Notulen van de Z8-bespreking op 10 okt. 1963.

Aanwezig waren: Prof. Dijkstra
Dr. Kruseman Aretz
Prof. v.d. Poel
Hr. Schmidt
Prof. v.d. Sluis
Prof. v. Wijngaarden
Hr. Zwanenburg

Correcties en aanvullingen Z8 nr. 1

De notulen Z8 nr. 1 worden, onder voorbehoud van de volgende correcties, goedgekeurd.

1e "Kruseman-Arets" moet zijn "Kruseman Aretz"

2e "RVU" moet zijn "RUU" (Rijksuniversiteit Utrecht)

3e De onderdelen van Z8, waarvoor de RUU speciale belangstelling heeft, zijn dezelfde als die van de PTT.

4e Bij degenen die meespraak in de ALGOL-groep verlangen, schaart zich EL.

Compatibiliteit

Prof. Van der Sluis stelt de vraag, of de compatibiliteit met de X1 die bij de bitrepresentatie van X8 opdrachten steeds in het oog is gehouden, niet tot ongewenst gecompliceerde assemblers leidt (lange lijsten). Een vluchtige beschouwing van het overzicht van de opdrachten en hun interne representaties laat vermoeden dat het met de gevreesde moeilijkheden wel zal meevallen. Uiteraard is er niet aan te ontkomen, dat er een aantal opdrachten zijn waarvan er een minder duidelijke relatie tussen externe en interne schrijfwijze bestaat.

Aritmetiek bij adresberekening als gevolg van indirecte adressering.

Het volgende wordt in de speciale aandacht van de hardware-groep aanbevolen

Overal waar bij de indirecte adressering voor de bepaling van het absolute adres gebruik gemaakt wordt van een 27-bits-woord, doet de inhoud (hetzij uit een register, hetzij uit een geheugenwoord) van de bits 18-25 niet mede. In elk van deze bits moet het teken bit (bit 26) gesubstitueerd gedacht worden. Dit geldt ook voor de van ouds bekende B-correctie. Als we het op deze wijze bewerkte 27-bits-woord X even aangegeven met X^1 dan geldt dus (notatie X8 no. 16)

$$B,n = B^1 + n - 16384$$
$$C,q = B^1 + q - 256$$

op dezelfde wijze

$$p,q = M^1 (L + p) + q - 256$$
$$(0 \leq p \leq 57)$$

waarin $L = M^1(63)$

en $q = r^{1^\cdot} + q - 256$

waarin r = A, S, B, T, F, of, L)

Het gevormde resultaat moet in ieder geval ≥ 0 en $< 2^{18}$ zijn. Voor installaties met niet zulke grote geheugens zal in plaats van 2^{18} een kleinere macht van 2 als bovengrens kunnen optreden, echter nooit kleiner dan 2^{15}.
Indien het berekende adres niet binnen de gestelde grenzen blijft zou dit het beste door een ingreepsignaal kunnen worden geïndiceerd.

Let wel, dat als gevolg van de hierboven genoemde conventie

$$A = M(L,3)$$

iets anders betekent dan

$$A = 3,0$$

Beide opdrachten hebben als geheugenoperand de inhoud van het indirecte adres L,3, de eerste neemt de volle 27 bits over in A, de tweede alleen de bits 0-17 en het tekenbit, terwijl de resterende bits van A eveneens met het tekenbit worden gevuld.

Agende Z8-commissie

De voorzitter stelt voor een aantal urgente punten op de agenda van deze en van de komende vergaderingen te zetten. Het betreft in volgorde van behandeling:

0e simulator
1e handcode
2e MC-Algol-compiler

Handcode

De handcode moet voldoen aan de eis dat elke machine-opdracht door één handcode statement is te representeren. Omgekeerd moet elke handcode-opdracht in één machine-operatie te verwezenlijken zijn. Ook indien de handcode vergaande bevoegdheden aan de assembler toekent, moet het de programmeur altijd mogelijk zijn aan de geheugenplaats, welke hij uitdrukkelijk aanwijst, een inhoud te geven, welke bit voor bit aan zijn specificaties voldoet.
Naar de faciliteiten, welke een handcode kan bieden, zijn er de volgende niveau's te onderscheiden

1e niveau — Machine-code. Elke statement voldoet aan de eigenschap, dat hij zonder verdere informatie is om te zetten in de bit-representatie van de machine-opdracht. (De code beschreven in X8 no. 16 is zo'n code).

2e niveau — Statisch relocatie systeem. De tekst van het programma laat nog een aantal grootheden ter definiering aan de assembler; meestal betreft dit de plaats van het programma en van de variabelen. Eenmaal in de machine ingebracht is zo'n programma niet meer te onderscheiden van een direct in machine-code geschreven programma (Voorbeeld: X1-handcode met enige restricties)

3e niveau — Dynamisch relocatie-systeem. Hierbij is een mogelijkheid van relocatie tijdens het uitvoeringsstadium voorzien.

Een tussen-niveau tussen 2e en 3e is een systeem, dat wel statisch is, maar dat rekening houdt met de simultane uitvoering van meer programma's.

Allereerst wordt besproken de code op niveau 1. Hiervoor zijn 2 voorstellen n.l. één van de heer Van der Meulen, toekomstig medewerker van het Electronisch Rekencentrum van de RUU, en de code beschreven in X8 nr. 16 (Deze code is eigenlijk al aanvaard, zie X8 nr. 20).

Prof. Van der Poel geeft in het kort het eerstgenoemde voorstel als volgt weer. De heer Van der Meulen duidt alle variabelen, die corresponderen met geheugen-inhouden, aan met M.

I) M = Mo = inhoud van adres 0
M1 = " " " 1
=
M32767 = " " " 32767

Bijzondere gevallen uit I zijn

II) M57
M58 = F
M59 = A
M60 = S
M61 = B
M62 = T
M63 = X

Uit I en II worden 6 nieuwe namen geformeerd

III) MF = inhoud van het adres dat door F wordt aangewezen
MA = " " " " " " A " "
MS = " " " " " " S " "
MB = " " " " " " B " "
MT = " " " " " " T " "
MX = " " " " " " X " "

Deze namen mogen weer worden geindiceerd met een van de getallen 0-511 (J)

MFj = inhoud van adres F + j

IV) De normale B-correctie wordt aangegeven met

MnB = M (n + B)

V) Naar analogie van IV kan M X k ook geschreven worden als MkX (de inhoud van k gemodificeerd met X)

VI) De variabele MKX (waarin $0 \leq k \leq 57$) mag als index van M optreden dus

MMkX

of na vervanging van MM door J

JkX

VII) Tenslotte mag ook JkX weer geïndiceerd worden

JkXj

VIII) De absoluut-versie van alle hiervoren genoemde "inhouds-vormen" wordt aangegeven door deze door een punt te laten voorafgaan. De betekenis van deze punt wordt echter pas van belang bij de toekenning van namen aan "inhouden".

De commissie acht het verschil tussen deze notatie-wijze en die uit X8 nr. 16 niet zo essentieel om daarvoor de laatste te wijzigen. Met name het aangeven van de absoluut-versie vond men te omslachtig.

Een ander aspect van het voorstel v.d. Meulen is de wijze waarop de operaties worden genoteerd. In de code worden operanden uit registers steeds als eerste genoemd, om het even of het een in - of een uit-opdracht betreft. De veranderende operand wordt door een "richting symbool" aangegeven als volgt:

Code v.d. Meulen	Code X8 nr. 16	Betekenis
A : + Mn	A + Mn	A: = A + Mn
A + : Mn	Mn+ A	Mn: = Mn+ A
A:= - Mn	A-- Mn	A: = - Mn
A-= : Mn	Mn=-A	Mn: = - A

Het richtings-symbool heeft voornamelijk betekenis voor gebruik in operatie - strings. Een en ander wordt door Prof. Van der Poel aan de hand van een voorbeeld toegelicht. Men is unaniem van mening dat het aanelkaar rijgen van opdrachten, zoals dat wordt voorgesteld, niet acceptabel is (Voor het programmeren in formules is ALGOL en niet de handcode).

De vergadering sluit, nadat afgesproken is, dat de discussie op donderdag 24 oktober om 10 uur in het M.C. wordt voortgezet.