maar geef nu de som in de "MAPPED I/O"-mode weer op de data-LED's. (N.B. let goed op het adres; raadpleeg de handleiding van de INSTRUCTOR.)

Vraagstuk 54

Lees de data van de "NON-EXTENDED I/O". Pas als schakelaar 7 in de 0-stand staat, wordt de data als definitief geaccepteerd. Vervolgens wordt een rotatie naar links gegeven en de data weer in de "NON-EXTENDED I/O"-mode op de data-LED's weergegeven. Daarna stopt het programma.

Vraagstuk 55

- Laad in de "EXTENDED I/O"-mode, via de schakelaars, een decimaal getal (35) in BCD-code in register R0. Vervolgens idem het getal 47 in BCD-code in register R1 laden.
Tel deze 2 getallen zonder meer bij elkaar op in register R0. Toon de som via de "EXTENDED I/O"-poort op de data-LED's. Voer het programma uit in de STEP-mode. Bekijk het resultaat.
- Welke waarde moet men bij de som optellen om een goede decimale som te krijgen?
Voeg deze instructie toe en bekijk het resultaat.
- Nu is de som nog niet correct? Waarom?
Voeg nog één instructie toe en bekijk dan het resultaat opnieuw, dat nu dus correct moet zijn!

Vraagstuk 56

Plaats in het geheugen op de adressen H'0100' t/m H'0105' drie getallen in de BCD-code, en wel de decimale getallen 1258, 0128 en 9527 (dus 2 bytes per getal vereist).

Tel deze getallen vervolgens bij elkaar op en plaats het resultaat in de registers R1 en R2. Zorg dat de som correct is (zie oefening 55). Als men vervolgens de SENSE-toets indrukt, dienen in de "NON-EXTENDED I/O"-mode de minst significante cijfers op de data-LED's getoond te worden. Na opnieuw drukken, de meest significante cijfers.

Vraagstuk 57

- Schrijf een programma dat een binair getal van 8 bits (geen teken, dus ook geen 2-complement) omzet in een BCD-code zonder teken. Daar het resultaat meer dan 1 byte nodig heeft, wordt het resultaat in de registers R0 en R1 geplaatst.
Het getal wordt op de "NON-EXTENDED I/O"-schakelaar aan het programma ter beschikking gesteld (gebruik de SENSE-toets).
- Controleer na uitvoering van het programma of het gewenste resultaat bereikt is.

Vraagstuk 58

Schrijf een programma dat een programma bestaande uit ASCII-tekens, omzet in BCD-tekens.

De ASCII-tekens worden ingevoerd in de "NON-EXTENDED I/O"-mode via de schakelaar (gebruik de SENSE-toets voor de scheiding), geconverteerd en per paar in het geheugen geplaatst, te beginnen op adres H'0100' e.v.

Vraagstuk 59

Schrijf een programma dat het mogelijk maakt de reactietijd te bepalen en in BCD-code op de "NON-EXTENDED I/O" LED's zichtbaar te maken. Maak voor de meting gebruik van een tijds subroutine (zie oefening 44), maar nu met een doorlooptijd van 1 seconde.*

Men drukt eerst op de SENSE-toets.

Na het doorlopen van de tijds subroutine (b.v. 5 seconden), gaat na enige tijd de FLAG-lamp aan. Men drukt nu zo snel mogelijk op de SENSE-toets. De tijd in seconden die verstrijkt, moet nu met behulp van de I/O-LED's zichtbaar gemaakt worden.

* N.B. Voor een nauwkeuriger meting mag men gebruik maken van de 50 Hz interrupt van het lichtnet (schakelaar onderzijde in stand A.C.), doch het programma wordt dan ingewikkelder. De oplossing wordt niet gegeven, maar probeer dit eventueel zelf uit!

Vraagstuk 60

- Schrijf een programma om twee binaire getallen (zonder tekens) te vermenigvuldigen. De getallen staan in het geheugen (zelf laden d.m.v. toetsenbord) op de plaatsen H'0100' en H'0101'. Het resultaat dient op dezelfde plaatsen teruggeschreven te worden.
- Controleer het resultaat na uitvoering van het programma.
- Herhaal de procedure in de STEP-by-STEP-mode en controleer de werking.

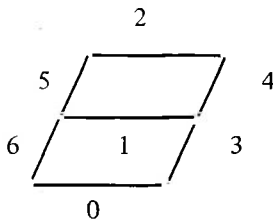
Vraagstuk 61

De I/O-apparatuur geeft via een SENSE-input een signaal. Tegelijkertijd maakt deze zijn oproep kenbaar op een poortingang.

- Schrijf een programma dat de SENSE-bit regelmatig bekijkt. Als er een SENSE is, dan laadt de INSTRUCTOR de nagebootste oproep (door een I/O schakelaar in de nulstand te zetten) van de "EXTENDED I/O"-poort (men noemt dit "polling").
- In register R0 worden de oproepen gecontroleerd en één er uit geselecteerd. Dit geschiedt door de rotatie naar links van de inhoud van R0 door de Carry. Deze Carry wordt gecontroleerd. Tijdens het schuiven wordt R3 gedecrementeerd. Op deze wijze verkrijgt men een index voor de te kiezen subroutine. Voor de subroutines kunnen "dummy"-routines gebruikt worden. Deze dummies plaatsen een nummer in register R1 en zijn dan gereed. Toon de inhoud van R1 op de LED's.

Vraagstuk 62

Van een display-element, bestaande uit 7 segmenten, zijn de segmenten genummerd als in onderstaande tekening is aangegeven:



Een BCD-cijfer is in één byte gegeven, en wel zodanig dat de meest significante bits 0 zijn. Een BCD-cijfer wordt in de geheugenplaats geladen m.b.v. de I/O-schakelaars in de "MEMORY-MAPPED I/O"-mode.

Schrijf een programma dat dit BCD-cijfer in de geheugenplaats laadt na het indrukken en weer loslaten van de SENSE-toets.

Het gelezen cijfer wordt d.m.v. een tabel "vertaald" in de 7-segmenten-code, en vervolgens op de output LED's zichtbaar gemaakt. De nummers van de segmenten komen overeen met de bits van een "BCD-code-vertaal-byte" in de tabel. Het programma moet zodanig ontworpen worden dat telkens een nieuw BCD-cijfer aangeboden kan worden.

Vraagstuk 63

Twee decimale getallen van twee cijfers elk (BCD-code) dienen d.m.v. het toetsenbord in het geheugen geplaatst te worden.

- Schrijf een programma waardoor deze getallen bij elkaar worden opgeteld en het resultaat zichtbaar gemaakt wordt via de "NON-EXTENDED I/O" op de LED's.

Doe dit voor de volgende getallen:

A	B
20	45
40	88
28	46
44	98

- Controleer na afloop van deze operatie de overflow van de getallen.

Vraagstuk 64

Als vraagstuk 63. Nu dient echter de FLAG-lamp te branden als de som groter dan 99 is.

Vraagstuk 65

Twee decimale getallen van 5 cijfers elk dienen opgeteld te worden en de som dient op de plaats van één van de beide getallen geplaatst te worden (zie ook oefening 56).

Zet m.b.v. het toetsenbord de getallen op de plaatsen H'0100' e.v.

Schrijf dit programma. Neem voor de beide getallen de waarden 78395 en 84237.

Wordt de SENSE-toets ingedrukt, dan dienen eerst de minst significante cijfers op de output LED's getoond te worden. Dan, bij het opnieuw indrukken, de volgende twee, en tenslotte dienen de twee meest significante cijfers van de som getoond te worden.

Vraagstuk 66

- a) Schrijf een programma dat twee decimale getallen in het geheugen zonder teken van twee cijfers elk van elkaar aftrekt ($A - B$), en het resultaat op een van de geheugenplaatsen plaatst. Indien het resultaat negatief is, dient de FLAG-lamp te gaan branden. De getallen dienen d.m.v. het toetsenbord in het geheugen geplaatst te worden.

Neem de volgende waarden:

A	B
40	16
48	32
50	74
68	74

- b) Controleer na afloop het resultaat in het geheugen.

Vraagstuk 67

- a) Schrijf een programma dat twee getallen (BCD-code) van elkaar aftrekt ($A - B$). De getallen dienen d.m.v. het toetsenbord in het geheugen geladen te worden. Als het resultaat negatief is, dient de FLAG-lamp te gaan branden. Het resultaat moet op de plaats van getal A geplaatst worden.

Neem de volgende waarden:

A	B
7439	6678
6678	7439

- b) Controleer na de aftrekking de plaats van getal A.

Vraagstuk 68

Ontwerp een programma dat het mogelijk maakt in de spiegelcode van 0 t/m 15 te tellen in modulo 16. Als de schakelaar 0 van de input in de '1'-stand staat, dient er te worden opgeteld, als hij in de '0'-stand staat, dient er te worden afgetrokken. De "tel"-input is de SENSE. Na het indrukken van de SENSE-toets wordt de stand van de teller veranderd. (wacht dan 20 ms).

Direct na het indrukken wordt bepaald of er opgeteld of afgetrokken moet worden. Dit kan gewijzigd worden tijdens het wachten op het indrukken van SENSE (wacht na SENSE 20 ms).

Neem voor elke variabele een byte. Toon de "tel"-stand in de "NON-EXTENDED OUTPUT"-mode op de LED-bits 7, 6, 5 en 4.

Het optellen resp. aftrekken in de spiegel- of Gray-code berust op de onderstaande Boolse formules:

$$\begin{aligned} \text{Optellen: } A &= A(\overline{C+D}) + \overline{BCD} \\ B &= B(\overline{C+D}) + \overline{ACD} \\ C &= D(A \oplus B)' + \overline{CD} \\ D &= (A \oplus B \oplus C)' \end{aligned}$$

$$\begin{aligned} \text{Aftrekken: } A &= A(\overline{C+D}) + \overline{BCD} \\ B &= B(\overline{C+D}) + \overline{ACD} \\ C &= D(A \oplus B) + \overline{CD} \\ D &= (A \oplus B \oplus C) \end{aligned}$$

De telcode is hieronder aangegeven:

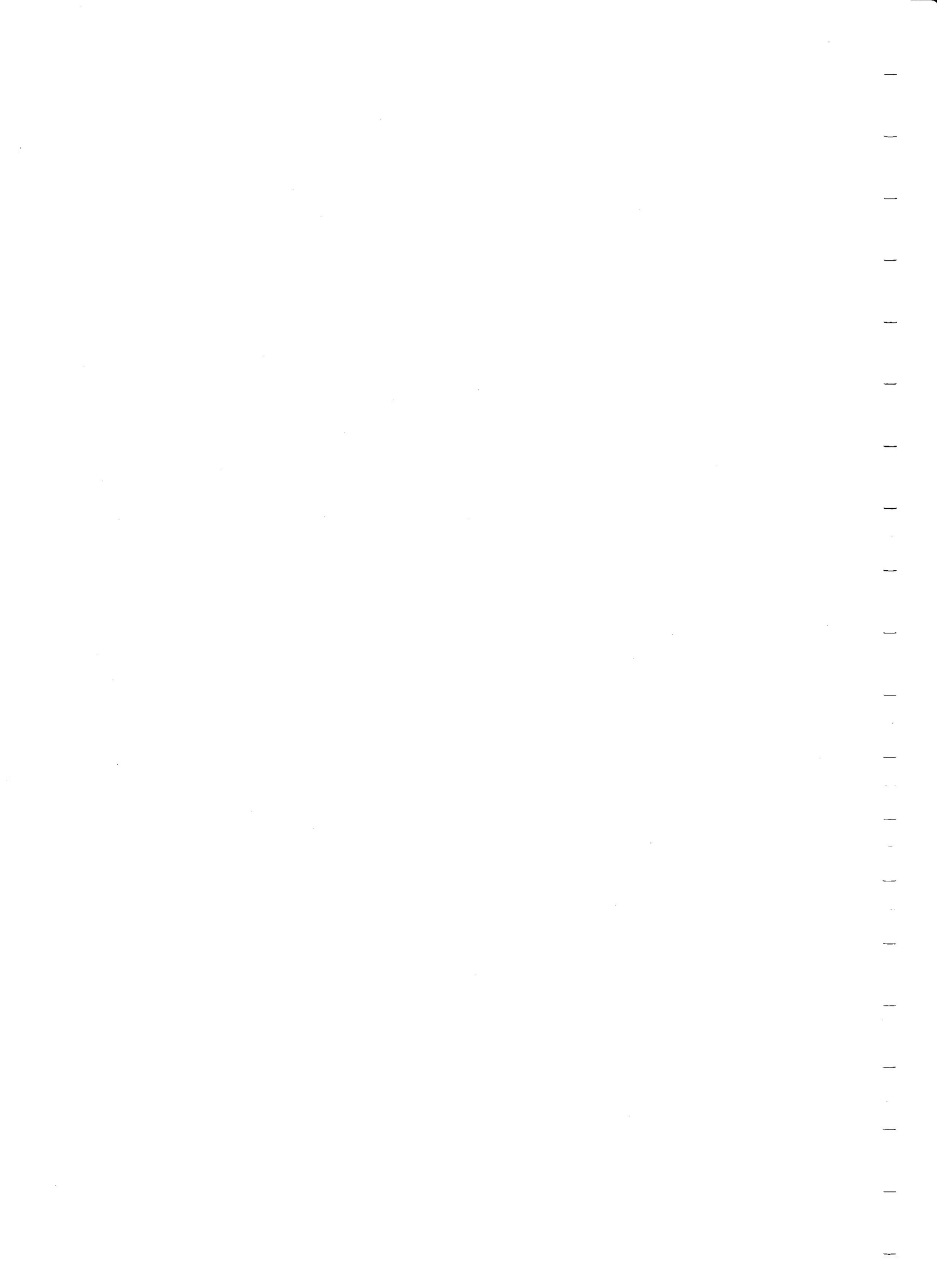
Telcode	A	B	C	D	Om tussen beide routines te kunnen kiezen, is gebruik gemaakt van de BSXA-instructie, die het mogelijk maakt uit twee uitgangen te selecteren. Hiertoe is de waarde van register R3 hetzij 0 dan wel 3.
0	0	0	0	0	Door het sequentiële gedrag van de processor is een stel master-slave variabelen gekozen, die resp. MB, MC, MD, A, B, C, D heten.
1	0	0	0	1	
2	0	0	1	1	
3	0	0	1	0	
4	0	1	1	0	
5	0	1	1	1	
6	0	1	0	1	
7	0	1	0	0	
8	1	1	0	0	
9	1	1	0	1	De delay-routine(s) kan men tijdelijk vervangen door NOP-instructies.
10	1	1	1	1	
11	1	1	1	0	
12	1	0	1	0	
13	1	0	1	1	
14	1	0	0	1	
15	1	0	0	0	

Vraagstuk 69

Ontwerp een programma voor dezelfde tel-methode als in oefening 68, echter met dien verstande dat de vier variabelen in één byte aanwezig zijn en wel in de bits 7, 6, 5 en 4.

We tellen dus binair in de meest significante bits (van R0). Vervolgens wordt de inhoud in R1 geplaatst en één plaats naar rechts geschoven. Dan wordt $R0 \oplus R1$ gevormd in R0. Door nu bit 3 te verwijderen, blijft de spiegelcode over. Het eerste programma is nagenoeg gelijk aan het hoofdprogramma van oefening 68. De delay-subroutine(s) zijn identiek.

ANTWOORDEN OP DE VRAAGSTUKKEN VOOR DE MICROPROCESSOR
2650 EN DE INSTRUCTOR 50



Vraagstuk 1

<i>ingedrukte toets</i>	<i>DP</i>	<i>commentaar</i>
MON	HELLO	
REG	r=	De INSTRUCTOR wacht op een nummer
0	.r0=XX	De inhoud van reg. 0 = XX (de X betekent hier elke hexadecimale waarde die kan voorkomen)
ENT NXT	.r1=XX	Dit is de inhoud van reg. 1 bank 0
ENT NXT	.r2=XX	Idem reg. 2, bank 0
ENT NXT	.r3=XX	Idem reg. 3, bank 0
ENT NXT	.r4=XX	Dit is de inhoud van reg. 1, bank 1
ENT NXT	.r5=XX	Idem reg. 2, bank 1
ENT NXT	.r6=XX	Idem reg. 3, bank 1
ENT NXT	.PU=XX	Dit is de inhoud van het PSU (Program Status Word Upper byte)
ENT NXT	.PL=XX	Dit is de inhoud van het PSL (Program Status Word Lower byte)

Opnieuw indrukken van ENT NXT geeft weer .r0=XX

Vraagstuk 2

<i>ingedrukte toets</i>	<i>DP</i>	<i>commentaar</i>
MON	HELLO	
REG	r=	INSTRUCTOR wacht op een nummer
4	.r4=XX	Dit is de inhoud van reg. 1, bank 1, maar we willen reg. 0 hebben
REG	r=	INSTRUCTOR wacht op een nummer
0	.r0=XX	Inhoud reg. 0 wordt getoond

Vraagstuk 3

Er staat al .r0=XX

- ENT NXT
- .r1=XX (inhoud noteren)
- telkens ENT NXT indrukken

Vraagstuk 4

<i>ingedrukte toets</i>	<i>DP</i>	<i>commentaar</i>
REG	r=	INSTRUCTOR wacht op een nummer
0	.r0=XX	Toont inhoud van reg. 0
twee maal 0	.r0=00	Inhoud reg. 0 wordt H'00'
ENT NXT	.r1=XX	Toont inhoud van reg. 1
1 en dan 0	.r1=10	Inhoud van reg. 1 wordt H'10'

Enz. steeds ENT NXT en de goede hexadecimale waarden indrukken t/m .r6=13; daarna:

ENT NXT	.PU=XX	Inhoud PSW Upper (PSU) = H'XX'
2 x F	.PU=FF	PSU wordt H'FF'
ENT NXT	.PL=XX	Inhoud PSW Lower (PSL) = H'XX'
2 x E	.PL=EE	PSL wordt H'EE'
ENT NXT		Vergeet laatste ENT NXT niet!

Herhaal de procedure ter controle en eventuele verbetering.

Vraagstuk 5

<i>ingedrukte toets</i>	<i>DP</i>	<i>commentaar</i>
MEM	.Ad.=	INSTRUCTOR wacht op een adres
2 x (of 4 x) 0	.Ad.=0000	
ENT NXT	.0000 XX	De inhoud van geheugenplaats H'0000' wordt getoond (noteer dit!)
2 x toets 0	.0000 00	De inhoud wordt H'00'
ENT NXT	.0001 XX	Toont inhoud van adres H'0001'
0 en dan 1	.0001 01	De inhoud wordt H'01'

Enz., steeds ENT NXT en de goede hexadecimale waarden indrukken. Vergeet de laatste ENT NXT niet!

Vraagstuk 6

<i>ingedrukte toets</i>	<i>DP</i>	<i>commentaar</i>
MEM	.Ad.=	INSTRUCTOR wacht op adres
2 x (of 4 x) 0	.Ad.=0000	
ENT NXT	.0000 00	De inhoud van adres H'0000' wordt getoond (moet H'00' zijn)
ENT NXT	.0001 01	

Enz., steeds ENT NXT indrukken, waarna het adres en de inhoud van dat adres getoond wordt (als het goed is, moet dus steeds de inhoud gelijk zijn aan de 2 meest rechtse cijfers van het adres).

Vraagstuk 7

<i>ingedrukte toets</i>	<i>DP</i>	<i>commentaar</i>
MEM	.Ad.=	INSTRUCTOR wacht op adres
b.v. 1 en 3	.Ad.=0013	Het gekozen adres is b.v. H'0013'
ENT NXT	.0013 13	De inhoud van H'0013' is H'13'
2 x toets 0	.0013 00	Wordt veranderd in H'00'
ENT NXT		Vergeet de laatste ENT NXT niet!

Vraagstuk 8

<i>ingedrukte toets</i>	<i>DP</i>	<i>commentaar</i>
REG	r=	
F	.Ad.=	Snellaadmethode wacht op adres
2 x (of 4 x) 0	.Ad.=0000	Beginadres snelladen is H'0000'
ENT NXT	.0000	Wacht op waarde voor adres H'0000'
2 x toets 0	.0000 00	Inhoud van H'0000' wordt H'00'
weer 0	.0001 0	Volgende adres wordt genomen
toets 1	.0001 01	Inhoud van H'0001' is nu H'01'
weer 0	.0002 0	Volgende adres wordt genomen
toets 2	.0002 02	Inhoud van H'0002' wordt H'02'

Enz., t/m adres H'001F', druk dan een andere toets in (b.v. MON)

Vraagstuk 9

Snel controleren kan niet, men moet dus hetzelfde doen als voor oefening 6!

Vraagstuk 10

Antwoord 10a

Program Counter (PC)

Antwoord 10b

<i>ingedrukte toets</i>	<i>DP</i>	<i>commentaar</i>
REG	r=	
C	.PC=XXXX	De inhoud van PC is H'XXXX'

Druk dan achtereenvolgens 0, 1, E en F in. De nieuwe waarde wordt:

.PC=01EF

Vraagstuk 11

<i>ingedrukte toets</i>	<i>DP</i>	<i>commentaar</i>
REG	r=	
C	.PC=01EF	

Vraagstuk 12

Antwoord 12a

<i>ingedrukte toets</i>	<i>DP</i>	<i>commentaar</i>
MEM	.Ad.=	INSTRUCTOR wacht op een adres
2 x (of 4 x) 0	.Ad.=0000	Beginadres
ENT NXT	.0000 XX	Toont inhoud van adres H'0000'

C en 0	.0000	C0	De inhoud van H'0000' wordt H'C0' (=NOP)
ENT NXT	.0001	XX	
C en 0	.0001	C0	H'C0' = NOP
ENT NXT	.0002	XX	
4 en 0	.0002	40	H'40' = HALT
ENT NXT			Vergeet de laatste ENT NXT niet!

N.B. Men kan dit eventueel ook met de snellaadmethode doen, dus door i.p.v. MEM nu REG F in te drukken (ENT NXT vervalt dan steeds).

Antwoord 12b

<i>ingedrukte toets</i>	<i>DP</i>	<i>commentaar</i>
REG	r=	
7	.PU=XX	Toont inhoud PSW Upper = H'XX'
2 x 0	.PU=00	PSW Upper wordt H'00'
ENT NXT	.PL=XX	Toont inhoud PSW Lower = H'XX'
2 x 0	.PL=00	PSW Lower wordt H'00'
ENT NXT		Vergeet laatste ENT NXT niet!

Antwoord 12c

<i>ingedrukte toets</i>	<i>DP</i>	<i>commentaar</i>
REG	r=	
7	.PU=00	Toont inhoud PSW Upper = H'00'
ENT NXT	.PL=00	Toont inhoud PSW Lower = H'00'
MEM	.Ad.=	
2 x (of 4 x) 0	.Ad.=0000	Geef adres H'0000' op
ENT NXT	.0000	C0 De inhoud van adres H'0000' is H'C0' (=NOP)
ENT NXT	.0001	C0 De inhoud van adres H'0001' is H'C0' (=NOP)
ENT NXT	.0002	40 De inhoud van adres H'0002' is H'40' (= HALT)

Antwoord 12d

<i>ingedrukte toets</i>	<i>DP</i>	<i>commentaar</i>
REG	r=	
C	.PC=XXXX	Toont waarde PC

2 x (of 4 x) 0	.PC=0000	Verander in H'0000'
ENT NXT		
STEP	0001 C0	NOP-instructie
STEP	0002 40	HALT-instructie
STEP		Programma klaar, RUN-lamp gaat uit
RST	0000 C0	Toont NOP-instructie op adres H'0000'
MON	HELLO	Terug naar MONITOR

Vraagstuk 13

Antwoord 13a

<i>ingedrukte toets</i>	<i>DP</i>	<i>commentaar</i>
REG	r=	
0	.r0=XX	Toont de inhoud van register R0 = H'XX'
2 x A	.r0=AA	Verander in H'AA'
ENT NXT	.r1=XX	Toont de inhoud van register R1 = H'XX'
2 x F	.r1=FF	Verander in H'FF'
ENT NXT		

Antwoord 13b

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>
0000	01	laad reg. R1 in R0
0001	40	zet HALT-instructie
		} N.B. Van nu af aan zal het laden van een programma als bekend worden verondersteld!
<i>ingedrukte toets</i>	<i>DP</i>	<i>commentaar</i>

REG	r=	
C	.PC=XXXX	Toont waarde PC
2 x (of 4 x) 0	.PC=0000	Verander in H'0000'
ENT NXT		
RUN		Start het programma
Als het programma uitgevoerd is, dooft het RUN-lampje.		
MON		Terug naar MONITOR
RST	HELLO	RUN-lampje gaat weer aan*
REG	r=	
0	.r0=FF	Toont inhoud reg. R0 = H'FF'
ENT NXT	.r1=FF	Toont inhoud reg. R0 = H'FF'

* Dit is de aangewezen methode om na een HALT weer in de MONITOR-mode te komen; dus eerst MON, en daarna de RST-toets indrukken!

Vraagstuk 14

Antwoord 14a

<i>adres</i>	<i>instructie</i>		<i>commentaar</i>
0000	04	00	Laad reg. R0 met H'00'
0002	05	AA	Laad reg. R1 met H'AA'
0004	01		Laad inhoud R1 in R0
0005	40		Zet HALT-instructie

Antwoord 14b

RST RUN-lampje dooft als programma klaar is

Antwoord 14c

<i>ingedrukte toets</i>	<i>DP</i>	<i>commentaar</i>
REG	r=	
C	.PC=XXXX	Toont waarde PC=H'XXXX'
2 x (of 4 x) 0	.PC=0000	Maak PC = 0
ENT NXT		
STEP	0002 05	Laad R1-instructie
STEP	0004 01	Laad inhoud R1 in R0
STEP	0005 40	HALT-instructie
STEP		RUN-lampje dooft, programma klaar
RST	0000 04	Laad R0-instructie wordt getoond en het RUN-lampje gaat weer aan

Antwoord 14d

<i>ingedrukte toets</i>	<i>DP</i>	<i>commentaar</i>
REG	r=	
0	.r0=AA	Toont inhoud R0 = H'AA'

Vraagstuk 15

Antwoord 15a

<i>adres</i>	<i>instructie</i>		<i>commentaar</i>
0000	0A	22	Laad R2 relatief; PC geeft adres aan van de volgende instructie, dus invullen $24 - 2 = 22$
0002	40		Zet HALT-instructie
0024	EF		Zet op adres H'0024' de waarde H'EF'

Antwoord 15b

<i>ingedrukte toets</i>	<i>DP</i>	<i>commentaar</i>
MON		Terug naar MONITOR
RST	HELLO	RUN-lampje dooft, programma klaar
REG	r=	
2	.r2=EF	In reg. R2 staat nu H'EF'

Antwoord 15c

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>
0000	0E 80 24	Laad (indirect) adres H'0024'
0003	40	Zet HALT-instructie
0024	01	} Op H'0024', H'0025' staat het adres van de data, n.l. adres H'0100'
0025	00	
0100	EF	Adres H'0100' bevat de data H'EF'

Druk vervolgens toets RST in

Vraagstuk 16**Antwoord 16a**

<i>ingedrukte toets</i>	<i>DP</i>	<i>commentaar</i>
MEM	.Ad.=	
00AA	.Ad.=00AA	Sla aan 0, 0, A, A
ENT NXT		
	.00AA=AA	Toont data op adres H'00AA' = H'XX'
2 x F	.00AA=FF	Verander in H'FF'
ENT NXT		Definitief veranderd

Antwoord 16b

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>
0000	08 A8	laad R0 relatief met data van adres H'00AA' (AA - 2 = A8).
0002	40	Zet HALT-instructie

Antwoord 16c

druk RST in; als programma klaar is, dooft het RUN-lampje

druk MON in, dan RST, vervolgens REG en tenslotte 0

vóór de uitvoering: .r0=XX (toont de inhoud R0 = H'XX')

na de uitvoering: .r0=FF (de inhoud van R0 is nu H'FF')

Antwoord 16d

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>
0000	0F 80 0F	Laad R3 indirect met adres H'000F'
0003	HALT	ZET HALT-instructie
000F	00	} Op H'000F', H'0010' staat het adres van de data, n.l. adres H'00AA'
0010	AA	

druk nu RST in, enz.

Vraagstuk 17

Antwoord 17a

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>
0000	0D 21 00	Laad absoluut R0 (indexregister =R1) met auto-increment (dus 0D 2XXX) met adres H'0100' e.v.
0003	CD 61 10	Berg op absoluut R0 (indexregister =R1) indexed (CC=11), dus CD 6XXX met adres H'0110' e.v.
0006	E5 04	Is inhoud R1 = 4?
0008	98 76	Zo ja, klaar; zo nee, spring dan terug naar 0 relatief, displacement = -10 (H'76')
000A	40	Zet HALT-instructie

<i>ingedrukte toets</i>	<i>DP</i>	<i>commentaar</i>
MEM	.Ad.=	
0100	.Ad.=0100	Sla aan: 0, 1, 0, 0
ENT NXT	.0100=XX	Toont inhoud
0A	.0100=0A	Verander in H'0A'
ENT NXT	.0101=XX	Enz.
.	.	.
.	.	.
	.0105=0E	Laatste waarde ingevoerd
ENT NXT		Vergeet laatste ENT NXT niet!
REG	r=	
1	.r1=XX	Toont inhoud R1 = H'XX'
FF	.r1=FF	Verander in -1 (=H'FF')
ENT NXT		
REG	r=	
C	.PC=XXXX	Toont waarde PC
2 x (of 4 x) 0	.PC=0000	Maak PC = 0
ENT NXT		PC = 0 definitief
RUN		Voer programma uit

Antwoord 17b

MEM	.Ad.=	
0110	.Ad.=0110	Sla aan: 0, 1, 1, 0
ENT NXT		
	.0110 0A	Klopt inderdaad!
ENT NXT	.0111 0B	Enz.

Antwoord 17c

REG	r=	
0	.r0=0E	Toont inhoud R0 = H'0E' (= laatste data-waarde)
ENT NXT	.r1=04	Inhoud R1 = 4 (laatste index-waarde)

Vraagstuk 18**Antwoord 18a**

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>
0000	05 FF	Initieer indexregister R1 = -1
0002	0D A0 40	Laad indirect absoluut R0 (indexregister =R1) met auto-increment (IC=01), dus 0D AXXX (8+2='A') met adres H'0040'
0005	CD E0 42	Berg op indirect absoluut R0 (indexregister =R1), indexed (IC=11), dus CD EXXX (8+6='E') met adres H'0042'
0008	E5 03	Is inhoud R1 = 3?
000A	98 76	Zo ja, klaar, zo nee, spring dan terug naar H'0002' relatief, displacement = -10 (=H'76')
000C	40	Zet HALT-instructie
0040	01	} Op H'0040', H'0041' staat het begin-adres van "oorsprong", nl. H'0100'
0041	00	
0042	01	} Op H'0042', H'0043' staat het begin-adres van "doel", nl. H'010F'
0043	0F	

ingedrukte toets DP *commentaar*

MEM	.Ad.=	
0100	.Ad.=0100	Sla aan: 0, 1, 0, 0
ENT NXT	.0100 XX	Toont inhoud
02	.0100 02	Verander in 02
ENT NXT	.0101 XX	Enz.
00	.0103 00	Laatste waarde ingevoerd
ENT NXT		Niet vergeten!
RST		Programma wordt uitgevoerd

Antwoord 18b

Ga tewerk zoals bij 17b:

<i>ingedrukte toets</i>	<i>DP</i>	<i>commentaar</i>
MEM	.Ad=	
010F	.Ad.=010F	Sla aan: 0, 1, 0, F=beginadres "doel"
ENT NXT	010F 02	Controleer of alles goed is Enz.

Vraagstuk 19

Antwoord 19a

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>
0000	05 FF	Initieer indexreg. R1 = -1
0002	06 05	Initieer indexreg. R2 = 5
0004	0D 21 00	Laad absoluut R0 (i.r.=R1) met auto-increment (IC=01) adres H'0100' e.v.
0007	CE 41 10	Berg op absoluut R0 (i.r.=R2) met auto-decrement (IC=10) adres H'0110' e.v.
000A	E5 04	Is inhoud R1 = 4?
000C	98 76	Zo niet, spring dan terug relatief naar adres H'0004'
000E	40	Zet HALT-instructie

Antwoord 19b

Zoals bij oefening 17.

Antwoord 19c

Zoals bij oefening 17.

Vraagstuk 20

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>
0000	05 FF	Initieer indexregister R1 = -1 (H'FF')
0002	06 04	Initieer indexregister R2 = 4
0004	0D A0 40	Zie commentaar oefening 18
0007	CE C0 42	Idem, maar auto-decrement (IC=10) (8+4='C')
000A	E5 03	
000C	98 76	
000E	40	
0040	01 00	Beginadres "oorsprong"
0042	01 0F	Beginadres "doel"

Druk RST-toets in en zie verder antwoord op oefening 18b

Vraagstuk 21

Antwoord 21a

0000	02	Laad inhoud R2 in R0
0001	81	Tel inhoud R1 op bij R0.
0002	C1	Laad R0 in R1
0003	40	Zet HALT-instructie

Antwoord 21b

Conversie: $64 = H'40'$, $52 = H'34'$, $120 = H'78'$

$$\left. \begin{array}{l} -64 = 255 - 64 + 1 = 192 = H'C0' \\ -52 = 255 - 52 + 1 = 204 = H'CC' \\ -120 = 255 - 120 + 1 = 136 = H'88' \end{array} \right\} \begin{array}{l} (H'FF' = 255) \\ 2\text{-complement notatie} \end{array}$$

Laden door REG, 1 en de waarde aan te slaan, vervolgens ENT NXT en de tweede waarde aanslaan.

a: R1 = 74	(decimaal 116)	.PL= 42, dus OVF = 0	C = 0
b: R1 = 8C	(decimaal -116)	.PL= 83, dus OVF = 0	C = 1
c: R1 = F4	(decimaal -12)	.PL= 82, dus OVF = 0,	C = 0
d: R1 = 0C	(decimaal -12)	.PL= 43, dus OVF = 0,	C = 1
e: R1 = AC	(overflow)	.PL= 86, dus OVF = 1,	C = 0
f: R1 = 44	(decimaal 68)	.PL= 63, dus OVF = 0,	C = 1
g: R1 = BC	(decimaal -68)	.PL= 82, dus OVF = 0,	C = 0
h: R1 = 54	(overflow)	.PL= 67, dus OVF = 1,	C = 1

Vraagstuk 22

Conversie: $20 = H'14'$, $60 = H'3C'$

$$-80 = 255 - 80 + 1 = 176 = H'B0' \quad 2\text{-complement notatie}$$

Antwoord 22a

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>
0000	0C 00 10	Laad R0 met inhoud adres H'0010'
0003	0D 00 11	Laad R1 met inhoud adres H'0011'
0006	0E 00 12	Laad R2 met inhoud adres H'0012'
0009	81	Tel R1 op bij R0, $R0 = R0 + R1$
000A	82	Tel R2 op bij R0, $R0 = (R0 + R1) + R2$
000B	CC 00 10	Berg inhoud R0 op in adres H'0010'
000E	40	Zet HALT-instructie
000F	C0	"Dummy"
0010	14	$H'14' =$ decimaal 20
0011	B0	$H'B1' =$ decimaal -80
0012	3C	$H'3C' =$ decimaal 60

Antwoord 22b

Na afloop is de inhoud van $H'0010' = H'00' = 0$;
.PL= 23, dus OVF = 0 en C = 1

Vraagstuk 23

Antwoord 23a

0000	0C	01	0F	Laad inhoud van adres $H'010F'$ in R0
0003	84	05		Tel 5 op: $02 + 5 = H'07'$
0005	CC	01	0F	Zet 7 terug op $H'010F'$
0008	0C	01	10	Enz., tel 5 op bij de andere waarden
.				.
....	40			Zet HALT-instructie

De procedure vergt minder tijd met indirecte indexering! Vergelijk oefening 24.

Antwoord 23b

Maak PC = 0 en druk RUN-toets in

Antwoord 23c

Maak PC = 0 en druk STEP-toets in

STEP	0003	84	Optelinstructie R0
STEP	0005	CC	Opberginstructie R0

enz.

Vraagstuk 24

Antwoord 24a

<i>adres</i>	<i>instructie</i>			<i>commentaar</i>
0000	05	FF		Initieer indexregister R1 = -1
0002	0D	A0	40	Laad indirect R0 (zie oefening 18)
0005	84	05		Tel 5 op
0007	CD	E0	40	Berg op indirect R0
000A	E5	03		Is R1 = 3?
000C	98	74		Zo niet, ga dan terug relatief naar adres $H'0002'$
000E	40			Zet HALT-instructie
0040	01			Beginadres van "oorsprong" is $H'0100'$
0041	00			

Antwoord 24b

Maak PC=0 en druk op de RUN-toets

Antwoord 24c

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>
.PC = 0		
STEP	0002 0D	Laad absoluut (indirect) R0-instructie
STEP	0005 84	Tel op bij R0 instructie
STEP	0007 CD	Berg op absoluut (indirect) instructie Enz.

Vraagstuk 25**Antwoord 25a**

Laad d.m.v. toetsenbord in R0 de vector '11110101' = H'F5'

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>
0000	05 9E	Laad R1 met '10011110' = H'9E'
0002	41	R0 "EN" R1, resultaat in R0
0003	40	Zet HALT-instructie
voor: .r0 = F5	'11110101'	
na : .r0 = 94	'10010100'	PL = A2, dus CC = 10

Antwoord 25b

voor: .r0 = A3	('10100011')	
na : .r0 = 82	('10000010')	PL = 82, dus CC = 10
voor: .r0 = 60	('01100000')	
na : .r0 = 00	('00000000')	PL = 02, dus CC = 00
voor: .r0 = 33	('00110011')	
na : .r0 = 12	('00010010')	PL = 42, dus CC = 01
voor: .r0 = 7A	('01111010')	
na : .r0 = 1A	('00011010')	PL = 42, dus CC = 01

Vraagstuk 26**Antwoord 26a**

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>
0000	05 B0	Laad R1 met '10110000' = H'B0'
0002	61	R0 "OF" R1, resultaat in R0
0003	40	Zet HALT-instructie
voor: .r0 = F5	('11110101')	
na : .r0 = F5	('11110101' = hetzelfde)	PL = 82, dus CC = 10

Antwoord 26b

```

voor: .r0 = A3 ('10100011')
na  : .r0 = B3 ('10110011') .PL = 82, dus CC = 10 (was bij "EN" 10)
-----
voor: .r0 = 60 ('01100000')
na  : .r0 = F0 ('11110000') .PL = 82, dus CC = 10 (was 00)
-----
voor: .r0 = 33 ('00110011')
na  : .r0 = B3 ('10110011') .PL = 82, dus CC = 10 (was 01)
-----
voor: .r0 = 7A ('01111010')
na  : .r0 = FA ('11111010') .PL = 82, dus CC = 10 (was 01)
-----

```

Bij "OF" is de CC steeds '10', het resultaat is altijd groter dan de oorspronkelijke waarde.

Vraagstuk 27**Antwoord 27a**

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>
0000	04 F0	Laad R0 met H'F0' = '11110000'
0002	24 A9	"EXcl.OR" immediate met '10101001' = H'A9' (inv. bits 0, 3, 5, 7, overige ongewijzigd laten)
0004	40	Zet HALT-instructie

Antwoord 27b

```

vóór: .r0 = F0 ('11110000')
ná  : .r0 = 59 ('01011001') .PL = 42, dus CC = 01

```

Vraagstuk 28**Antwoord 28a**

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>	
0000	07 05	Laad R3 met H'05'	('00000101')
0002	04 17	Laad R0 met H'17'	('00010111')
0004	D3	Rotate links R3 (H'0A' =	'00001010')
0005	83	Tel R3 op bij R0 (H'21' =	'00100001')
0006	D0	Rotate links R0 (H'42' =	'01000010')
		R3 = H'0A' =	'00001010')
0007	A3	R0 = R0 - R3 (H'38' =	'00111000')
0008	40	Zet HALT-instructie	

Antwoord 28b

We zien dat het resultaat H'38' is (zie boven).

Vraagstuk 29

Antwoord 29a

0000	04 40	Laad R0 met H'40' (= flag bit = '1')
0002	92	Laad inhoud R0 in het PSU
0003	40	Zet HALT-instructie

Antwoord 29b

.PC = 0

RUN FLAG-lamp gaat aan.

.PU =61 FLAG-bit = '1'

Vraagstuk 30

Antwoord 30a

ingedrukte toets *DP* *commentaar*

REG r=

0 .r0 = 40 of 00 (FLAG-bit "aan" of "uit")

adres *instructie* *commentaar*

0000 92 Laad inhoud R0 in het PSU ('40' of '00')

0001 40 Zet HALT-instructie

Antwoord 30b

.PC = 0000

RUN FLAG-lampje gaat "aan" of "uit"

"aan" .PU = 61 "uit" .PU = 21

Vraagstuk 31

Antwoord 31a

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>
0000	77 00	Maak PSL = 0, dus kies bank 0
0002	07 33	Laad R3 met H'33'
0004	03	Laad inhoud R3 in R0

Antwoord 31b

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>
0005	77 10	Laad PSL (RS-bit) met H'10', dus bank 1
0007	C2	Laad inhoud R0 in R2'
0008	40	Zet HALT-instructie

Antwoord 31c

Druk RST-toets in. Nadat het programma is uitgevoerd (RUN-lampje dooft dan), de MON- en RST-toets indrukken.

<i>ingedrukte toets</i>	<i>DP</i>	<i>commentaar</i>
REG	r=	
0	.r0= 33	De waarde H'33' staat nog in R0
3	.r0= 33	En ook nog in R3, bank 0
5	.r5= 33	En ook in R2, bank 1

Vraagstuk 32**Antwoord 32a**

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>
0000	77 08	Maak PSL RS-bit 0 (bank 0) en WC-bit 1 (WC = With Carry)
0002	05 80	Laad register R1, bank 0 met H'80'
0004	D1	Rotate links R1 door de Carry
0005	77 18	Maak PSL RS-bit 1 (bank 1) en WC = 1
0007	06 00	Maak inhoud R2 = 0
0009	D2	Rotate links R2, bank 1
000A	40	Zet HALT-instructie

Antwoord 32b

Druk RST-toets in. Nadat het programma is uitgevoerd (RUN-lampje dooft dan), de MON- en RST-toets indrukken.

<i>ingedrukte toets</i>	<i>DP</i>	<i>commentaar</i>
REG	r=	
8	.PL= 1A	('00011010') RS=1, WC=1, Carry = 0
2 x ENT NXT	.r1= 00	De meest linkse "1" van H'80' is uit reg. R1 in de Carry geschoven
4 x ENT NXT	.r5= 01	De Carry-bit is in reg. R2 bank 1 geschoven, Carry is nu dus '0'

Vraagstuk 33

Antwoord 33a

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>
0000	77 02	Maak PSL COM-bit='1' (logisch vergelijken)
0002	0C 01 00	Laad absolute 1ste vector H'97' in reg. R0
0005	EC 01 01	Vergelijk absolute "logische" inhoud R0 met 2de vector, die op plaats H'0101' staat
0008	40	Zet HALT-instructie

Antwoord 33b

Druk RST-toets in. Nadat het programma is uitgevoerd (RUN-lampje dooft dan), de MON- en RST-toets indrukken.

<i>ingedrukte toets</i>	<i>DP</i>	<i>commentaar</i>
MON		
REG	r=	
8	.PL= 82	Dus CC = 10, d.w.z. de vector H'97' in R0 is <i>kleiner</i> dan de 2de vector H'34' bij "logisch" vergelijken

Vraagstuk 34

Antwoord 34a

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>
0000	77 00	Maak PSL COM-bit = '0' ("aritmetisch" vergelijken)
(idem als oefening 33)		

Antwoord 34b

.PL = 82, dus CC = 10. Dus ook bij "aritmetisch" vergelijken is de 1ste vector kleiner dan de 2de vector

Vraagstuk 35

Antwoord 35a

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>
0000	76 00	Maak PSU = 0, dus ook de SENSE-bit
0002	B4 80	Test de SENSE-bit van het PSU. Als de SENSE-toets ingedrukt is: = '1', anders = '0'
0004	98 7C	Niet ingedrukt, spring terug relatief naar adr. H'0002' displace = -4 (H'7C')
0006	40	Zet HALT-instructie

Antwoord 35b

Maak PC = 0, druk de RUN-toets in. Na afloop dooft de RUN-lamp; druk nu de MON- en RST-toets in.

Druk REG en 8 in: .PL= 02, dus CC = 00; de SENSE-bit is nu derhalve '0'.

Vraagstuk 36

Antwoord 36a

<i>ingedrukte toets</i>	<i>DP</i>	<i>commentaar</i>
MEM	.Ad.=	
0100	.Ad.=0100	Sla aan: 0, 1, 0, 0
	0100 XX	Inhoud van adres H'0100' is H'XX'
	0100 80	Verander in H'80'
ENT NXT	0101 XX	
	0101 44	Verander in H'44'
ENT NXT	0102 XX	
	0102 67	Verander in H'67'
ENT NXT	0103 XX	Vergeet laatste ENT NXT niet!

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>
0000	20	Maak register R0 = 0 d.m.v. "EXcl.OR"
0001	05 FF	Laad indexregister R1 met -1 (= H'FF')
0003	8D A1 00	Tel op absolute bij R0 (indexregister = R1) met auto-increment (IC=01) met indirect adres H'0100' e.v.
0006	E5 02	Is inhoud register R2 = 2?
0008	98 79	Zo niet, spring terug relatief naar H'0003'
000A	77 00	Maak PSL COM-bit = 0 ("aritmetisch" vergelijken)
000C	E4 00	Vergelijk inhoud R0 met 0, CC=10 als R0 < 0
000E	9A 02	Spring naar HALT als CC≠0, dus R0≠0
0010	04 00	Maak inhoud R0 = 0
0012	40	Zet HALT-instructie

Antwoord 36b

<i>ingedrukte toets</i>	<i>DP</i>	<i>commentaar</i>
REG	r=	
0	.r0= 2B	
REG	r=	
8	.PL= 86	Dus CC = 10, OVF = 1 (resultaat negatief)

Vraagstuk 37

Antwoord 37a

Idem als oefening XX, maar zet op adres H'0100' de waarde H'01'.

Antwoord 37b

<i>ingedrukte toets</i>	<i>DP</i>	<i>commentaar</i>
REG	r=	
0	.r0= 35	
REG	r=	
8	.PL= 63	Dus C = 01, OVF = 0 (resultaat positief)

Vraagstuk 38**Antwoord 38a**

<i>ingedrukte toets</i>	<i>DP</i>	<i>commentaar</i>
MEM	.Ad.=	
0100	.Ad.=0100	Sla aan: 0, 1, 0, 0
ENT NXT	0100 XX	Inhoud adres H'0100' is H'XX'
01	0100 01	Verander in H'01'
ENT NXT	0101 02	Enz.
.	.	.
0A	0109 0A	Laatste waarde = 10 (H'0A')
ENT NXT		Vergeet de laatste ENT NXT niet!

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>
0000	20	Maak inhoud R0 = 0 d.m.v. "EXcl.OR"
0001	05 FF	Laad indexregister R1 met (=H'FF')
0003	8D 21 00	Tel absoluut op bij R0 (indexregister=R1) met auto-increment (IC=01), dus 8D 2X XX met inhoud adres H'0100' e.v.
0006	E5 09	Is inhoud indexregister R1 = 9 ?
0007	98 79	Zo niet, spring terug relatief naar H'0003'
0008	40	Zet HALT-instructie

Druk RST toets in

Antwoord 38b

<i>ingedrukte toets</i>	<i>DP</i>	<i>commentaar</i>
REG	r=	
0	.r0= 37	Som is H'37' = 55 in R0
ENT NXT	.r1= 09	Laatste indexwaarde 09 in R1

Vraagstuk 39

Antwoord 39a

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>
0100	76 40	Zet FLAG-bit van PSU "aan"
0102	05 C8	Laad register R1 met H'C8' = 200
0104	F9 7E	Tijdslus; decrement R1; spring 2 plaatsen terug relatief
0106	74 40	Zet FLAG-bit van PSU "uit"
0108	40	Zet HALT-instructie

De instructie op adres H'0102' duurt 6 klokperiodes.

De instructie op adres H'0104' duurt 9 klokperiodes. Deze wordt 200 keer doorlopen. 1 klokperiode = 1 μ s Dus de tijd tussen de instructie op H'0100' (FLAG "aan") en H'0106' (FLAG "uit") bedraagt:

$$t = (6 + 200 \times 9) \times 1 \mu\text{s} = 1806 \mu\text{s} = 1,8 \text{ ms}$$

Antwoord 39b

<i>ingedrukte toets</i>	<i>DP</i>	<i>commentaar</i>
REG	r=	
C	.PC=XXXX	Toont waarde PC = H'XXXX'
0100	.PC=0100	Verander in H'0100'
ENT NXT		Niet vergeten!
RUN		Programma wordt uitgevoerd

Vraagstuk 40

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>
0100	76 40	Maak FLAG-bit = '1' ("aan")
0102	05 00	Laad R1 met 0
0104	06 00	Laad R2 met 0
0106	07 20	Laad R3 met H'20' = 32 (of een andere waarde (tijdslus wordt korter of langer))
0108	F9 7E	decrement R1, spring 2 plaatsen terug relatief als $\neq 0$
010A	FA 7C	decrement R2, spring 4 plaatsen terug relatief als $\neq 0$
010C	FB 7A	decrement R3, spring 6 plaatsen terug relatief als $\neq 0$
010E	74 40	Maak FLAG-bit = '0' ("uit")
0110	40	

Vraagstuk 41

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>
0000	C0	Pas op met adres H'0000' en H'0001'
0001	C0	Maak er "dummies" van (NOP)
0002	B4 80	Test PSU SENSE-bit
0004	98 7C	Indien '0', keer dan terug naar H'0002' rel.
0006	1F 01 00	Indien '1' spring dan onvoorwaardelijk naar adres H'0100'

0100	76 40	Zie oefening 40
.		Zie oefening 40
010E	74 40	Zie oefening 40

0110	1F 00 02	Spring onvoorwaardelijk terug naar adres H'0002'
0113	40	Zet HALT-instructie

Vraagstuk 42

Antwoord 42a

Wijzig	:	0004	98	7C	Branch on Condition False ("loop" als S='1')
in	:	0004	18	7C	Branch on Condition True ("loop" als S='0')

Antwoord 42b

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>
0006	77 10	Zet RS= 1 van PSL, dus bank 1
0008	05 01	(eventueel andere waarde). Tijdlus voor "debouncing" tussen "aan" en "uit" van FLAG-lampje. Decrement R1, spring 2 plaatsen terug als ≠0 Decrement R2, spring 4 plaatsen terug als ≠0
000A	06 00	
000C	F9 7E	
000E	FA 7C	
0010	77 00	Zet RS = 0, kies bank 0
0012	1F 01 00	Spring onvoorwaardelijk naar adres H'0100'

Vraagstuk 43

Conversie:

$$326 = H'0146'$$

$$852 = H'0354'$$

$$230 = H'00E6', \quad -230 = H'FFFF' - H'00E6' + 1 = H'FF1A'$$

$$5 = H'0005'$$

$$302 = H'012E', \quad -302 = H'FFFF' - H'012E' + 1 = H'FED2'$$

Invoeren met snellaadmethode:

druk REG- en F-toets in en vervolgens 01 46 03 54 enz. Vergeet tot slot niet, ENT NXT aan te slaan!

a) Programma

Voor het minst significante deel luidt het programma:

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>
0000	20	Maak R0 = 0
0001	07 0A	Laad indexregister R3 met 10
0003	75 11	Clear Carry en kies bank 0
0005	8F C0 40	Tel bij R0 op byte van operand 2 indirect met auto-decrement (indexregister = R3), dus 8F CXXX met adres H'0040'
0008	B5 04	Test OVF-bit van PSL
000A	98 02	Indien geen overflow, ga dan verder
000C	05 01	Indien wel overflow, laad R1 dan met 1
000E	FB 75	Indien decrement R3 ≠ 0, ga dan weer optellen
0010	C2	Heeft men alle bytes gehad, laad dan som in R2

Het programma voor het meest significante deel luidt als volgt:

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>
0011	20	Maak R = 0
0012	07 0B	Laad indexregister R3 met 11
0014	75 11	Clear Carry en kies bank 0
0016	8F C0 42	Tel bij R0 op van operand 1 indirect auto-decrement
0019	E7 02	Is R3 = 2 ?
001B	18 02	Zo ja, spring dan eruit
001D	FB 77	Decrement R3: ga volgende byte van operand 1 optellen
001F	81	R0 = R0 + R1 (overflow)
0020	C1	Laad som (m.s.) in R1
0021	40	Zet HALT-instructie
0040	01	Effectieve adres = H'0100'
0041	00	
0042	00	Effectieve adres - 2 = H'00FE'
0043	FE	

b) Controle

De som bedraagt: $H'028B' = 2 \times 256 + 8 \times 16 + 11 = 651$ decimaal.

Druk de toetsen REG en 0 in.

.r0 = 02 (meest significante deel van de som); druk nu toets ENT/NXT in

.r1 = 02 (in R0 en ook in register R1); druk opnieuw toets ENT/NXT in

.r2 = 8B (minst significante deel van de som); druk weer toets ENT/NXT in

.r3 = 02 (laatste indexwaarde R3); druk toets ENT/NXT vier keer in

.PU = 40 Druk toets ENT/NXT in

.PL = 42 De CC is dus 01

Vraagstuk 44

....	B4	80	Test PSU SENSE-bit
	98	7C	Indien 0, ga dan "loopen"
	76	40	Maak FLAG-bit 1 ("aan")
	3F	01 02	Spring naar tijds subroutine, adres H'0102'
	74	40	Maak FLAG-bit 0 ("uit")
	3F	01 02	Spring naar tijds subroutine
	1B	70	Ga terug naar begin

Voor tijdprogramma zie oefening 40:

0100	76	40	"Zet FLAG" vervalt
0102	04	00	Dus hierna wordt gesprongen
....			
0106	06	0A	Verander 20 in 0A (ca. 5 seconden "delay")
....			
010E	17		Verander in 17 (RETC)

Vraagstuk 45

....	B4	80	Idem als bij oefening 44
....
....	74	40	t/m maak FLAG-bit 0 ("uit")
	3F	80 50	Spring indirect naar tijds subroutine (5 seconden "delay")
	3F	80 50	(twee keer = 10 seconden "delay") *
	1B	6A	Spring relatief terug naar begin

0050	01		Effectieve adres van subroutine = H'0102'
0051	02		
0106	06	14	ca. 10 seconden "delay"

* Men kan natuurlijk i.p.v. twee keer aan te roepen de tijds subroutine wijzigen.

Vraagstuk 46

0000	07	00		Laad R3 met 0 (R3= teller een aantal malen SENSE indrukken)
0002	C0			NOP (men kan hier eventueel F3 invullen: toon de inhoud van R3 op de LED's (NON-EXT))
0003	CF	01	08	Red de inhoud van R3 in adres H'0108'
0006	06	01		Laad R2 met 1 (0,5 seconde "delay" voor het maken van een onderscheid met 2x SENSE indrukken)
0008	3F	01	52	Spring naar tijds subroutine
000B	B4	80		Controleer SENSE-toets
000D	98	7C		Indien 0, ga dan "loopen"
000F	0F	01	08	Ingedrukt, laad teller dan weer in R3
0012	87	01		Verhoog teller met 1
0014	C0			Eventueel hier F3 invullen: toon teller op LED's
0015	CF	01	08	Berg teller weer op
0018	E7	07		Is de teller 7 ?
001A	18	22		Zo ja, ga dan naar HALT
001C	76	80		Maak SENSE-bit "1" van PSU
001E	07	03		Laad R3 met 3 (3x door "loop")
<hr/>				
0020	06	02		Laad R2 met 2 (ca. 1 seconde "delay")
0022	3F	01	52	Spring naar tijds subroutine
0025	B4	80		Controleer SENSE (moet 1 blijven bij lang indrukken)
0027	98	5D		Heeft men niet lang ingedrukt, ga dan terug naar H'000B'
0029	FB	75		Is decrement R3 \neq 0, dan nogmaals "loopen"
<hr/>				
002B	0F	01	08	Laad teller weer in R3 (index voor BSXA)
002E	F7	01		Is teller oneven?
0030	1C	00	00	Zo ja, ga dan terug naar het begin
0033	06	01		Laad R2 met 1 (0,5 seconde "delay")
0035	3F	01	52	Spring naar tijds subroutine
0038	BF	80	80	BSXA, spring (index R3) indirect naar H'0080'
003B	1F	00	00	Na terugkeer uit subroutine ga terug naar begin
003E	40			Zet HALT-instructie
<hr/>				
0080	01			} Effectieve beginadres van de subroutines staat na H'0100' (R3 =0)
0081	00			
<hr/>				
0102	1B	0C		Spring onvoorwaardelijk naar subroutine (R3 = 2)
0104	1B	1A		Spring onvoorwaardelijk naar subroutine (R3 = 4)

0106	1B	38	Spring onvoorwaardelijk naar subroutine (R3 = 6)
0108	00		Hier wordt de teller opgeborgen

De subroutines zijn hieronder opgesomd.

Subroutine 0 (R3 = 2):

0110	76	40	FLAG "aan"
0112	74	80	Maak SENSE-bit "0"
0114	06	02	(ca. 1 seconde "delay" om goed te laten zien dat de FLAG-lamp "aan" is)
0116	3F	01 52	Spring naar tijds subroutine
0119	B4	80	Controleer SENSE-bit
011B	98	7C	Indien 0, ga dan "loopen"
011D	74	40	Indien 1, dan FLAG "uit"
011F	17		Return

Subroutine 1 (R3 = 4):

0120	76	40	FLAG "aan"
0122	3F	01 50	Sprong naar tijds subroutine (5 seconden)
0125	74	40	FLAG "uit"
0127	3F	01 50	5 seconden "delay"
012A	76	40	FLAG weer "aan"
012C	3F	01 50	5 seconden "delay"
012F	74	40	FLAG "uit"
0131	17		Return

Subroutine 2 (R3 = 6):

0140	76	40	FLAG "aan"
0142	3F	01 50	Spring naar tijds subroutine (5 seconden)
0145	74	40	FLAG "uit"
0147	17		Return

Tijds subroutine:

0150	06	0A	(0A = ca. 5 seconden "delay")
0152	04	00	
0154	05	00	
0156	F8	7E	...
0158	F9	7C	Delay-lussen
015A	FA	7A	...
015C	17		Return

Vraagstuk 47

0000	76	40	FLAG "aan"
0002	3B	17	Spring naar tijdsbroutine
0004	1B	03	Spring over adres H'07' heen
0006	C0		NOP ("dummy")

0007	00	10	H'0010' is het effectieve adres van de Interrupt routine

0009	74	40	FLAG "uit"
000B	3B	0E	Spring naar tijdsbroutine
000D	1B	7E	Terug naar begin
000F	40		HALT ("dummy")

0010	07	04	Interrupt routine, maak R3 = 4
0012	3B	07	Spring naar tijdsbroutine (4x = 20 seconden "delay")
0014	FB	7C	Is R3 ≠ 0, ga dan terug naar H'0010'
0016	74	20	Enable Interrupt (maak II bit "0")
0018	1F	00 00	Terug naar begin

001B	04	00	Tijd-delay-subroutine
001D	05	00	
001F	06	0A	(0A = ca. 5 seconden "delay")
0021	F8	7E	
0023	F9	7C	
0025	FA	7A	
0027	17		Return

Vraagstuk 48

Antwoord 48a

....	75	C0	Maak CC = 00
	72		Lees data (schakelaars) in R2
	40		HALT

Antwoord 48b

Druk de toetsen REG en ENT NXT enkele keren in

.PL = 02 CC = 00

.R3 = 06

Vraagstuk 49

Antwoord 49a

```
....      75  C0      Maak CC = 00
           56  07      Lees data (schakelaars) via poort H'07' in R2
           40          HALT
```

Antwoord 49b

Druk de toetsen REG en ENT NXT enkele keren in

```
.PL = 02      CC = 00
.R3 = 05
```

Vraagstuk 50

Antwoord 50a

```
....      0E  0F  FF  Laad R2 met inhoud van adres H'0FFF'
           40          HALT
```

Antwoord 50b

Druk de toetsen REG en ENT NXT enkele keren in

```
.PL = 00      CC = 00
.R3 = XX      Doet niet ter zake
```

Vraagstuk 51

Laden op de bekende wijze door MEM in te drukken en een adres aan te slaan, b.v. 0100, en dan de data, dus 02, ENT NXT, enz. We krijgen dan:

H'0100' = 02, H'0101' = 06, H'0102' = 05 en H'0103' = 00.

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>
....	20	Maak R0 = 0
	05 04	Laad indexregister R1 met 4
	8D 41 00	Tel bij R0 met autodecrement (i.c.=10, R1) de inhoud op van adres H'0100' e.v.
	59 7B	(BRNR) Is R1 ≠ 0, ga dan weer optellen
	F0	Toon de som (= H'0D') op de data-LED's.
	40	HALT
of anders:	20	
	05 03	Laad R1 met 3
	8D 61 00	Tel op indexed (ic=11, R1) ...
	F9 7B	Spring relatief terug met decrement R1
	F0	Toon R0
	40	HALT

(Men kan natuurlijk ook met auto-increment werken, maar dat vereist een extra vergelijkings-instructie, n.l. COMI,R1 04).

Vraagstuk 52

Verander F0 in D4 07 Bied data (schakelaars) aan via poort H'07'
 40 HALT

Vraagstuk 53

Verander F0 in CC 0F FF Laad som (R0) in adres H'0FFF'
 40 HALT

Vraagstuk 54

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>
...	70	Lees data (schakelaars) in R0
	F4 80	Controleer bit 7 van de data (R0)
	98 7B	Zo niet '1', ga dan terug en lees nieuwe data
	D0	Zo wel '1', accepteer en rotate links R0
	F0	Toon de geroteerde data op de LED's
	40	HALT

Vraagstuk 55

Antwoord 55a

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>
...	54 07	Lees (35) via poort H'07' in R0 (3 5)
	55 07	Lees (47) via poort H'07' in R1 (4 7)
	81	Tel R1 op bij R0, som in R0 (7 12)
	D4 07	Toon som via poort H'07' op data-LED's Som = 7C, dus fout
	40	HALT

Antwoord 55b

Zet tussen 81 en D4 07 de instructie:

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>
....	84 66	Tel op H'66' bij R0 (7 12)
		vóór BCD-telling (6 6)
		carry = 1 $\xrightarrow{\quad}$ +
		som = (14 2)

Het resultaat is dus nog 6 te veel.

Antwoord 55c

Voeg tussen 84 66 en D4 07 de volgende instructie in:

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>	
....	94	Decimaal Adjust R0	(14 2)
			-6

			(8 2)

Het resultaat is nu 82, dus correct.

Vraagstuk 56

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>
....	75 01	Maak Carry-bit = 0
	77 08	Zet With Carry-bit van PSL
	07 02	Aantal bytes per getal (2) in register R3
	05 00	Maak R0 = 0
	0F 41 00	Laad byte van 1ste operand in R0
	84 66	Tel op H'66' voor BCD-optelling
	51	Rotate rechts R1 (interdigit-carry->carry)
	8F 61 02	Tel op byte van 2de operand
	94	Decimal Adjust R0
	CF 61 02	Berg tussenresultaat op ter plaatse van 2de operand
	D1	Rotate links R1
	5B 70	Laad weer byte van 1e operand indien R3 ≠ 0

Nogmaals hierna hetzelfde programma intikken, maar verander:

0F 41 00	in	0F 41 02	(tussenresultaat wordt 1ste operand)
8F 61 02	in	8F 61 04	(tel op 3de operand)
CF 61 02	in	CF 61 00	(berg resultaat op ter plaatse van 1ste operand)

N.B. Het eerste programma deel kan waarschijnlijk compacter worden gemaakt met indexering. (Zoek eventueel zelf een effectievere methode.)

vervolgens:

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>
....	0C 01 00	Laad middelste deel van som in R0
	C3	R0 -> R3
	0C 01 01	Laad laatste deel van som in R0
	C2	R0 -> R2
	B4 80	Controleer SENSE-bit
	98 7C	Indien 0, ga dan "loopen"

F2		Ingedrukt, toon het minst significante deel op LED's
3B	06	Sprong naar tijdsbroutine voor onderscheid met 2 keer indrukken van SENSE-toets.
B4	80	Controleer SENSE
98	7C	Indien 0, ga dan "loopen"
03		Laad R3 in R0
C1		Laad R0 in R1
F1		Ingedrukt, toon het meest significante deel van de som
40		
<hr/>		
....		Zet hier de tijdsbroutine neer (zie oefening 44) met b.v. 06 04 (= ca. 2 seconden "delay").

Vraagstuk 57

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>
....	77 0A	Zet PSL WC-bit en COM (logical comp.)-bit
	75 01	Maak Carry = 0
	0C 01 00	Laad R0 met binair getal (op adres H'0100')
	C1	Laad R0 in R1
	45 0F	Clear meest significante bits van binair getal
	85 66	Tel H'66' op voor BCD-optelling
	95	Decimal Adjust R1 (meest significante deel in R1)
	44 F0	Clear minst significante bits
	E4 10	Vergelijk logisch met H'10'.
	1A 09	Als minst significante bits = 0, spring dan eruit
	A4 0F	Trek af van R0 (H'10'-1)
	85 7B	Tel op bij R1 (H'16' + H'66'-1)
	95	Decimal Adjust R1
	84 00	Tel 0 op
	1B 73	Indien niet 0, ga dan weer vergelijken
<hr/>		
	B4 80	Test SENSE
	98 7C	Indien 0, ga dan "loopen"
	F1	Ingedrukt, toon minst significante deel (R1) op LED's
	3B	Sprong naar tijdsbroutine (zie oefening 56)
	B4 80	Controleer SENSE
	98 7C	
	F0	Ingedrukt, toon meest significante deel (R0) op LED's
	40	HALT

Vraagstuk 58

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>
....	75 09	Clear WC- en C-bit
	07 00	Laad indexregister R3 met 0
	B4 80	Controleer SENSE
	98 7C	Indien 0, ga dan "loopen"
	70	Lees ASCII-teken (schakelaars) in R0
	A4 30	Maak er BCD van (trek H'30' af)
	C1	Laad R0 in R1
	CF 61 00	Berg 1 BCD-teken op in H'0100' e.v. (indexregister=R3, indexed only IC.=11)
	3B 15	Spring naar tijdsroutine om onderscheid te maken tussen 2 x indrukken van SENSE-toets
	B4 80	Controleer SENSE
	98 7C	
	70	Ingedrukt, lees volgende ASCII-teken in R0
	A4 30	Maak er BCD van
	D1	} Schuif over 4 bits naar links (R1) (vorige ASCII teken)
	D1	
	D1	
	D1	
	61	"OR" minst significante deel (R1)
	CF 61 00	Berg 2 BCD-tekens in 1 byte
	3B 03	Spring naar tijdsroutine
	DB 60	Increment R3, lees ander teken indien niet 0
	40	HALT

	04 00	Tijdsroutine
	enz.
	
	17	Return

Vraagstuk 59

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>
....	74 40	FLAG "uit"
	07 00	Maak R3 = 0
	F3	Clear LED's
	B4 80	Controleer SENSE
	98 7C	Indien 0, ga dan "loopen"
	06 0A	Ingedrukt: laad R2 met H'0A' (5 seconden "delay")
	3B 10	Spring naar tijdsroutine

76	40	FLAG "aan"
B4	80	Controleer SENSE
18	06	Ingedrukt: toon tijd in seconden op LED's
<hr/>		
3B	06	Niet ingedrukt: spring naar tijdsbroutine
87	01	1x door tijdsbroutine (1 seconde): verhoog R3 met 1
1B	76	Ga SENSE weer controleren
<hr/>		
F3		Toon tijd in seconden (in R3) op LED's
40		HALT
<hr/>		
06	02	1 seconde "delay"
04	00	Tijdsbroutine
05	00	
05	00	
F8	7E	
F9	7C	
FA	7A	
17		Return

Vraagstuk 60

Antwoord 60a

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>
....	0D 01 00	Laad R1 met 1e operand
	0E 01 01	Laad R2 met 2de operand
	77 08	Zet WC-bit van PSL
	20	Maak R0 = 0
	07 08	Laad R3 met 8
<hr/>		
	75 01	Clear Carry
	F5 01	Controleer bit 1 van inhoud R1
	98 01	Indien 0, geen optelling
	82	Indien 1 (R0 = R0 + R2), wel optelling
	50	Rotate rechts R0
	51	Rotate rechts R1
	FB 75	Decrement R3; spring terug relatief naar Clear Carry als R3 ≠ 0
<hr/>		
	CC 01 00	Berg resultaat R0 op ter plaatse van 1e operand
	CD 01 01	Berg resultaat R1 op ter plaatse van 2de operand
	17	Return

Antwoord 60b

<i>ingedrukte toets</i>	<i>DP</i>	<i>commentaar</i>
MEM	.Ad.=	
0100	.Ad.=0100	Sla aan: 0,1,0,0
ENT NXT	.0100= XX	Verander in b.v. H'12' = 18
18	.0100= 12	
ENT NXT	.0101= XX	Verander in b.v. H'05' = 5
05	.0101= 05	
ENT NXT		

Druk RST in (of druk REG, C en 0 – (zet PC=0) – ENT/NXT in en druk vervolgens RUN in).

Na afloop (doven van de RUN-lampje) druk men in:

<i>ingedrukte toets</i>	<i>DP</i>	<i>commentaar</i>
MEM	.Ad.=	
0100	.Ad.=0100	Sla aan 0,1,0,0
ENT NXT	.0100= 00	Meest significante deel resultaat = 0
ENT NXT	.0101= 5A	Minst significante deel = H'5A' = 90 (=18 × 5)

Antwoord 60c

Maak PC = 0 en druk de STEP-toets in en bekijk de werking: optellen en schuiven volgens de bekende methode!

Vraagstuk 61

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>
0000	3F 00 30	Spring naar delay-subroutine
	B4 80	Test SENSE
	98 7C	Als 0, blijf testen
	54 07	Laad R0 met data-schakelaars
	07 07	Laad R3 met 3

	77 08	Zet WC-bit van PSL
	D0	Roteer R0 links door Carry
	B5 01	TEST Carry-bit
	18 02	Als 1, ga verder
	FB 77	Als 0, roteer verder, decrement R3

	75 08	Clear WC-bit van PSL
	03	} Vermenigvuldig R3 met 3
	D3	
	83	
	C3	

	BF	00	50	Spring naar juiste subroutine
	D5	07		Toon waarde in R1 op LED's
	40			HALT

0030	04	00		Tijd-delay subroutine
	05	00		
	06	02		ca. 1 seconde delay
	F8	7E		
	F9	7C		
	FA	7A		
	17			Return

0050	05	00		Laad R1 met 0 (schakelaar 0)
	17			Return
0053	05	01		Laad R1 met 1 (schakelaar 1)
	17			Return
0056			Enz.

0065	05	07		Laad R1 met 7
	17			Return

Vraagstuk 62

waarde	8	4	2	1		8	4	2	1	
bit-nummer	7	6	5	4		3	2	1	0	
cijfer 0	0	1	1	1		1	1	0	1	= H'7D'
1	0	1	1	0		0 ₍₁₎	0 ₍₁₎	0	0	= H'60' ('0C')
2	0	1	0	1		0	1	1	1	= H'57'
3	0	0	0	1		1	1	1	1	= H'1F'
4	0	0	1	1		1	0	1	0	= H'3A'
5	0	0	1	0		1	1	1	1	= H'2F'
6	0	1	1	0		1	1 ₍₀₎	1	1	= H'6F' ('6B')
7	0	0	0	1		1	1	0	0	= H'1C'
8	0	1	1	1		1	1	1	1	= H'7F'
9	0	0	1	1		1	1	1	1 ₍₀₎	= H'3F' (3E')

Plaats de tabel b.v. op de plaatsen H'0100' e.v. met de snellaadmethode (druk REG en F in, enz.).

Programma

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>
....	B4 80	Controleer SENSE
	98 7C	Indien 0, blijf dan controleren
	3B 0C	Ingedrukt: spring naar delay-loop
	0F 0F FF	Lees getal (schakelaars) in R3
	0F 61 00	Lees in R0 (indexregister=R3) de goede tabelwaarde
	CC 0F FF	Toon deze waarde op de output LED's
	1B 6F	Terug naar begin
	40	HALT
<hr/>		
	06 02	Laad R1 met 2 (1 seconde "delay")
	04 00	
	05 00	
	F8 7E	Delay-lussen.
	F9 7C	Delay-lussen
	FA 7A	Delay-lussen
	17	Return
<hr/>		
0100	7D	DP-segmenten-code voor cijfer 0
0101	60	DP-segmenten-code voor cijfer 1
0102	57	2
0103	1F	3
0104	3A	4
0105	2F	5
0106	6F	6
0107	1C	7
0108	7F	8
0109	3F	9

Vraagstuk 63

Zie oefening 56: het programma is hetzelfde (2 bytes per getal); verander de eerste CF 61 02 in: CF 61 00 (resultaat op plaats van 1ste operand.

laat het stuk ----- Nogmaals ... ----- weg!

Antwoord 63a

Zet de data in 2 bytes in het geheugen, dus 0020 en 0045 enz. op de adressen H'0100' t/m H'0103'.

Antwoord 63b

20 + 45 = 65	Druk MEM en 0100 in: (of evt. REG en 1	.0100= 00 .r1 = 00)	, geen overflow
40 + 88 = 128	” ” ”	.0100 01	, overflow
28 + 46 = 74	” ” ”	.0100= 00	, geen overflow
44 + 98 = 142	” ” ”	.0100= 01	, overflow

Vraagstuk 64

Vooraan toevoegen: 74 40 FLAG “uit”.

Toevoegen vóór HALT:

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>
....	
	59 02	Indien R1 ≠ 0, spring dan 2 verder
	1B 02	Spring naar HALT
	76 40	R0 = 1, zet FLAG “aan”
	40	HALT (of eventueel 17 =RETURN)

Vraagstuk 65

Als oefening 56 met enige wijzigingen:

0000	75 01	
	77 08	
		Aantal bytes per getal is 3
	05 00	
	0F 41 00	
	84 66	
	51	
	8F 61 03	
	94	
	CF 61 03	Berg resultaat op ter plaatse van 2de operand
	D1	
	5B 70	

vervolgens:

	07 03	Laad R3 met 3.
	B4 80	Controleer SENSE-toets
	98 7C	Indien 0, ga dan “loopen”
	0F 41 03	Laad R0 (indexregister=R3, IC=10, autodecrement) met byte van het resultaat
	F0	Toon byte op LED's (NON-EXT)
	3B 03	Spring naar tijdsbroutine

5B	73	Lees volgende byte van resultaat in R0
40		HALT

04	00	Tijds subroutine; voor onderscheid SENSE twee keer indrukken
05	00	
06	04	2 seconden "delay"
F8	7E	
F9	7C	
FA	7A	
17		Return

Druk REG F en 0100 in; vervolgens de waarden 07 83 95 en 08 42 37 en tenslotte ENT/NXT invoeren. Druk dan RST in
De som is: $078395 + 084237 = 162632$.

Dus na 1x SENSE wordt getoond de waarde 32, na 2x de waarde 26 en na 3x de waarde 16.

Vraagstuk 66

Antwoord 66a

Zet de getallen op de adressen H'0100' en H'0101'

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>
....	74 40	FLAG "uit"
	0F 01 00	Laad byte operand 1 en R3
	AF 01 01	Trek operand 2 af
	97	Decimal Adjust R3
	B5 01	Controleer Carry bit
	18 02	Indien 1, ga dan verder
	76 40	Indien 0, zet FLAG dan "aan" (borrow)
	CF 01 00	Berg resultaat op ter plaatse van operand 1
	40	HALT

Antwoord 66b

Druk MON en RST in; vervolgens MEM en sla dan 0, 1, 0, 0 aan:

.0100= 24	(40 - 16 = 24)	FLAG "uit"
.0100= 16	(48 - 32 = 16)	FLAG "uit"
.0100= 76	(50 - 74 = (100) - 24 = 76)	FLAG "aan"
.0100= 94	(68 - 74 = (100) - 6 = 94)	FLAG "aan"

Vraagstuk 67

Antwoord 67a

Zet de getallen op de plaatsen H'0100' t/m H'0103'

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>
....	74 40	Maak FLAG "uit"
	77 09	Zet WC- en C-bit van PSL
	07 02	Aantal bytes per getal (2) in R3
	0F 41 00	Laad R0 met byte 1e operand (indexregister=R3, IC=10 auto-decrement)
	AF 61 02	Trek byte 2de operand af (i.r.=R3)
	94	Decimal Adjust R0
	5B 06	Skip Test C indien dit niet meest significante byte is
	B5 01	Controleer Carry-bit
	18 02	Indien 1, ga dan verder
	76 40	Indien 0, zet dan FLAG "aan" (borrow)
	CF 61 00	Berg resultaat op ter plaatse van 1e operand
	5B 6C	Indien R3 niet 0 is, laad dan weer byte 1e operand
	40	HALT

Antwoord 67b

Druk MON en RST in, vervolgens MEM en sla dan 0, 1, 0, 0 aan:

.0100= 07

ENT NXT

.0101= 61 FLAG "uit" ($7439 - 6678 = 0761$)

.0100= 92

ENT NXT

.0101= 39 FLAG "aan" ($6678 - 7439 = (10000) - 1239 = 9239$).

Vraagstuk 68

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>
0000	20	Maak R0 = 0.
	CC 00 40	variabele A = 0
	CC 00 41	„ B = 0
	CC 00 42	„ C = 0
	CC 00 43	„ D = 0
000D	F0	(Begin) Toon waarde variabele op LED's
	B4 80	Controleer SENSE op 1.
	1A 7C	Indien niet 1, blijf dan controleren
	3F 01 20	Spring naar delay-subroutine (20 ms)

73			Lees optellen of aftrekken (schakelaar)
47	01		Schrap overbodige bits
03			Laad R3 in R0
D0			Roteer links
83			Tel R3 op
C3			R0 → R3 (er is dus vermenigvuldigd met 3)
BF	00	50	Spring naar optellen of aftrekken
B4	80		Controleer SENSE op '0'
18	7C		Indien '1', blijf dan controleren
3F	01	20	Spring naar delay-subroutine (20 ms)
0C	00	40	Laad variabele A in R0
44	80		Neem bit 7
0D	00	41	Laad variabele B in R1
45	40		Neem bit 6
21			Zet in R0
0D	00	42	Laad variabele C in R1
45	20		Neem bit 5
21			Zet in R0
0D	00	43	Laad variabele D in R1
45	10		Neem bit 4
21			Zet in R0
1F	00	0D	Spring terug naar begin

0040			Hier staat variabele	A
0041			” ” ”	B
0042			” ” ”	C
0043			” ” ”	D
0044			” ” ”	MB
0045			” ” ”	MC
0046			” ” ”	MD

0050	1F	00	C0	Spring naar aftrekken
	0C	00	40	Vorm in R0: $(A \oplus B \oplus C)' = MD$
	2C	00	41	
	2C	00	42	
	24	FF		
	CC	00	46	Berg MD op
	0C	00	40	Vorm in R0: $D(A \oplus B)' + \bar{C}\bar{D} = MC$

	2C	00	41	
	24	FF		
	4C	00	43	
	0D	00	43	
	25	FF		
	4D	00	42	
	61			
	CC	00	45	Berg MC op
	0C	00	42	Vorm in R0: $B(\bar{C} + D) + \bar{A}C\bar{D} = MB$
	24	FF		
	6C	00	43	
	4C	00	41	
	0D	00	43	
	6D	00	40	
	25	FF		
	4D	00	42	
	61			
	CC	00	44	Berg MB op
	0C	00	43	Vorm $A(C + D) + B\bar{C}\bar{D} = A$
	6C	00	42	
	4C	00	40	
	0D	00	42	
	6D	00	43	
	25	FF		
	4D	00	41	
	61			
00A7	CC	00	40	Berg A op
	0C	00	44	Maak B = MB
	CC	00	41	
	0C	00	45	Maak C = MC
	CC	00	42	
	0C	00	46	
	CC	00	43	Maak D = MD
	17			Keer terug naar hoofdprogramma
00C0	0C	00	40	Vorm $(A \oplus B \oplus C) = MD$
	2C	00	41	
	2C	00	42	

CC	00	46	Berg MD op
0C	00	40	Vorm $D(A \oplus B) + C\bar{D} = MC$
2C	00	41	
4C	00	43	
0D	00	43	
25	FF		
4D	00	42	
61			
CC	00	45	Berg MC op
0C	00	42	Vorm $B(\bar{C} + D) + AC\bar{D} = MB$
24	FF		
6C	00	43	
4C	00	41	
0D	00	43	
25	FF		
4D	00	42	
4D	00	40	
61			
CC	00	44	Berg MB op
0C	00	43	Vorm A opnieuw
6C	00	42	
4C	00	40	
0D	00	42	
6D	00	43	
6D	00	41	
25	FF		
61			
1F	00	A7	Spring naar H'00A7' (transport)

Tijd-delay-subroutine (20 ms):

0120	04	0A	Laad R0 met 10
	05	C8	Laad R1 met 200
	F9	7C	
	F8	7A	
	17		Return

Vraagstuk 69

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>
0000	20	Maak R0 = 0
	CC 00 20	Variabele = R0
	F0	(Begin) Toon waarde op I/O LED's
	B4 80	Controleer SENSE op 1
	1A 7C	
	3F 01 20	Spring naar delay subroutine (20 ms)
	73	Lees (schakelaars) optellen en aftrekken
	47 01	Schrap overbodige bits
	03	} Vermenigvuldig R0 met 3
	D0	
	83	
	C3	Vermenigvuldiging klaar
	BF 00 50	Spring naar optellen of aftrekken
	B4 80	Controleer SENSE op 0
	18 7C	Indien 1, blijf dan controleren
	3F 01 20	Spring naar delay-subroutine
	1F 00 04	Terug naar begin
<hr/>		
0020		Hier staat waarde van de variabele
<hr/>		
0050	1F 00 61	Spring naar aftrekken
	0C 00 20	Optellen
	84 10	
0058	CC 00 20	(Conversie) Berg variabele op
	C1	Kopie van R0 in R1
	51	Schuif R1
	21	$R1 \oplus R0$
	44 F0	Verwijder overbodige bits
	17	Keer terug naar hoofdprogramma
	0C 00 20	Aftrekken
	A4 10	
	1F 00 58	Spring naar Conversie

Tijd-delay-subroutine:

<i>adres</i>	<i>instructie</i>	<i>commentaar</i>
0120	05 0A	Laad R1 met 10
	06 C8	Laad R2 met 300
	FA 7E	
	F9 7A	
17		Return

Electronic components and materials

for professional, industrial
and consumer uses

from the world-wide
Philips Group of Companies



- Argentina:** FAPESA I.y.C., Av. Crovara 2550, Tablada, Prov. de BUENOS AIRES, Tel. 652-7438/7478.
- Australia:** PHILIPS INDUSTRIES HOLDINGS LTD., Elcoma Division, 67 Mars Road, LANE COVE, 2066, N.S.W., Tel. 427 08 88.
- Austria:** ÖSTERREICHISCHE PHILIPS BAUELEMENTE Industrie G.m.b.H., Triester Str. 64, A-1101 WIEN, Tel. 62 91 11.
- Belgium:** M.B.L.E., 80, rue des Deux Gares, B-1070 BRUXELLES, Tel. 523 00 00.
- Brazil:** IBRAPE, Caixa Postal 7383, Av. Brigadeiro Faria Lima, 1735 SAO PAULO, SP, Tel. (011) 211-2600.
- Canada:** PHILIPS ELECTRONICS LTD., Electron Devices Div., 601 Milner Ave., SCARBOROUGH, Ontario, M1B 1M8, Tel. 292-5161.
- Chile:** PHILIPS CHILENA S.A., Av. Santa Maria 0760, SANTIAGO, Tel. 39-40 01.
- Colombia:** SADAPE S.A., P.O. Box 9805, Calle 13, No. 51 + 39, BOGOTA D.E. 1., Tel. 600 600.
- Denmark:** MINIWATT A/S, Emdrupvej 115A, DK-2400 KØBENHAVN NV., Tel. (01) 69 16 22.
- Finland:** OY PHILIPS AB, Elcoma Division, Kaivokatu 8, SF-00100 HELSINKI 10, Tel. 1 72 71.
- France:** R.T.C. LA RADIOTECHNIQUE-COMPELEC, 130 Avenue Ledru Rollin, F-75540 PARIS 11, Tel. 355-44-99.
- Germany:** VALVO, UB Bauelemente der Philips G.m.b.H., Valvo Haus, Burchardstrasse 19, D-2 HAMBURG 1, Tel. (040) 3296-1.
- Greece:** PHILIPS S.A. HELLENIQUE, Elcoma Division, 52, Av. Syngrou, ATHENS, Tel. 915 311.
- Hong Kong:** PHILIPS HONG KONG LTD., Elcoma Div., 15/F Philips Ind. Bldg., 24-28 Kung Yip St., KWAI CHUNG, Tel. NT 24 51 21.
- India:** PEICO ELECTRONICS & ELECTRICALS LTD., Ramon House, 169 Backbay Reclamation, BOMBAY 400020, Tel. 295144.
- Indonesia:** P.T. PHILIPS-RALIN ELECTRONICS, Elcoma Division, 'Timah' Building, Jl. Jen. Gatot Subroto, P.O. Box 220, JAKARTA, Tel. 44 163.
- Ireland:** PHILIPS ELECTRICAL (IRELAND) LTD., Newstead, Clonskeagh, DUBLIN 14, Tel. 69 33 55.
- Italy:** PHILIPS S.p.A., Sezione Elcoma, Piazza IV Novembre 3, I-20124 MILANO, Tel. 2-6994.
- Japan:** NIHON PHILIPS CORP., Shuwa Shinagawa Bldg., 26-33 Takanawa 3-chome, Minato-ku, TOKYO (108), Tel. 448-5611.
(IC Products) SIGNETICS JAPAN, LTD, TOKYO, Tel. (03)230-1521.
- Korea:** PHILIPS ELECTRONICS (KOREA) LTD., Elcoma Div., Philips House, 260-199 Itaewon-dong, Yongsan-ku, C.P.O. Box 3680, SEOUL, Tel. 794-4202.
- Malaysia:** PHILIPS MALAYSIA SDN. BERHAD, Lot 2, Jalan 222, Section 14, Petaling Jaya, P.O.B. 2163, KUALA LUMPUR, Selangor, Tel. 77 44 11.
- Mexico:** ELECTRONICA S.A. de C.V., Varsovia No. 36, MEXICO 6, D.F., Tel. 533-11-80.
- Netherlands:** PHILIPS NEDERLAND B.V., Afd. Elonco, Boschdijk 525, 5600 PB EINDHOVEN, Tel. (040) 79 33 33.
- New Zealand:** PHILIPS ELECTRICAL IND. LTD., Elcoma Division, 2 Wagener Place, St. Lukes, AUCKLAND, Tel. 867 119.
- Norway:** NORSK A/S PHILIPS, Electronica, Sørkedalsveien 6, OSLO 3, Tel. 46 38 90.
- Peru:** CADESA, Rocca de Vergallo 247, LIMA 17, Tel. 62 85 99.
- Philippines:** PHILIPS INDUSTRIAL DEV. INC., 2246 Pasong Tamo, P.O. Box 911, Makati Comm. Centre, MAKATI-RIZAL 3116, Tel. 86-89-51 to 59.
- Portugal:** PHILIPS PORTUGESA S.A.R.L., Av. Eng. Duharte Pacheco 6, LISBOA 1, Tel. 68 31 21.
- Singapore:** PHILIPS PROJECT DEV. (Singapore) PTE LTD., Elcoma Div., P.O.B. 340, Toa Payoh CPO, Lorong 1, Toa Payoh, SINGAPORE 12, Tel. 53 88 11.
- South Africa:** EDAC (Pty.) Ltd., 3rd Floor Rainer House, Upper Railway Rd. & Ove St., New Doornfontein, JOHANNESBURG 2001, Tel. 614-2362/9.
- Spain:** COPRESA S.A., Balmes 22, BARCELONA 7, Tel. 301 63 12.
- Sweden:** A.B. ELCOMA, Lidingövägen 50, S-115 84 STOCKHOLM 27, Tel. 08/67 97 80.
- Switzerland:** PHILIPS A.G., Elcoma Dept., Allmendstrasse 140-142, CH-8027 ZÜRICH, Tel. 01/43 22 11.
- Taiwan:** PHILIPS TAIWAN LTD., 3rd Fl., San Min Building, 57-1, Chung Shan N. Rd, Section 2, P.O. Box 22978, TAIPEI, Tel. 5513101-5.
- Thailand:** PHILIPS ELECTRICAL CO. OF THAILAND LTD., 283 Silom Road, P.O. Box 961, BANGKOK, Tel. 233-6330-9.
- Turkey:** TÜRK PHILIPS TICARET A.S., EMET Department, Inonu Cad. No. 78-80, ISTANBUL, Tel. 43 59 10.
- United Kingdom:** MULLARD LTD., Mullard House, Torrington Place, LONDON WC1E 7HD, Tel. 01-580 6633.
- United States:** (Active devices & Materials) AMPEREX SALES CORP., Providence Pike, SLATERSVILLE, R.I. 02876, Tel. (401) 762-9000.
(Passive devices) MEPCO/ELECTRA INC., Columbia Rd., MORRISTOWN, N.J. 07960, Tel. (201) 539-2000.
(IC Products) SIGNETICS CORPORATION, 811 East Argues Avenue, SUNNYVALE, California 94086, Tel. (408) 739-7700.
- Uruguay:** LUZILECTRON S.A., Rondeau 1567, piso 5, MONTEVIDEO, Tel. 9 43 21.
- Venezuela:** IND. VENEZOLANAS PHILIPS S.A., Elcoma Dept., A. Ppal de los Ruices, Edif. Centro Colgate, CARACAS, Tel. 36 05 11.