Notulen van de vergadering der Z8-commissie op 5 december 1963.

Aanwezigen: H.H. Dek, Dijkstra, Kruseman Aretz, v.d. Meulen, v.d. Poel, Schmidt, v.d. Sluis, van Wijngaarden en Zwanenburg.

De notulen Z8 Nr. 5

Punt 1 inzake het F-register moet als volgt luiden:
Bij alle in-opdrachten in het F-register zullen de tekenbits van de geheugen-operand op gelijkheid gecontroleerd worden; indien de tekens niet gelijk zijn wordt OF gezet.
Bij de schoon in-opdrachten zullen de tekens (al of niet verschillend) ongewijzigd in de tekenbits van het F-register worden gecopiëerd. Bij aritm. in-opdrachten doet het teken van de staart (zowel van F als van de geheugen-operand) niet mede; het resultaat in F heeft 2 gelijke tekens.
Bij de schoon uit-opdrachten zullen de tekenbits van F (al of niet verschillend) ongewijzigd in de tekenbits in het geheugen worden gecopiëerd.

Directieven.

Prof. v.d. Poel doet enkele voorstellen over de directieven van de handcode. In de bijlage worden deze voorstellen nader uit de doeken gedaan.

Volgende vergadering.

Datum : 19 december 1963
Tijd : 10 uur
Plaats : M.C.

Als agendapunt draagt de secretaris voor twee voorstellen van de heer A.W. Dek met betrekking tot de octale getallen en de notatie daarvan (Z8 nr.4) en wel

1° een voorstel, om dit soort getallen ook als operand in opdrachten te gebruiken. Dit kan van voordeel zijn bij logische vermenigvuldiging e.d.

2° Volgens Z8 nr. 4 moet een octaal getal altijd uit 9 cijfers bestaan. De heer Dek suggereert om de programmeur van de verplichting te ontslaan om ook niet significante nullen te moeten opschrijven.

LZ/AGP

Bijlage Z8 Nr. 6.

Resumé van het voorstel van Prof. van der Poel
inzake directieven handcode X8.

De handcode zal, behalve uit opdrachten en andere informatie, ook elementen bevatten, welke het de assembler mogelijk maken de vorenbedoelde informatie op de door de programmeur gewenste wijze in de machine af te beelden. Deze elementen zullen in het vervolg directieven worden genoemd. Thans volgt een overzicht van de tot nog toe besproken directieven:

1. Plaatsbepalend directief:

   Vorm: LOCATE (<x>)

   Voor de parameter<x>mag ingevuld worden een absoluut adres (b.v. 123), maar ook een naam (van een label of van een variabele). In het laatste geval moet de plaats behorend bij de naam reeds bekend zijn (zie onder 2^{o}). Tot deze standaardnamen zal naar alle waarschijnlijkheid ook een naam behoren, die het mogelijk maakt de keuze van het absolute adres aan de coördinator over te laten.

   De assembler bergt nu de achter het directief LOCATE volgende informatie op in opeenvolgende adressen, te beginnen met het adres aangewezen door<x>

2. Directieven BEGIN en END

   Door deze directieven wordt het mogelijk in het programma een blokstructuur, ongeveer als in ALGOL te introduceren.
   De vorm van het eerste directief is

   <labelnaam>: BEGIN (<naam>,<naam>)

   De labelnaam mag behalve in één geval niet ontbreken; het fungeert als de naam van het blok.
   De parameterlijst bevat de namen van de locale variabelen. Zij bestaan alleen gedurende de activiteit van dat blok.
   Behalve van de variabelen bevat de lijst ook de namen van alle binnen dat blok voorkomende labels. Door het opnemen van een naam in de parameterlijst wordt die naam gedeclareerd.

   De assembler eist nu dat elke vrije naam, welke gebruikt wordt, wordt gedeclareerd.

   Een bijzonderheid is dat in de parameterlijst ook de labels van opdrachten in binnenblokken mogen worden gedeclareerd. In tegenstelling tot ALGOL is het hierdoor mogelijk van een buitenblok ergens naar een willekeurig punt in een binnenblok te springen.

Ingevolge de eis dat alle labels binnen een blok vooraf moeten worden gedeclareerd, moeten in de parameter lijst van een blok ook alle namen van de zich op één niveau lager bevindende binnenblokken in dat blok worden gedeclareerd. Dit levert moeilijkheden op voor het alles omvattende binnenblok, omdat hiervoor de label niet kan worden gedeclareerd. In verband hiermede blijft BEGIN van het binnenblok ongelabeld.

Het directief END heeft voor de blokstructuur de zelfde betekenis als end in ALGOL. De vorm is

END (< labelnaam >)

De parameter moet de naam van het blok bevatten dat door END inactief wordt. Logischerwijs is de vermelding van deze naam overbodig; het is dan ook alleen bedoeld om de overzichtelijkheid van het programma te verhogen, terwijl het aan de assembler de mogelijkheid biedt controles op de logische structuur door te voeren.
Zoals reeds gezegd, worden door END alle variabelen, welke in de parameter lijst van de bijbehorende BEGIN zijn gedeclareerd, ongeldig. In het algemeen zal hun plaats in het geheugen door nieuwe variabelen worden ingenomen.
De allerlaatste END (behorend bij de allereerste ongelabelde BEGIN) heeft geen parameter.
Het directief START (< x >)
Voor <x> bestaan dezelfde voorschriften als voor LOCATE.

De functie van START is

1° dat aan de coördinator wordt medegedeeld, dat het programma(deel), dat zojuist is ingelezen kan worden gestart.

2° dat de activiteiten van de assembler tot nader order worden gestaakt.

Een voorbeeld van het gebruik van de hiervoren genoemde directieven:

```
    LOCATE (123)
    BEGIN (L)
L:  BEGIN (K, X, Y, Z)
    <opdracht>
    <opdracht>
    .........
K:  BEGIN (P. Q R)
    <opdracht>
    ........
```

```
END (K)
<opdracht>
........
........
END (L)
END
START (L)
```

DIRECTIEVEN DYN en STAT

Vanaf het moment dat het directief DYN wordt gelezen, worden voor alle namen, welke ter declaratie worden aangeboden, dynamische adressen uitgetrokken.

Het directief DEFINE

De algemene vorm is

DEFINE (<definenaam> FORMAL (<formalnaam>,<formalnaam,.....)
<stuk programma>)

De bedoeling van define is om het eenmalig geschreven <stuk programma> dat als parameter wordt medegegeven tijdens de assemblage op een enkele aanwijzing meermalig in de tekst te kunnen insereren. De aanduiding FORMAL heeft betrekking de de daarachter gegeven formelen.

Tijdens de assemblage wordt de tekst van define (de parameters dus) ergens opgeslagen. De inserering vindt plaats zodra ergens in de na volgende tekst de <definenaam> wordt geconstateerd (de z.g. call). De call moet voorzien zijn van een parameterlijst, waarin de opsomming de actuele parameters, in getal gelijk aan de formele parameters in de DEFINE declaratie. <stuk programma> wordt nu te voorschijn gehaald en op de plaats van de call geïnsereerd, waarbij de namen van de formelen worden vervangen door de namen van de overeenkomstige actuelen. <stuk programma> mag alle elementen van een normaal programma bevatten, dus ook de nodige directieven.

De declaratie van DEFINE draagt een ander karakter dan die van een subroutine. De laatste is ook in het object stadium maar éénmaal aanwezig; de programmatekst van DEFINE, is echter evenveel maal in de machine aanwezig als het aantal call's. Er wordt verwacht dat DEFINE voordelig gebruikt kan worden om de aanroepen van macro's met wat langere parameterlijsten, wat korter de kunnen noteren.