

Notulen van de vergadering der Z8-commissie op 31 oktober 1963.

Aanwezigen: HH. Kruseman Aretz, v.d.Meulen, van der Poel, Schmidt,  
van der Sluis, van Wijngaarden en Zwanenburg.

Deze notulen geven een recapitulatie van de geboekte resultaten van de Z8 besprekingen, welke gewijd waren aan de handcode van de EL-X8. Met handcode wordt bedoeld een code op het 1e niveau, zoals beschreven in Z8 nr. 2, met dien verstande dat reeds op dit niveau elementen worden geïntroduceerd, die - naar verwacht wordt-deel uit zullen maken van een hogere niveau-code.

De handcode houdt zich niet alleen bezig met de wijze, waarop opdrachten moeten worden genoteerd, maar ook met de regels voor beschrijving van andere informatie. De tot nog toe behandelde soorten informaties zijn:

- 1e opdrachten;
- 2e getallen;
- 3e labels;
- 4e commentaar.

In het hierbijgevoegde overzicht I wordt de syntax van de handcode gedetailleerd beschreven.

Overzicht II bevat een complete lijst van opdrachten, beschreven in de overzicht I behandelde code, en van de interne representatie.

\* volgt nog

OVERZICHT IHandcode EL-X8Symbolenset.

De code zal gebruik maken van de volgende symbolen:

$\langle \text{symbol} \rangle ::= \langle \text{letter} \rangle | \langle \text{cijfer} \rangle | \langle \text{delimiter} \rangle$

$\langle \text{letter} \rangle ::= A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z$

Opm. In plaats van hoofdletters mogen ook kleine letters worden gebruikt. In de code zal er nimmer gebruik gemaakt worden van het verschil tussen hoofdletters en kleine letters.

$\langle \text{cijfer} \rangle ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9$

$\langle \text{delimiter} \rangle ::= \langle \text{operator} \rangle \langle \text{separato} \rangle \langle \text{haakje} \rangle$

$\langle \text{operator} \rangle ::= + | - | * | / | '+' | '*' | =$

$\langle \text{separato} \rangle ::= ' | ; | 'NL' | , | : | . |_{10}$

Opm. Het symbool 'NL' vertegenwoordigt de overgang op een nieuwe regel.

Het symbool  $_{10}$  mag vervangen worden door de letter E, in de hardware representatie hoeft het dus niet aanwezig te zijn.

$\langle \text{haakje} \rangle ::= [ | ] | ( | )$

Opm. In de code zullen [ en ] resp. ) dezelfde betekenis hebben, zodat steeds een stel haakjes aanwezig hoeft te zijn.

Spaties hebben geen betekenis.

OPDRACHTEN NOTATIE

a. over de operanden

$\langle n \rangle ::= 0 | 1 | \dots | 32767$

$\langle m \rangle ::= -16384 | -16383 | \dots | 16383$

$\langle q \rangle ::= -256 | -255 | \dots | 255$

$\langle r \rangle ::= 0 | 1 | \dots | 57$

$\langle i \rangle ::= 0 | 1 | \dots | 7$

$\langle j \rangle ::= 0 | 1 | \dots | 15$

$\langle s \rangle ::= 0 | 1 | \dots | 31$

$\langle n_2 \rangle ::= -32767 | -32766 | \dots | 32767$

$\langle q_2 \rangle ::= -511 | -510 | \dots | 511$

<naam> ::= <letter> | <naam> <letter> | <naam> <cijfer>  
<statnaam> ::= <naam>  
<dynnaam> ::= <naam>  
<statvar> ::= <statnaam> | <statnaam>[<n2>]  
<dynvar> ::= <dynnaam> | <dynnaam>[<q2>]  
<modnaam> ::= M<r> | MA | MC | MG | MS | MT | MD  
<statbvar> ::= <statnaam> [B] | <statnaam> [B + <n>]  
<stat> ::= M [<n>] | C [<i>] | K [<j>] | D | <statvar>  
<dyn> ::= <modnaam> | <modnaam> [<q>] | <dynvar>  
<statb> ::= M [B] | M [B + <m>] | <statbvar>  
<reg> ::= A | B | F | G | S  
<statr> ::= <stat> | <reg>  
<variabele operand> ::= <reg> | <stat> | <statb> | <dyn>

b. over de opdrachten

<opdracht> ::= <aritmetische opdracht> | <niet aritmetische opdracht>  
<conditievolging> ::= | U, | Y, | N,  
<conditiezetting> ::= | P | , Z | , E

Aritmetische opdrachten:

<aritmetische opdracht> ::=  
    <conditievolging> <doeloperand> <operator> <facteker>  
                        <bronoperand> <conditiezetting>  
    <doeloperand> ::= AS | <variabele operand>  
    <facteker> ::= | + | -  
    <bronoperand> ::= <variabele operand> | : <stat> | : <dyn> | <n>

Opm. De combinering van mogelijke substituties tot één opdracht is aan bepaalde regels gebonden. Niet alle combinaties zijn namelijk toegestaan. Men raadplege hiervoor Overzicht II.

### Niet aritmetische opdrachten

```

<niet aritmetische opdracht> ::= < sprongopdracht > | < doe-opdracht >
                                         | < schuifopdracht > | < normeeropdracht > |
                                         | < transportopdracht > | < maaltienopdracht > | < stopopdracht >

< sprongopdracht > ::= < conditievolging > < sprong >
< sprong > ::= < directe sprong > | < additieve sprong > | < tellende sprong >
                                         | < vaste subr. sprong > | stapelende subr. sprong

< directe sprong > ::= < directe sprongnaam > (< bronoperand >)

< directe sprongnaam > ::= GOTO | GOTOR

< additieve sprong > ::= < additieve sprongnaam > (< facteken > < bronoperand >)

< additieve sprongnaam > ::= JUMP | JUMPR

< tellende sprong > ::= < telnaam > ( : < stat > , < i > )

< telnaam > ::= REP | REPP | REPZ | REPE

< vaste subr. sprong > ::= SUB ( : < stat > , < j > )

< stapelende subr. sprong > ::= SUBC (< bronoperand >) | SUBCD (< bronoperand >)

```

<doeopdracht> ::= <doenaam>(<variabele\_operand>)

<doenaam> ::= DO | DOS

<schuifnaam> :: = <schuifrichting> <schuifwijze> <schuifcircuit>

<schuifrichting> :: = L | R

<schuifwijze> ::= U | C

<schuifcircuit> :: = A | S | AS | SA

$\langle \text{schuifaantal} \rangle ::= \langle s \rangle \mid B + \langle s \rangle$

<normmeeropdracht> ::= <conditievolging> <normnaam> <conditiezetting>

< normeernaam > ::= NOR < normeercircuit >

< normeercircuit>      :: = A | S | AS

<transportopdracht> ::= <conditievolging> <transportnaam> <conditiezetting>

<transportnaam> ::= <register> <transportteken> <register>

<register> ::= A | S | B

<transportteken> ::= | M

<maaltienopdracht> ::= <conditievolging> <maaltiennaam> <conditiezetting>

<maaltiennaam> ::= TENA | TENAS

<stopopdracht> ::= <conditievolging> STOP.

c. opdrachten-separatoren

<opdrachtenseparator> ::= ; | 'NL'

d. Toelichting opdrachtennotatie.

De notatie van opdrachten onderscheidt 2 vormen:

Bij de eerste wordt de opdracht geschreven in de gedaante van een aritm. expressie, waarin 2 operanden voorkomen, gescheiden door een operator.

In deze vorm worden alle aritmatische operaties, zowel in vaste komma als drijkende komma-notatie, uitgedrukt. Als aritmatische operaties worden ook opgevat de logische optelling (operator '+') en de logische vermenigvuldiging (operator '\*') alsmede vervanging (uitgedrukt door het symbool =). Het resultaat van de operatie is tevens de nieuwe waarde van de eerstgenoemde van de 2 operanden.

De tweede vorm van operatie in de procedure. De operatie, welke moet plaatsvinden, wordt met een naam aangegeven.

Indien de naam niet voldoende tot uitdrukking brengt op welke operand of operanden de operatie moet worden uitgevoerd, worden deze door haakjes omsloten, als parameters achter de procedurenaam aangegeven.

Tot de groep van operanden, die op deze wijze worden geschreven, behoren alle sprong-opdrachten en de schuif- en normmeeropdrachten.

Met de introductie van de notatie van opdrachten in de gedaante van procedures is tevens de basis gelegd voor de uitbreiding van de code met macro-opdrachten. Deze kunnen n.l. in dezelfde vorm worden gegoten. Het enige verschil met de op deze wijze genoteerde machine-opdrachten is, dat de laatste niet gedeclareerd behoeven te worden.

Een aantal opmerkingen over detailpunten.

1. Met G wordt bedoeld het F-register, indien de bij de opdracht betrokken 2e operand een enkel-lengte woord is.

De operaties met G als doeloperand functioneren als de overeenkomstige operatie met F. De een-woords geheugen-operand wordt voor de bewerking aangevuld gedacht tot een 2-woords operand waarin de kop nul is.

De operaties met G als bronoperand schrijven de staart van het F-register - al of niet met tekenwisseling - in één geheugenwoord.

Bij operaties in het F-register verwijst het in de opdracht genoemde adres van de geheugenoperand naar de kop:

F = M (n) brengt dus M (n) en M (n + 1) naar het F-register;

G = M (n) brengt M (n) naar de staart van het F-register en brengt 0 naar de kop.

2. D is de inhoud van adres 63.

3. C (i) is identiek met M (i), en wordt alleen zo genoemd indien in gebruik als teladres bij de tellende sprong.

4. K (j) is identiek met M (8 + g), het krijgt deze speciale naam in verband met zijn functie bij subr. sprongen.

De opsomming van

de niet-aritmetische opdrachten begint met een overzicht van de sprong-opdrachten.

De genoemde sprong-functies kunnen als volgt worden beschreven:

|       |   |
|-------|---|
| GOTO  | : De besturing wordt verplaatst naar het adres aangegeven door de in de parameterlijst genoemde operand;          |
| GOTOR | : Als GOTO; als nevenfunctie worden de kleine registers hersteld;   |
| JUMP  | : } De besturing wordt over zoveel plaatsen t.o.v. de   |
| JUMPR | : } lexicografisch volgende opdracht verplaatst, als is aangegeven door de in de parameterlijst genoemde operand; |
| REP   | : } Tellende sprongen. De eerste verplaatst de besturing altijd.  |
| REPP  | : } ongeacht de waarde van de telling, de 2e, 3e en 4e  |
| REPZ  | : } alleen, indien resp. voldaan is aan de relatie  |
| REPE  | : } telling > 0, telling = 0 en telling > - 1.  |
| SUB   | : Subroutinesprong: het adres van de lexicografisch volgende opdracht wordt gered in K[ j ].                      |
| SUBC  | : Stapelende subroutinesprong.  |
| DO    | : Executieopdracht.   |
| DOS   | : idem , het adres van de operand opgegeven in de parameterlijst wordt in S gedumpt.                              |

Vervolgens de schuif- en normmeeropdrachten:

De functienamen van de schuifopdrachten zijn als volgt te verklaren:

1e letter : L voor schuifrichting links en R voor schuifrichting rechts;

2e letter : U voor schuifoperaties, waarbij informatie wordt uitgeschoven en C voor rondschuifopdrachten;

3e en evt.

4e letter : circuit, waarin geschoven wordt: A, S, AS of SA.

De normeeropdrachten NORA, NORAS en NORAS voor normering in de circuits A, S en AS.

De snelle 10-vermenigvuldigingen in S resp. AS worden tot stand gebracht door de opdrachten TENS resp. TENA.

### NOTATIE VAN GETALLEN.

#### a. octale getallen.

<octaal getal> ::= '<0C> <0C> <0C><0C><0C><0C> <0C> <0C><0C>'  
<0C> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7

De faciliteit om octale getallen te kunnen invoeren, is uitsluitend bedoeld om een geheugenwoord een bit voor bit gespecificeerde inhoud te geven.

#### b. decimale getallen.

We onderscheiden getallen van enkele woordlengte en van dubbele woordlengte. Tot de eerste categorie behoren de vaste-kommagetallen (gehele getallen en breuken); tot de 2e de drijvende-kommagetallen.

<facteken> ::= | + | -  
<ongetekend geheel getal> ::= <cijfer>|<ongetekend geheel getal><cijfer>  
<geheel getal> ::= <facteker> <ongetekend geheel getal>  
<breuk> ::= <facteken> . <ongetekend geheel getal>  
<vast getal> ::= <geheel getal>|<breuk> |  
                          <geheel getal> . <ongetekend geheel getal>

<drijvende komma-getal> ::= <vast getal> <sub>10</sub> <geheel getal>

Opm.: Zoals reeds vermeld, mag het symbool <sub>10</sub> worden vervangen door E.

### NOTATIE VAN LABELS.

Als eerste stap op de weg naar een code op het 2e niveau wordt de mogelijkheid ge-introduceerd aan geheugeninhouden een naam te geven. De naamgeving is gehouden aan de regel:

<naam> ::= <letter>|<naam> <letter>|<naam> <cijfer>

Het adres van de met <naam> geïdentificeerde geheugenwaarde wordt aangegeven met: <naam> .

De naamgeving geschiedt door vóór de representatie van de betrokken geheugenplaats de naam te vermelden. (Bij opdrachten dus vóór de opdracht).

### COMMENTAAR.

Toelichtende tekst bij een in een handcode geschreven programma is mogelijk. Dit soort teksten, samengevat onder de naam "comment" wordt door de assembler geregeld, mits het voldoet aan de volgende definitie.

<comment> ::= " <elke reeks van symbolen, met uitzondering van ;> ;  
(Met " wordt bedoeld de opeenvolging van 2 symbolen ' en niet het aanhalingssteken.)

## Overzicht II

Dit overzicht bevat in tabelvorm de opdrachtenset van de X 8; de set is compleet op enkele opdrachten na, deze betreffen voornamelijk de in- en uitvoer. In de tabellen is van iedere opdracht ook de binaire representatie van het functie-deel (bit 15 - 26) weergegeven; de informatie hierover is te vinden en de kop en in de eerste kolom. De elementen, waaruit door samenvoeging een opdracht kan worden geformeerd, bevinden zich onder de horizontale en rechts van de vertikale dubbele lijn. Hoe de vorenbedoelde samenvoeging moet gebeuren, wordt hieronder nadere uiteengezet.

In de tabel zijn de successieve kolommen, in groepen ingedeeld, op blad 1 b.v. drie stuks, n.l. de groepen "conditievolging", "kern van de opdracht" en "conditiezetting"; elk van deze groepen bestaat uit 4 kolommen. Op de andere bladen komt een iets andere groepindeling voor. De vertikale lijnen, welke de groepen van elkaar scheiden, zijn herkenbaar aan het feit, dat ze tot boven in de kop doorlopen.

Men kan nu een opdracht samenstellen door van de kolomelementen op één regel er van links naar rechts gaand uit elke groep één te kiezen en de gekozen elementen in de aangegeven volgorde aanéén te schrijven.

### De operanden

De aanduidingen STAT, STATR, STABT en DYN zijn algemene benamingen voor operanden van verschillende aard. In onderstaande tabel is aangegeven welke operandvorm er voor de algemene benamingen kan worden gesubstitueerd.

Met vormen : STAT en : DYN worden bedoeld de absolute adressen van de met STAT resp. DYN aangegeven variabelen. Onder : STAT zijn ook als operand opgenomen de gehele getallen 0 t/m 32767.

| STAT   | STATR                                  | STABT               | DYN   |
|--|--|---------------------|---|
| M[n]<br>C[i]<br>K[j]<br>D                                  | STAT<br>F<br>G<br>A<br>S<br>B<br>M[62] | M[B]<br>M[B+m]      | Mr[q]<br>MA[q]<br>MC[q]<br>MG[q]<br>MS[q]<br>MT[q]<br>MD[q]                           |
| 0 ≤ n ≤ 56 of<br>63 ≤ n ≤ 32767<br>0 ≤ i ≤ 7<br>0 ≤ j ≤ 15 |  | - 16384 ≤ m ≤ 16383 | - 256 < q < 255, 0 ≤ r ≤ 57<br>De indicering [q] mag voor<br>q = 0 achterwege blijven |

Onder STAT en DYN horen ook thuis de variabelen, die in de handcode met namen, al of niet voorzien van een index, worden aangegeven. Het inleesprogramma zal op een desbetreffende indicatie in het programma de naam vervangen door één van de onder STAT of DYN vermelde operandvormen.

Als STABT kan ook voorkomen een naam gevolgd door de indicering [B] of [B+n<sup>2</sup>] (-32767 ≤ n<sup>2</sup> ≤ 32767): De naam moet dan als een van de onder STAT genoemde vormen gedeclareerd zijn.

### Opmerking:

De adressen 57 t/m 62 zijn bijzondere adressen. Uit-opdrachten - dit zijn opdrachten, waarbij het resultaat (zonder evt. gebruik van de U-variant) in het geheugen geschreven wordt - met deze adressen zijn verboden. Er kan dus nooit iets in deze plaatsen opgeborgen worden.

Bij in-opdrachten (resultaat in register) mogen deze adressen wel gebruikt worden. Als operand fungeert dan niet de inhoud van een geheugenplaats maar de inhoud van een register en wel

|          |   |                                 |
|----------|---|---------------------------------|
| adres 57 | : | kop F-register                  |
| 58       | : | staart F-register               |
| 59       | : | A-register                      |
| 60       | : | S-register                      |
| 61       | : | B-register                      |
| 62       | : | T-register + een bits registers |

Bij de adressering volgens STATB of DYN moet men steeds in het oog houden, dat het uiteindelijk geldende adres, indien het een uit-opdracht betreft, niet een van de waarden 57 t/m 62 kan aannemen. Bij gebruik van operanden van het type STATR (alleen aan te wenden bij in-opdrachten) zijn deze bijzondere plaatsen met de corresponderende registernamen adresseerbaar.

#### De adresbits (0 - 14)

De bits 0 - 14 specificeren het adres van de bij de opdracht betrokken geheugen operand. We onderscheiden 3 gevallen:

- 1e. de bits 0 - 14 bevatten in binaire vorm het adres van de operand.  
Dit is het geval bij de adresvormen STAT, STATR en : STAT. (De register-vorm van STATR n.l. F, G, A, S, B worden gerepresenteerd door de getallen 57, 58, 59, 60, 61);
- 2e. het adres van de operand wordt verkregen door het binaire getal in de bits 0 - 14 te verhogen met B en te verlagen met 16384. Dit geschiedt als een operand van het type STABT wordt gebruikt;
- 3e. het adres wordt afgeleid uit de binaire waarde van de bits 9 - 14 (= p) en uit die van de bits 0 - 8 (= q). Het type van de operand is DYN of : DYN.  
Het adres wordt gevormd door de inhoud van een door q aangewezen geheugenplaats of register te verhogen met q en te verlagen met 256. De bedoelde geheugenplaatsen of registers zijn  

|                           |   |                           |
|---------------------------|---|---------------------------|
| voor p = r (0    r    57) | : | adres D + r               |
| p = 58                    | : | staart F-register         |
| p = 59                    | : | register A                |
| p = 60                    | : | "      S                  |
| p = 61                    | : | "      B                  |
| p = 62                    | : | "      T (opdrachtteller) |
| p = 63                    | : | adres 63.                 |

EL X8

## Interne codering van opdrachten

88

Interne codering van opdrachten

|  |   |   |  |   |   |  |  |  |  |  |  |   |     |
|--|---|---|--|---|---|--|--|--|--|--|--|---|-----|
| bits 26-25<br>(Functiedeel)                | bits 24-23-22-21<br>(conditievolgning)      | bits 16, 15<br>(conditievolgning)           | bits 20, 19<br>(kern van de opdracht)  |   |   |  | bits 18, 17<br>(conditiezetting)                             |  |  |  |  |   |     |
|  |   |   | 00   | 01  | 10  | 11   | 00   | 01   | 10   | 11   | 00   | 01  | 10  |
| Aritmetiek vaste komma                     |   |   |  |   |   |  |  |  |  |  |  |   |     |
| 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0              | U, U | N, N | B + STATR<br>B - STATR<br>B = STATR<br>B = - STATR<br>STAT + B<br>STAT - B<br>STAT = B<br>STAT = - B | Y, Y | N, N | B + STAT<br>B - STAT<br>B = STAT<br>B = - STAT<br>STAT + B<br>STAT - B<br>STAT = B<br>STAT = - B | DYN<br>DYN<br>DYN<br>DYN<br>DYN<br>DYN<br>DYN<br>DYN         | B + STATB<br>B - STATB<br>B = STATB<br>B = - STATB<br>STATB + B<br>STATB - B<br>STATB = B<br>STATB = - B | E, E                    | P, P                  | Z, Z                  | E, E     |     |
| Sporen-opdrachten<br>en execute opdrachten |   |   |  |   |   |  |  |  |  |  |  |   |     |
| 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0              | U, U | N, N | JUMP<br>JUMPR<br>GOTO<br>GOTOR   | Y, Y | N, N | { STATR<br>- STATR<br>STATR<br>?   | { STAT<br>- STAT<br>STAT<br>?                                | { STATB<br>- STATB<br>STATB<br>?   | { DYN<br>- DYN<br>DYN<br>?                                     | { DYN<br>- DYN<br>DYN<br>?                                   | { DYN<br>- DYN<br>DYN<br>?                                   | ?   |     |
| bit 17<br>(functie)                        |   |   |  |   |   |  |  |  |  |  |  |   |     |
| 0  | 1   | 00  | 01   | 10  | 11  | 00   | 01   | 10   | 11   | 00   | 01   | 10  | 11  |
| JUMP<br>JUMPR<br>GOTO<br>GOTOR             |   |   |  |   |   |  |  |  |  |  |  |   |     |
| bit 18, 17<br>(functie)                    |   |   |  |   |   |  |  |  |  |  |  |   |     |
| 00   | 01  | 10  | 11   | 00  | 01  | 10   | 11   | 00   | 01   | 10   | 01   | 10  | 11  |
| REP<br>REP                                 | REPP<br>REPP                                | REPE<br>REPE                                | REPE<br>REPE   | (:STAT,0)<br>(:STAT,4)                      | (:STAT,1)<br>(:STAT,5)                      | (:STAT,2)<br>(:STAT,6)   | (:STAT,3)<br>(:STAT,7)                                       |  |  |  |  |   |     |
| bit 19<br>(parameters)                     |   |   |  |   |   |  |  |  |  |  |  |   |     |
| 0  | 1   | 00  | 01   | 10  | 11  | 00   | 01   | 10   | 11   | 00   | 01   | 10  | 11  |
| JUMP<br>JUMPR<br>GOTO<br>GOTOR             |   |   |  |   |   |  |  |  |  |  |  |   |     |
| bit 20, 19<br>(parameters)                 |   |   |  |   |   |  |  |  |  |  |  |   |     |
| 0  | 1   | 00  | 01   | 10  | 11  | 00   | 01   | 10   | 11   | 00   | 01   | 10  | 11  |
| REP<br>REP                                 | REPP<br>REPP                                | REPE<br>REPE                                | REPE<br>REPE   | (:STAT,0)<br>(:STAT,4)                      | (:STAT,1)<br>(:STAT,5)                      | (:STAT,2)<br>(:STAT,6)   | (:STAT,3)<br>(:STAT,7)                                       |  |  |  |  |   |     |
| bit 15<br>(conditievolgning)               |   |   |  |   |   |  |  |  |  |  |  |   |     |
| 00   | 01  | 10  | 11   | 000   | 001   | 010  | 001  | 010  | 001  | 010  | 001  | 100   | 110 |
| bit 15<br>(conditievolgning)               |   |   |  |   |   |  |  |  |  |  |  |   |     |
| 0  | 1   | 0   | 1  | 0   | 1   | 0  | 1  | 0  | 1  | 0  | 1  | 0   | 1   |
| bit 17<br>(toegevoegd aan functiedeel)     |   |   |  |   |   |  |  |  |  |  |  |   |     |
| 1 1 0 0 1 1 1 1 0 1 1 1 1 1 1              | U, U | Y, Y | SUB(:STAT,0)<br>SUBCD(:STATR)<br>SUB(:STAT,4)<br>SUB(:STAT,12)<br>DO(:STATR)                         | U, U | N, N | SUB(:STAT,8)<br>SUBCD(:STATR)<br>SUB(:STAT,4)<br>SUB(:STAT,12)                                   | SUBCD(:STAT)<br>SUB(:STAT,5)<br>SUBCD(:STATN)<br>SUBCD(:DYN) | SUB(:STAT,1)<br>SUBCD(:STATB)<br>SUB(:STAT,13)<br>SUBCD(:STATB)  | SUB(:STAT,10)<br>SUBCD(:STATB)<br>SUB(:STAT,14)<br>DOS(:STATB) | SUB(:STAT,2)<br>SUBCD(:STATB)<br>SUB(:STAT,6)<br>DOS(:STATB) | SUB(:STAT,3)<br>SUBCD(:STATB)<br>SUB(:STAT,7)<br>DOS(:STATB) | SUB(:STAT,11)<br>SUBCD(:STATB)<br>SUB(:STAT,15) |     |

? niet gebruikt

E! 8

## interne codering van opdrachten

