

STICHTING
MATHEMATISCH CENTRUM
2e BOERHAAVESTRAAT 49
AMSTERDAM
REKENAFDELING

MICA

Een aanvulling op
en wijziging van
MICRO 48K (en MICRO 48 KL)
ten behoeve van het
gebruik van de
kaartleesapparatuur.

door

H.W. Roos Lindgreen

NR 10



januari 1970

Inhoud:

- 1 algemeen
- 2 geheugenbezetting
- 3 konventies
- 4 beschrijving MICA-routines
- 5 beschrijving MICA-variabelen
- 6 operateurs-aanwijzingen
- 7 labellijst
- 8 programma
- 9 mededeling aan gebruikers

1. Algemeen.

In dit rapport wordt het ELAN programma: MICA gegeven en beschreven. MICA is een wijziging op het MICRO 48K bedrijfssysteem van het Mathematisch Centrum en op het MICRO 48KL bedrijfssysteem (afgezien van de inzetadressen van de eerste pagina).

MICA maakt een bescheiden gebruik van de kaartlezer mogelijk. Kaartinput dient als laatste deel van de input te worden ingelezen. De operateur switcht van bandlezer naar kaartlezer door een "K" in te typen op de kommando-teleprinter. De kaartsymbolen worden vanuit het ALGOL 60 programma bereikt via de normale leesroutines welke voor bandinvoer bestaan. Dit wordt mogelijk gemaakt door uitsluitend dat deel van de reader section van het MICRO systeem te wijzigen, dat betrekking heeft op het proces: bandlezer naar trommel. Van kaart gelezen symbolen worden op zodanige wijze aan de bandlezerstroom toegevoegd, dat de tegenhanger van genoemd proces, dus REHEP, de aangeboden symbolen niet kan onderscheiden naar herkomst.

2. Geheugenbezetting.

MICA bestaat uit een klein gedeelte, waarvan geëist wordt dat het zich beneden 32K bevindt en uit een groter gedeelte, dat vrij lokateerbaar is. Beneden 32K : referentie-adressen uit het gedeelte boven 32K;

variabelen van MICA
 tesamen ca. 25 konsekutieve adressen
 Boven 32K: ca. 250 konsekutieve adressen.

3. Konventies.

In de beschrijving zijn routines van MICRO met een * gemerkt en de routines van MICA niet.

In de beschrijving en in het programma korrespondeert een boolean met een geheugenwoord en: >0 is true, en <0 is false.

In de beschrijving en in het programma zijn uitsluitend identifiers welke betrekking hebben op variabelen met kleine letters geschreven.

Waar, door middel van een inzet-aanwijzing, een MICRO-instructie is vervangen, wordt de vervangen instructie in het commentaar vermeld. Alle vervangen instructies zijn vervangen door een inhoudelijke sprong via een label die met een Q begint naar een label die met een X begint. Alle labels die met een Q beginnen bevatten een adres, in principe boven de 32K (vandaar "inhoudelijk"). Slechts vervangende instructies verwijzen via zo'n label naar een label die met een X begint.

Bij de beschrijving van de MICA variabelen zijn de volgende kortnotaties gebruikt:

- init = geïnitialiseerd door
- wijz = gewijzigd en mogelijk geraadpleegd door
- gebr = geraadpleegd, doch niet gewijzigd, door

4. MICA-ROUTINES.

KSTART 1,	ingang	: KEYBD*
	uitgang	: RESTORE*
	roept aan	: RESET CARD, CRDSTART, BRUSH*
	variabelen	: klag, ktype, done, ibusy
	funktie	: wanneer de operateur een "K" intypt via de kommando-teleprinter zal KEYBD* een sprong naar KSTART 1 uitvoeren. KSTART 1 behandelt de "K", waarbij van belang:

- a. reeds eerder voor dezelfde invoerjob een "K" getypt.
- b. nog tape in de bandlezer.
- c. bandlezer actief van de volgende, nog niet ingelegde, tape.

```

if a then else if b then ktype:= true else
begin < initialiseer kaartleesproces >; if c
then < inaktiveer bandlezer en aktiveer kaart-
lezer>; end

```

bijzonderheden: KEYBD^{*} kan KSTART 1 aanroepen dank zij de
"modification":
CW4 [3]: ['024 000 000'+:KSTART]

RESET CARD, ingang : KSTART 1, PROCESS 5^{*}
 uitgang : normaal
 roept aan : BRUSH^{*}
 variabelen : kflag, colcnt, case, crdss, crdar, crdbusy
 functie : initialiseren van de variabelen van het
 kaartleesproces, en tevens het bereiken
 van de rusttoestand van de kaartlezer.

bijzonderheden: een onderdeel van RESET CARD is RESET CARD 4,
welke wordt aangeroepen door CRD NOK,
PROCESS 5^{*} en CONTRACT 2^{*}

CRD START, ingang : KSTART 1, CRD READER 1, PROCESS 5^{*}
 uitgang : normaal
 roept aan : -
 variabelen : crdbusy, ibunof, crdss, crdar
 functie : aktiveert de kaartlezer (wanneer het appa-
 raat niet reeds actief was): geeft de kaart-
 lezer de opdracht één kaart in te lezen in
 de buffer gespecificeerd door ibunof.

bijzonderheden: een onderdeel van CRD START is TRY AGAIN,
aangeroepen door CRD NOK

CRD READER 1, ingang : IOINTP*

uitgang : RESTORE*

roept aan : CRD NOK, CRD START, CRD CONTRACT

variabelen : crdbusy, ibunof, crdar, comes, done,
atlmgn 1, atlmgn 2

funktie : behandeling ingreep van de kaartlezer.
De 80 ingelezen kaartsymbolen worden aan-
gevuld met een 81e symbool: een nlcr.
Wanneer het transport niet normaal is ver-
lopen, wikkelt CRD NOK dat af. Er wordt
een nieuwe inleesbuffer gespecificeerd en
wanneer deze buffer leeg is, wordt een
nieuwe kaartleesopdracht gegeven. De juist
ingelezen informatie wordt overgepakt naar
een secundaire buffer door CRD CONTRACT.

bijzonderheden: IONTP* kan CRD READER 1 bereiken door de
"modification":
INTAB[8+:CRD] : : CRD READER

CRD NOK, ingang : CRD READER 1

uitgang : RESTORE*

roept aan : CTYPE*, CRD CONTRACT, HOK* (,TRY AGAIN,
RESET CARD 4)

variabelen : crdss, comes, ibunof, crdar

funktie : behandeling "foutieve" kaarttransporten.
Is het transport als foutief gemerkt uit-
sluitend omdat het de laatste kaart van
het bestand was, dan wordt de juist gevulde
buffer als de laatste gemerkt en door CRD
CONTRACT overgepakt. In alle andere gevallen
zal het transport nietig worden verklaard
en de laatste startopdracht worden herhaald;
hetgeen vergezeld gaat van een mededeling
voor de operateur via de kommandoteleprinter
indien het vorige transport nog in orde was

(deze laatste restrictie dient om een waslijst van fouten op de teleprinter te voorkomen).

- bijzonderheden: 1) NBK is een toestand welke in de praktijk nauwelijks zal voorkomen (spanning af impliceert niet NBK).
- 2) de kaartlezer dient tweemaal gehokt te worden.
- 3) spanning af, motor af, en echte feed fail worden alle FEED FAIL genoemd.
- 4) leesfout en kaartslijp worden beide CARD ERROR genoemd.

CRD CONTRACT, ingang : CRD READER 1, CRD NOK, CONTRACT**, STORE

uitgang : normaal

roept aan : CONVERT, STORE, BUF TO DRUM, CONTRACT 2**

variabelen : colcnt, ibunoe, capos, done, dfdsgr

funktie : overpakken van 81 kaartsymbolen uit een primaire buffer naar een secundaire buffer, daarbij de symbolen konverterend naar flexowriter-code. Die konversie gebeurt door CONVERT en het overpakken door STORE. Is een secundaire buffer vol, dan wordt die buffer op trommeltransport gezet, door BUF TO DRUM.

CRD CONTRACT gaat in een loopje lang de kaartsymbolen en verricht wat administratie. Bij de konversie kunnen uit één kaartsymbool twee flexowriter-symbolen ontstaan door een eventueel benodigde case-definitie. Als gevolg hiervan zal het voorkomen dat een secundaire buffer vol is, terwijl de primaire buffer niet geheel leeg is. De programmering is zodanig, dat CRD CONTRACT in zo'n geval, zodra er weer een secundaire

buffer vrij is, de rest van de kaart kontraheert. Hetgeen onder meer te zien is aan de initialisering van het loopje van CRD CONTRACT, dat niet begint met S = konstante, doch met S = variabele (colcnt).

STORE,	ingang	: CRD CONTRACT
	uitgang	: normaal
	roept aan	: BUF TO DRUM
	variabelen	: capos, crbunof 5
	functie	: pak het in A meegegeven symbool in de door crbunof 5 gespecificeerde secundaire buffer en zet zo nodig die buffer op trommeltransport.
BUF TO DRUM,	ingang	: CRD CONTRACT, STORE
	uitgang	: normaal
	roept aan	: TO DRUM*
	variabelen	: capos, crbunof 5, atlmsgn 2
	functie	: het toevoegen van een endmarker aan het laatste symbool uit de buffer en het op transport zetten van de buffer d.m.v. een aanroep van TO DRUM.
CONVERT,	ingang	: CRD CONTRACT
	uitgang	: normaal
	roept aan	: -
	variabelen	: done, case
	functie	: het in A meegegeven symbool wordt gekonverteerd naar flexowriter-code. Het vertaalde symbool wordt afgeleverd in S, de eventuele case-definitie in A en in het conditieregister: C = Y : geen case-definitie in A C = N : wel case-definitie in A

- bijzonderheden: 1) CONVERT maakt gebruik van de konversie-tabel welke deel uitmaakte van de konversie-routine van MICRO. Deze tabel is ongewijzigd op dezelfde plaats in het kerngeheugen blijven staan.
- 2) alle kaartsymbolen met een ekivalent in de flexowriter-symbolen set worden op het korresponderende flexowriter symbool afgebeeld, de overige kaart-symbolen worden erase, d.w.z. 2^7-1 .

case, init : RESET CARD
 wijz : CONVERT
 gebr : -
 functie : bevat de heersende case-definitie, 122 of 124
 en is eventueel "ongedefinieerd" als de case-
 definitie ongedefinieerd is.

capos, init : TO DRUM*
 wijz : STORE
 gebr : CRD CONTRACT, BUF TO DRUM
 functie : symbolenteller in secundaire buffer, loopt:
 0,1,2,4,5,6,8,...,504

crdss[-1:+1], init : -
 wijz : CRD START
 gebr : CRD NOK
 functie : de startschakel voor de kaartlezer (startschakels
 worden één voor één, ongeketend, aangehangen).

6. Operateursaanwijzingen MICA:

- 1) Inlezen van het IP-bandje van MICA kan gebeuren:
 - . direkt nadat MICRO is ingelezen
 - . op een moment dat de machine is "leeggelopen" (ook de ponsbuffer!).
 Dit inlezen geschiedt aldus:
 - a. SVA omhoog
 - b. LS-IP en bandje MIC-IP inlezen
 - c. SVA omlaag
 - d. LS-NB (en systeem start opnieuw met opvragen DATE).

- 2) Toetsinslag K mag plaatsvinden tijdens of na de laatste band van een job. Meerdere malen de toetsinslag K voor dezelfde job heeft hetzelfde effect als éénmaal die toetsinslag. Dit effect is het volgende: wanneer de laatste band is ingelezen, zal het systeem van invoermedium veranderen en de rest van de invoer via de kaartlezer verwachten.

- 3) De invoer over de kaartlezer kan uitsluitend plaats vinden indien SPANNING, MOTOR, AUTO, START en GEREED aan zijn. Advies: altijd EINDE BESTAND aan. Eventuele kaartlees-fouten worden over de teleprinter gemeld en eisen speciale operateurshandelingen, zie 6).

- 4) Toetsinslag E, indien gegeven terwijl het systeem in kaartleestoestand is, mag plaatsvinden tijdens of na inlezen van het laatste kaartenpakket. De kaartinvoer zal gestaakt worden als èn een E is ingeslagen èn de laatste kaart is ingelezen èn bij het inlezen van de laatste kaart EINDE BESTAND aan was. Het systeem zal dan overgaan in de (normale) bandleestoestand en het volgende programma verwachten.

- 5) Toetsinslag N, heeft vrijwel dezelfde functie als toetsinslag E. Het verschil bestaat hierin, dat het systeem niet over gaat in de bandleestoestand, doch kaartinvoer verwacht voor de volgende run van hetzelfde programma.

- 6) In kaartleestoestand kan het systeem de volgende speciale mededelingen

doen over de teleprinter:

CRD NBK : de toestand is hopeloos, de kaartlezer kan niet worden gebruikt. Operateursreactie: toestand verbeteren (b.v. zekering console controleren, EL-service bellen).

INPUT CARDS: het systeem verwacht direkt kaart-invoer, het lopende programma staat er op te wachten. Operateursreactie: kaarten inlezen of de kaartinvoer beëindigen (met E of N).

FEED FAIL : de kaartlezer "krijgt geen kaarten". Operateursreactie: indien SPANNING af, dan SPANNING aan (denk om machine-storingen, dus SVA omhoog, SPANNING aan, SVA omlaag, BVA); indien MOTOR af, dan MOTOR aan; indien de kaarten niet op juiste wijze klaarliggen voor het "invoergat", dan situatie in orde brengen; GEREED aan.

CARD ERROR : de laatstgelezen kaart is òf onjuist gepost en daardoor verworpen, òf de kaartlezer heeft een fout gemaakt en daardoor de laatste kaart verworpen. Indien de kaartlezer aan de specificaties beantwoordt, zal deze laatste kaart in een apart aflegmagazijn zijn afgelegd. Operateursreactie: de kaarten in het invoermagazijn verwijderen (d.w.z. zorgen, dat die kaarten niet meer kunnen worden ingevoerd), m.u.v. de kaart welke zich al gedeeltelijk tussen de geleiderollen bevindt. Dan STOP aan, HAND aan, EEN KAART aan, AUTO aan, START aan. De reeds gedeeltelijk ingevoerde kaart bevindt zich nu ook in het foutenbakje, evenals de foute kaart. Pak die twee kaarten, voeg ze voor het verwijderde bestand en biedt het geheel opnieuw aan de kaartlezer aan. GEREED aan. Wordt de "foute" kaart nu opnieuw niet geaksepteerd (waarvan geen mededeling over de teleprinter wordt gedaan), dan de boven beschreven procedure herhalen doch zonder de "foute" kaart nogmaals aan te bieden. Beschouw de "foute" kaart nu als echt fout en verwittig de cliënt hiervan. Indien tijdens het kaartlezen reeds een "E" of "N" is ingeslagen en een kaart wordt fout gelezen; denk dan om eventuele einde bestand reacties.

7. Labellijst

'005146'	LOOP3	'017375'	QCRDCONTRACT
'005223'	WAIT	'017376'	CRDREADER
'005312'	PROCESS5	'017377'	ISTART
'005362'	TODRUM	'017400'	KFLAG
'005401'	START	'017401'	KTYPE
'005462'	CONTRACT	'017402'	CRDBUSY
'002114'	CTYPE	'017403'	COLCNT
'002077'	RESTORE	'017404'	CAMES
'001757'	BRUSH	'017405'	CASE
'001766'	HOK	'017406'	CAPOS
'005531'	CONTRACT2	'017410'	CRDSS
'017362'	MICRO CRDREADER	'130000'	XLOOP3
'001160'	D18	'130005'	XWAIT
'005074'	REMES	'130013'	CRDMES
'017336'	INTABLE	'130017'	XPROC5
'017357'	INSTR	'130025'	XPROC5A
'001237'	INTREP	'130033'	XTODRUM
'002027'	INTAB	'130037'	XISTART
'002560'	CW4	'130044'	XCONTRACT
'003733'	TRNSGN	'130051'	XCONTRACT2
'003731'	DONE	'130053'	KSTART1
'003730'	IBUSY	'130072'	RESETCARD
'003727'	IBUNOE	'130076'	RESETCARD4
'003726'	IBUNOF	'130106'	CRDSTART
'001155'	ATLMSGN1	'130113'	TRYAGAIN
'001156'	ATLMSGN2	'130121'	CRDCODEW
'003732'	DFDSGN	'130122'	CRDREADER1
'003722'	CRBUNOF5	'130146'	CRDNOK
'000005'	RE	'130164'	EXIT
'000031'	CRD	'130167'	NBK
'000244'	CRDAR	'130175'	FEEDFAIL
'017362'	QLOOP3	'130201'	CARDERROR
'017363'	QWAIT	'130205'	NBKMES
'017364'	QPROC5	'130210'	ENDOFDECK
'017365'	QPROC5A	'130217'	HOKCRD
'017366'	QTODRUM	'130223'	CRDCONTRACT
'017367'	QISTART	'130224'	NEXTCOL
'017370'	QCONTRACT	'130257'	STORE
'017371'	QCONTRACT2	'130305'	BUFTODRUM
'017372'	QCRDREADER	'130325'	CONVERT
'017373'	QKSTART	'130337'	SELECT
'017374'	QCRDSTART	'130342'	RESULT
		'130357'	EIGHT

" MICA , changes and extends MICRO 48K,
" so that card input is possible

'BEGIN'

" routines of MICRO 48K

M['05146 ']:	LOOP 3:
M['05223 ']:	WAIT:
M['05312 ']:	PROCESS 5:
M['05362 ']:	TO DRUM:
M['05401 ']:	ISTART:
M['05462 ']:	CONTRACT:
M['02114 ']:	CTYPE:
M['02077 ']:	RESTORE:
M['01757 ']:	BRUSH:
M['01766 ']:	HOK:
M['05531 ']:	CONTRACT 2:
M['17362 ']:	MICRO CRDREADER:

" constants of MICRO 48K

M['01160 ']:	D18:
M['05074 ']:	REMES:
M['17336 ']:	INTABLE:
M['17357 ']:	INSTR:
M['01237 ']:	INTREP:
M['02027 ']:	INTAB:
M['02560 ']:	CW4:

" variables of MICRO 48K

M['03733 ']:	trmsgn:
M['03731 ']:	done:
M['03730 ']:	ibusy:
M['03727 ']:	ibunoe:
M['03726 ']:	ibunof:
M['01155 ']:	atlmsgn 1:
M['01156 ']:	atlmsgn 2:
M['03732 ']:	dfdsgn:
M['03722 ']:	crbunof 5:

" parameters

M['00005 ']:	RE:	" apparatus number of tapereader
M['00031 ']:	CRD:	" apparatus number of cardreader

M[64 + [4 × : CRD]]:	crdar:	" apparatus register cardreader
-----------------------	--------	---------------------------------

" working space and reference-addresses in first module

MICRO CRDREADER [0]:

Q LOOP 3: : X LOOP 3
Q WAIT: : X WAIT
Q PROC 5: : X PROC 5
Q PROC 5 A: : X PROC 5 A
Q TO DRUM: : X TO DRUM
Q ISTART: : X ISTART
Q CONTRACT: : X CONTRACT
Q CONTRACT 2: : X CONTRACT 2

Q CRD READER: : CRD READER 1
Q KSTART: : KSTART 1
Q CRD START: : CRD START
Q CRD CONTRACT: : CRD CONTRACT

CRD READER: GOTO (Q CRD READER)
KSTART: GOTO (Q KSTART)

kflag: 'SKIP' 1
ktype: 'SKIP' 1
crdbusy: 'SKIP' 1
colcnt: 'SKIP' 1
comes: 'SKIP' 1
case: 'SKIP' 1
capos: 'SKIP' 1
crdss: 'SKIP' 2

" modifications upon MICRO-text

INTAB[8 + :CRD]: : CRD READER

CW4[3]: ['024 000 000' + : KSTART] " K
CW4[-17]: ['033 000 000' + : RESTORE] " R
CW4[-15]: ['014 000 000' + : RESTORE] " I

LOOP 3[9]: GOTO (Q LOOP 3) " [GOTO (: ISTART)
WAIT[4]: N, SUBC (Q WAIT) " [N, SUBC (: CTYPE)
PROCESS 5[25]: SUBC (Q PROC 5) " [A = trnsgn ,P
PROCESS 5[30]: SUBC (Q PROC 5 A) " [A = done ,P
TO DRUM[7]: SUBC (Q TO DRUM) " [A = 1
ISTART[0]: SUBC (Q ISTART) " [A = ibusy ,P
CONTRACT[3]: SUBC (Q CONTRACT) " [G = ibunoe
CONTRACT 2[0]: SUBC (Q CONTRACT 2) " [trnsgn = A ,P

" routines, activated by the modifications

```
M['130000']:      'BEGIN' 'MT' TRY AGAIN, RESET CARD, RESET CARD 4

X LOOP 3:         kflag  = -B           " not card input
                  ktype  = -B           " no K typed yet
                  crdbusy = -B           " cardreader not busy
                  comes  = B            " errormessage allowed
                  GOTO (: ISTART)       " start tapereader

X WAIT:           A = kflag             ,P      " card input?
                  Y, SUBCD (: CTYPE)    " type 'INPUT CARDS'
                    : CRDMES
                  N, SUBCD (: CTYPE)    " type 'INPUT TAPE'
                    : REMES
                  GOTOR (MC[-1])

CRDMES:           '02 10 44 14 0'      " CR LF BL I
                  '06 15 34 01 0'      " N P U T
                  '04 16 30 12 0'      " SP C A R
                  '22 24 77 00 0'      " D S EM

X PROC 5:         A = trnsgn            ,P      " message waiting for transport?
                  Y, A 'X' D18          ,Z      " message E ?
                  Y, kflag = -B         " tape input from now
                  Y, SUBC (: RESET CARD 4) " inactivate cardreader
                    A = trnsgn        ,P
                  GOTO (MC[-1])        " return with condition

X PROC 5 A:       A = done              ,P      " done with last tape?
                  Y, A = ktype          ,P      " K still waiting for action?
                  Y, ktype = -B         " K no longer waiting
                  Y, SUBC (: RESET CARD) " initialize card process
                    A = done          ,P
                  GOTO (MC[-1])        " return with condition

X TO DRUM:        A = 0
                  capos = A             " init secondary buffer
                  A = 1
                  GOTOR (MC[-1])

X ISTART:         A = kflag             ,P      " card input?
                  Y, B = 1              " adjust stackpointer
                  Y, GOTO (Q CRD START) " try to read a card
                    A = ibusy         ,P      " tapereader busy?
                  GOTO (MC[-1])        " return with condition

X CONTRACT:       A = kflag             ,P      " card input?
                  Y, B = 1              " adjust stackpointer
                  Y, GOTO (Q CRD CONTRACT) " contract a card
                    G = ibunoe        " select primary buffer
                  GOTOR (MC[-1])

X CONTRACT 2:     trnsgn = A
                  GOTO (Q PROC 5)
```


CRD READER 1:

```
'BEGIN' 'MT' CRD NOK, HOK CRD

['660 071 000' + :CRD] " clear IF cardreader
S = ibunof " select primary buffer
MS = B " not last buffer
A = 4095
MS[41] = A " 81st column := 'nlcr'
crdbusy = -B " cardreader not busy
A = crdar[0] ,P " transport not erroneous?
N, GOTO (: CRD NOK)
comes = B " errormessage allowed
A = done ,P " first card of deck?
Y, done = -B
Y, A = atlmsgn 1
Y, atlmsgn 2 = A
A = -MS[-1]
MS[-1] = A
ibunof = A
S = MA[-1] ,P " next primary buffer filled?
N, SUBC (: ISTART) " read another card
N, SUBC (: CONTRACT) " contract last card
GOTO (: RESTORE)
```

CRD NOK:

```
'BEGIN' FEED FAIL, CARD ERROR, NBK MES,
'MT' NBK, END OF DECK, EXIT
```

```
RUA (18) ,Z " NBK ?
Y, GOTO (: NBK)
A = crdss [-1] " take reportword
A 'x' 31 " remove extraneous bits
U, A - 16 ,Z " end of deck?
Y, GOTO (: END OF DECK)
U, A = comes ,P " errormessage allowed?
N, GOTO (: EXIT)
comes = -B " no longer errormessages
U, A 'x' 1 ,Z " not feed fail detected?
N, SUBC (: CTYPE) " type 'FEED FAIL'
: FEED FAIL
Y, SUBC (: CTYPE) " type 'CARD ERROR'
: CARD ERROR
SUBC (: HOK CRD) " HOK cardreader twice
SUBC (: TRY AGAIN) " repeat last start instruction
GOTO (: RESTORE)
```

EXIT:

NBK:

```
A = comes ,P " errormessage allowed?
Y, SUBC (: CTYPE) " type 'CRD NBK'
: NBK MES
Y, comes = -B " no longer errormessages
SUBC (: RESET CARD 4) " reset cardreader
GOTO (: EXIT)
```

FEED FAIL:

```
'02 10 44 26 0' " CR LF BL F
'20 20 22 04 0' " E E D SP
'26 30 14 11 0' " F A I L
'77 00 00 00 0' " EM
```

CARD ERROR:	'02 10 44 16 0'	" CR LF BL C
	'30 12 22 04 0'	" A R D SP
	'20 12 12 03 0'	" E R R O
	'12 77 00 00 0'	" R EM
NBK MES:	'02 10 44 16 0'	" CR LF BL C
	'12 22 04 06 0'	" R D SP N
	'23 36 77 00 0'	" B K EM
END OF DECK:	MS = -B	" mark last buffer
	A = MS[-1]	
	MS[-1] = -A	
	ibunof = -A	
	SUBC (: HOK CRD)	" HOK cardreader twice
	SUBC (: CONTRACT)	" contract a card
	GOTO (: RESTORE)	
	'END'	" end of CRD NOK
HOKCRD:	A = : crdar[1]	
	SUBC (: HOK)	" first HOK
	A = : crdar[1]	
	GOTO (: HOK)	" second HOK
	'END'	" end of CRD READER 1
CRD CONTRACT:	'BEGIN' 'MT' NEXT COL, CONVERT, STORE, BUF TO DRUM	
NEXT COL:	S = colcnt	
	U, S 'x' 1 ,Z	" symbol at right hand?
	RUS (1)	
	S + ibunoe	
	A = MS[41]	" take information word
	N, RUA (12)	" take symol at the left
	Y, A 'x' 4095	" take symbol at the right
	SUBC (: CONVERT)	" convert to flex-code,
		" no case definition needed?
	N, SUBC (: STORE)	" store case definition
	A = S	" take converted symbol
	S = 1	
	PLUSS (colcnt) ,Z	" card totally contracted?
	SUBC (: STORE)	" store converted symbol
	N, GOTO (: NEXT COL)	
	S = -81	
	colcnt = S	" init primary buffer
	A = ibunoe	
	S = MA[-1]	
	MA[-1] = -S	
	ibunoe = S	
	S = MA ,P	" not end of deck?
	Y, GOTOR (MC[-1])	
	S = capos ,Z	" secondary buffer on transport?
	N, SUBC (: BUF TO DRUM)	" put it on transport
	done = B	
	A = dfdsgr	
	dfdsgr = -B	
	GOTO (: CONTRACT 2)	

STORE:

```
MC = S           " preserve S
S = capos
RUS (2)
G = S
G + crbunof 5
S = capos
S 'x' 3         ,Z   " first symbol in word?
Y, MG[1] = S    " zero the word
Y, S = 1
N, JUMP (S)     " )
LUA (9)        " ) shift symbol
LUA (9)        " )
PLUSS (capos)
MG[1] + A      " store the symbol
S = 504        ,Z   " secondary buffer exhausted?
N, S = MC[-1]  " restore S
N, GOTOR (MC[-1])
SUBC (: BUF TO DRUM) " transport secondary buffer
S = MC[-1]    ,Z   " card completely contracted?
Y, GOTOR (MC[-1])
B = 1          " adjust stackpointer
GOTO (: CONTRACT) " contract rest of card
```

BUF TO DRUM:

```
S = capos
RUS (2)
G = S
G + crbunof 5
S = capos
S 'x' 3         ,Z   " 3 symbols in last word?
Y, F = 1
A = 256        " endmarker
N, LUA (9)     " )
N, S = 1       ,P   " ) shift endmarker
N, LCA (9)     " )
MG[1] + A      " add endmarker to symbol
               " last stored

G = crbunof 5
A = atlmsgn 2
MG = A
GOTO (: TO DRUM) " buffer on transport
```

```

CONVERT:          'BEGIN' 'MT' EIGHT, SELECT, RESULT

                  done = -B           " preserve stackpointer
                  S = 0
U, A = 4095        ,Z           " 'nlcr' ?
Y, S = 26
Y, GOTO (MC[-1])  " C = Y : no case
                  RUAS (3)           " zonepunchings into S
U, A 'X' 128      ,Z           " no punching in row 8?
N, GOTO (: EIGHT)
                  NORA              " normalize bits
                  A = INTABLE[B - 11] " word from conversion table

SELECT:          NORS                ,Z           " analyze zone punchings
N, DO (INSTR[B]) " shift word
                  A + 48

RESULT:         B = -done            " restore stackpointer
                  A 'X' 127          " remove extraneous bits
                  A + : INT REP      " base address conv table
                  S = MA             " select word from table
                  A = 124            " try upper case
                  LCS (10)          ,P           " guess correct?
N, A = 122        ,E           " perhaps lower case
                  RCS (1)           " case independant?
                  S 'X' 127         " remove extraneous bits
Y, JUMP (2)
U, A = case       ,Z           " case correct?
N, case = A       " new case
                  GOTO (MC[-1])     " C = Y : no case

EIGHT:         A 'X' 126            " remove d9, d8 and d1
                  NORA              ,Z           " normalize remaining bits
                  A = INTABLE[B - 19] " select word from table
Y, GOTO (: SELECT)
                  NORS              ,Z           " analyze zone punchings
N, DO (INSTR[B]) " shift word
                  A + 96
                  GOTO (: RESULT)

                  'END'             " end of CONVERT

                  'END'             " end of CRD CONTRACT

                  'END'             " end of 'routines'

                  'END'             " end of program

```

25-11-1969

Wijziging MC ALGOL 60 systeem t.b.v. kaartleesapparatuur.

Door de wijziging van het MC ALGOL 60 systeem is het mogelijk om op bescheiden wijze van de kaartleesapparatuur gebruik te maken. Deze wijziging is van tijdelijke aard, en de gebruikers wordt aangeraden zich in verbinding te stellen met de heer J. van Loenen alvorens een kaartleesprogramma te konstrueren.

Volgt een korte beschrijving.

Slechts de invoergegevens voor een programma kunnen in de vorm van 80 - koloms - ponskaarten worden aangeboden.

Hoewel de werkelijke gang van zaken anders is, kan de wijze van verwerking van de kaarten, van gebruikersstandpunt uit, als volgt worden beschreven:

- . de aangeboden kaarten worden in fiktieve ponsband omgezet; van iedere kaart worden alle 80 kolommen, naar flexowriter-symbolen gekonverteerd en vervolgens "geponst", waarbij case-definities slechts voor zover noodzakelijk worden gegeven; na verwerking van iedere kaart (80 kaart-symbolen) wordt een NLCR "geponst".
- . vervolgens wordt deze fiktieve ponsband als laatste band van het invoerbestand ingelezen en de gebruiker zal in zijn programma dus op "normale" wijze (read, READ, RESYM) over de kaart-informatie kunnen beschikken, zie tabel.

Het gebruik van REHEP zal dus, indien de informatie in kaarten stond, niet het "gaatjes patroon" in de kaart opleveren, doch het gaatjespatroon van het korresponderende flexowritersymbool; en men wordt dus ernstig afgeraden om REHEP bij kaarten te gebruiken.

De huidige regeling met betrekking tot de in kaarten geponste informatie is zoals gezegd een tijdelijke. Wanneer deze regeling in de toekomst door een andere wordt vervangen zullen waarschijnlijk die programma's welke slechts read, READ en RESYM voor kaarten gebruiken vrijwel ongewijzigd kunnen blijven werken, hetgeen zeker niet zal gelden wanneer ook van REHEP voor kaarten gebruik gemaakt is.

Z.O.Z.

symbol	kaartcode	RESYMwaarde	symbol	kaartcode	RESYMwaarde
0	0	0	U	0-4	30
1	1	1	V	0-5	31
2	2	2	W	0-6	32
3	3	3	X	0-7	33
4	4	4	Y	0-8	34
5	5	5	Z	0-9	35
6	6	6	^	12	80
7	7	7]	12-2-8	101
8	8	8	.	12-3-8	88
9	9	9	<	12-4-8	72
A	12-1	10	(12-5-8	98
B	12-2	11	+	12-6-8	64
C	12-3	12		12-7-8	127
D	12-4	13	-	11	65
E	12-5	14	[11-2-8	100
F	12-6	15	*	11-4-8	66
G	12-7	16)	11-5-8	99
H	12-8	17	;	11-6-8	91
I	12-9	18	¬	11-7-8	76
J	11-1	19	/	0-1	67
K	11-2	20	10	0-2-8	89
L	11-3	21	,	0-3-8	87
M	11-4	22	-	0-5-8	126
N	11-5	23	>	0-6-8	74
O	11-6	24	?	0-7-8	122
P	11-7	25	:	2-8	90
Q	11-8	26	√	4-8	79
R	11-9	27	'	5-8	120
S	0-2	28	=	6-8	70
T	0-3	29	"	7-8	121
(overgang nieuwe kaart		119)	spatie	blanke kolom	93

Kaartsymbolen welke niet in de tabel voorkomen worden door read,RESYM en READ geskipt; de REHEP-waarde van dergelijke symbolen is 127.