

STATISTISK SENTRALBYRÅS HÅNDBØKER

Nr. 18

Oslo, 1. august 1960

P R O G R A M M E R I N G F O R D E U C E

A n n e t h e f t e

P R O G R A M M E R I N G F O R D E U C E

A n n e t h e f t e

Statistisk Sentralbyrå

Oslo, 1960

F o r o r d

Statistisk Sentralbyrå sender med dette ut Programmering for DEUCE, Annet hefte. Dette hefte tar sikte på å gi programmerere orientering om bruk av kontrollbordet og input/outputenheten og om de erfaringer en har gjort gjennom ett års arbeid med hensyn til organisasjon av programmeringsarbeidet.

Håndboken er utarbeidd av førstesekretær Thor Aastorp ved Kontoret for sentral databearbeiding.

Statistisk Sentralbyrå, Oslo, 11. mai 1960

Sigmy Arctander

Svein Nordbotten

I n n h o l d

	Side
1. KONTROLLBORDET	5
1.1. Generelt om kontrollbordet	5
1.2. Monitors	5
1.3. Output Staticiser	7
1.4. Input Dynamiciser	8
1.5. Styring av maskinens hastighet	10
1.6. "Go"-lampen	11
1.7. "TCA"- og "TCB"-lampene	11
1.8. "Instruction Staticiser"-lampene	11
1.9. "Transtim"- og "Timci"-bryterne	11
1.10. "External Tree"-bryteren	11
1.11. Tvunget vegvalg	14
1.12. "Clear Store"-bryteren	15
1.13. Alarm	15
1.14. "Read"- og "Punch"-bryterne	15
1.15. Programme Display	16
1.16. Magnettrommel-lampene	16
1.17. Diverse lamper	17
2. IBM 528	18
2.1. Generell beskrivelse av IBM 528	19
2.2. Den binære metoden	19
2.2.1. Binær innlesing	20
2.2.2. Binær punching	24
2.2.3. Tidsspesifikasjoner	26
2.3. Den desimale metoden	29
2.3.1. Buffer-minne	30
2.3.2. 6-bits binærkoder	31
2.3.3. Desimal innlesing	33
2.3.4. Desimal punching	37
2.3.5. Samtidig punching og lesing	37
2.3.6. Tidsspesifikasjoner	38
2.4. Plugging på IBM 528	39
2.4.1. Standard plugget pluggbord	39
2.4.2. Kort- og "job"-nummer	39

	Side
2.5. Kontrollbrytere og alarmlamper	40
2.5.1. Kontrollbrytere	40
2.5.2. Alarmlamper	40
2.6. Kort-"crash"	41
3. SKRIVING, PUNCHING, TESTING OG KJØRING AV PROGRAM	41
3.1. Skrivning av program	41
3.1.1. Diagrammer	42
3.1.2. Koding	48
3.2. Punching av program	50
3.2.1. Initialkort	52
3.2.2. Clock Track	54
3.2.3. Read to Drum	55
3.3. Program-testing	55
3.3.1. Kontroll før testing på DEUCE	56
3.3.2. Testing på DEUCE	56
3.4. Kjøring av programmet	59
3.4.1. Operatorinstruks	60
4. LETTKODINGSSYSTEMER FOR DEUCE	61
4.1. Behov for lettkodingssystemer	61
4.2. Gruppering av forskjellige lettkodingssystemer	62
4.3. Detailed Coding Scheme	63
4.4. Storage Allocation and Coding Programme	64
4.5. Alfasetemet	64
4.6. General Interpretive Schemes	64
4.7. Tabular Interpretive Schemes	65
4.8. Alfasetemets translator	65
4.9. Easicode	66
4.10. SODA	66

Tegning av bryterne på IBM 528

Tegning av kontrollbordet

Tegning av pluggbordet

1. KONTROLLBORDET

1.1. Generelt om kontrollbordet

Bryterne og lampene på kontrollbordet til DEUCE kan deles opp i to grupper:

- a) Lamper og "monitors" som forteller operatøren hva DEUCE har gjort, hva den gjør eller hva den skal til å gjøre.
- b) Brytere og lamper som kan brukes til å dirigere eller styre DEUCE.

Normalt kan en bruke alt det utstyr som finnes på kontrollbordet, men under programmeringsarbeidet bør en ha til hovedregel at programmene skal lages slik at de blir minst mulig avhengig av manuelle inngrep på kontrollbordet. Slike manuelle operasjoner krever ofte relativt lang tid og gir store muligheter for operatørfeil. De hjelpemidlene som finnes på kontrollbordet, blir derfor helst brukt under testing av programmer og under det tekniske vedlikehold av maskinen.

Da kontrollbordet oftest blir brukt til å iaktta eller forandre den normale arbeidsgangen i DEUCE, vil en først ganske kort repetere denne arbeidsgangen.

DEUCE arbeider ved å utføre en serie av ordrer eller instruksjoner. Hver ordre spesifiserer normalt en overføring innen maskinen og lagerplassen for neste ordre i serien. En ordre som skal utføres, blir overført til TS COUNT i samme minor-cycle som den er lagret i. I neste minor-cycle som kalles "Set-up Minor-cycle", blir forbindelsene til den Source, Destination og Next Instruction Source (NIS) som ordren spesifiserer, valgt ut og satt opp. Selve overføringen vil imidlertid først finne sted senere, og blir styrt av et signal eller impuls som kalles "Transtim" eller "TT" og som varer et helt antall minor-cycles. Tidspunktet og varigheten av denne Transtim-impulsen blir bestemt av "Wait", "Characteristic" og i noen tilfelle av "Time" i ordren. Neste ordre blir overført til TS COUNT i den minor-cycle som en annen impuls, "TCI", opptrer. Tidspunktet for denne impulsen bestemmes av "Time", og varigheten er alltid én minor-cycle.

Enkelte ordrer kan være såkalte stopp-ordrer; disse ordrene blir overført til TS COUNT på vanlig måte, men blir ikke utført før kontrollorganet mottar en bestemt impuls som kalles "Single-Shot". Så snart denne impulsen kommer, fortsetter DEUCE operasjonene med full hastighet inntil ny stopp-ordre.

1.2. Monitors

Ved hjelp av to "monitors" kan innholdet av en hvilken som helst posisjon i hurtigminnet avleses. På disse monitors blir hvert ord representert som en

horisontal linje á 32 grønne punkter eller prikker. En binær ener kommer som et lysende grønt punkt, mens nuller opptre som svakt lysende punkter. Her som over alt ellers i DEUCE, er det omvendt binær representasjon slik at laveste bit representeres av ytterste venstre punkt og det høyeste bit kommer til høyre.

Som nevnt er det to monitors; på høyre monitor kan en avlese innholdet av de lange delay-linjene og TS COUNT, mens innholdet i de øvrige registrene i hurtigminnet kan avleses på venstre monitor.

På høyre monitor er det 32 horisontale linjer á 32 prikker slik at alle ordene innen en delay-linje får plass samtidig. Ved hjelp av en bryter like under monitoren kan en velge hvilken som helst DL. Denne bryteren har 14 forskjellige stillinger, 12 av disse står for DL 1 - DL 12, mens en ved å sette bryteren i posisjon 13 kan få avlest TS COUNT på monitoren. Den 14. posisjonen på bryteren er foreløpig ikke i bruk.

Når det befinner seg et program eller del av et program i hurtigminnet, sier en at linjene på monitoren er riktig innstilt, når den øverste linjen på monitoren viser innholdet av minor-cycle 0 i den DL som bryteren er innstilt på. Etter at en har kjørt et program inn i hurtigminnet, vil en finne at alle delay-linjene viser samme minor-cycle på øverste linje, men om den øverste linje tilsvarende minor-cycle 0 eller ikke vil være helt tilfeldig. Opplysningene på monitoren vil som regel være nyttigst når monitoren er riktig innstilt. Dette kan en gjøre ved hjelp av en knapp på kontrollbordet som heter "M.c. slip". Denne knappen sitter midt mellom og litt under de to monitorene. Ved å trykke på denne knappen blir alle linjene skiftet to plasser nedover; de to nederste linjene blir flyttet til toppen av monitoren. På grunn av dette, og fordi koplingen er slik at den øverste linjen på monitoren alltid representerer en "lik" minor-cycle, kan en ved å trykke maksimum 15 ganger på "M.c. slip"-knappen få innstilt linjene på monitoren riktig. (En vil imidlertid finne at i noen tilfelle kan en trykke på knappen uten at noen skifting finner sted. I slike tilfelle trykker en bare på knappen inntil linjene flytter seg.)

TS COUNT vil en ikke kunne undersøke på monitoren når DEUCE er i gang. Når DEUCE står, dvs. ingen operasjon blir utført, vil en på monitoren finne 32 utgaver av den ordren som står i TS COUNT. P 1, NIS, S, D, P 15 og P 32 (G 0) er like på alle linjene, P 16 og P 25 er lik null på alle linjene, mens P 22 - P 24 og P 31 vil avvekslende være 0 og 1, slik at de på monitoren vil opptre som middels lysende punkter. Når monitoren er riktig innstilt, finner en at den linjen som tilsvarende den minor-cycle, hvor ordren ble hentet inn i TS COUNT, vil ha samme Wait og Time som ordren. Den neste linje på monitoren vil ha en Wait og Time som er én mindre, linjen deretter vil ha Wait og Time som er to mindre, osv. Den linjen som tilsvarende minor-cycle 30 vil ha en Wait og Time

som direkte oppgir første operasjonscycle og den minor-cycle hvor neste ordre skal overføres til TS COUNT.

E k s.:

DEUCE står med ordren "2,4-13,0,0,31" i TS COUNT.

Denne ordren er overført fra DL 2₈. Når monitoren er riktig innstilt, finner en følgende:

linje 9 (m.c. 8):	2,4-13,0,0,31
" 10 (m.c. 9):	2,4-13,0,31,30
" 11 (m.c.10):	2,4-13,0,30,29
osv.	
linje 31 (m.c.30):	2,4-13,0,10,9

En kan altså på linje 31 direkte avlese at operasjonen skal utføres i m.c. 10, dvs. DL 4₁₀ skal overføres til TS 13, og at neste ordre overføres til TS COUNT i m.c. 9, dvs. at det blir innholdet av DL 2₉ (kfr. NIS).

På den venstre monitoren vil en finne de øvrige registrene i hurtigminnet, nemlig alle TS, DS og QS. Disse er ordnet slik: Øverst kommer fire linjer som viser innholdet av TS 13, TS 14, TS 15 og TS 16. Deretter kommer 3 dobbeltlinjer som representerer DS 19, DS 20 og DS 21. Til slutt er det to grupper med 4 linjer i hver, og disse to grupper av linjer viser innholdet i QS 17 og QS 18. Linjene som viser de korte (TS) og de dobbelte (DS) registrene vil alltid være riktig innstilt, men dette er ikke alltid tilfelle med de linjene som viser QS 17 og QS 18. Det er imidlertid ordnet slik at den øverste linjen i hver av de to gruppene alltid inneholder en "lik" minor-cycle. Når en trykker på "M.c. slip", vil linjene i QS-gruppene bli skiftet rundt to plasser av gangen. Når linjene på høyre monitor er riktig innstilt, vil også linjene på venstre monitor være riktig innstilt.

De to monitors har to innstillingsknapper hver; med disse kan en regulere skarpheten (Focus) og lysstyrken (Brilliance).

1.3. Output Staticiser

Vanligvis vil en sørge for at de resultatene og opplysningene som en ønsker ut fra DEUCE, blir punchet på hullkort. I enkelte tilfelle når en bare er interessert i et enkelt ord fra DEUCE, kan en få overført dette fra hurtigminnet til kontrollbordet for avlesing. Ordet må da avleses på en rekke á 32 lamper som kalles O u t p u t S t a t i c i s e r - l a m p e n e.

En kan avlese innholdet av hvilket som helst ord i hurtigminnet på kontrollbordet ved å sende ordet til D e s t i n a t i o n 29. De OS-lampene som tilsvarer binære enere blir tent, mens de øvrige blir uforandret. Dette betyr at hvis en ikke har sørget for å slukke samtlige OS-lamper før en

overføring til D 29 finner sted, kan en ikke være sikker på at det ordet som er representert på OS-lampene er en nøyaktig kopi av det ordet som er overført til D 29. OS-lampene kan slukkes ved hjelp av en ordre eller manuelt ved å bruke "Clear OS"-bryteren. Trigger-ordren "8-24" bevirker at samtlige OS-lamper blir slukket. Til høyre for OS-lampene er det en "Clear OS"-bryter hvormed en kan slukke alle lampene. Denne bryteren er slik at den springer tilbake til normalstillingen så snart den slippes. Selv om en holder bryteren i nedre posisjon, har ikke dette noen varig virkning, fordi bryteren er koplet slik at den slukker OS-lampene bare en gang for hver gang den trykkes ned.

Overføring av et ord til Destination 29 og OS-lampene bør ikke foregå mens det passerer kort i punchebanen, fordi Destination 29 vil da stå i forbindelse med puncheknivene også, slik at innholdet av ordet blir punchet i den raden som for øyeblikket ligger under knivene. Da D 29 står i forbindelse med både OS-lampene og puncheknivene, kan en se at de ordene som blir sendt til D 29 for punching også kommer på OS-lampene, men ordene vil være synlige på lampene så kort tid at det i praksis er umulig å avlese disse. OS-lampene blir nemlig slukket like etter at hver rad i kortet er punchet.

1.3.1. Output via kontrollbordet

Ordren "M-29" vil overføre en kopi av innholdet av lagerplass M i hurtigminnet til Output Staticiser-lampene via Output Staticiser (D 29). Output Staticiser (OS) har til oppgave å omforme serierepresentasjonen av et ord til parallellrepresentasjon.

Vanligvis vil det ikke være noen grunn til å la ordren "M-29" være stoppordre, derimot kan det ofte være en fordel om neste ordre i programmet er stoppordre, slik at DEUCE stopper når "M-29" er utført og neste ordre er overført til TS COUNT. Operatøren kan da bruke den tid han trenger til å lese av ordet på lampene. Når ordet er avlest, settes programmet i gang igjen med en "Single-Shot".

E k s.:

En er interessert i å avlese innholdet av TS 13 på OS-lampene (innholdet av TS 13 kunne også vært avlest på øverste linje i venstre monitor). En får da følgende ordrer:

8-24	(Slukk alle OS-lampene)
13-29	
(neste ordre) x	(Stoppordre)

1.4. Input Dynamiciser

Ved hjelp av Input Dynamiciser kan operatøren overføre et enkelt ord til hurtigminnet fra kontrollbordet. Ordet som skal over-

føres, settes opp på en rekke med 32 lamper som kalles Input Dynamiciser-lampene, og vil da være tilgjengelig fra Source 0 for overføring til enhver Destination. Dette vil imidlertid bare være tilfelle når det ikke er kortbevegelse i lesebanen i 528, for når det er tilfelle, er nemlig ID-lampene ikke effektive lenger, og Source 0 gir da de ordene som er punchet i de radene som passerer lesebørstene.

Foruten de 32 ID-lampene som er nevnt, er det 32 ID-brytere, en "Clear ID"-bryter, 32 "Special-Number"-brytere og en "Special Number $\pm$ "-bryter. Ved hjelp av de 32 ID-brytere kan de enkelte ID-lampene tennes eller slukkes. Hvis en ID-bryter blir presset nedover, blir tilsvarende ID-lampe tent; blir bryteren presset opp, slukkes lampen. Fra begge stillingene vil bryteren springe tilbake til normalstilling så snart den slippes. Til høyre for ID-lampene er "Clear ID"-bryteren. Denne bryteren kan også presses både opp og ned; i begge tilfelle blir alle ID-lampene slukket. Forskjellen mellom de to stillingene er at fra den nedre stilling vil bryteren springe tilbake til mellomstillingen av seg selv, mens den derimot kan bli stående i øvre stilling. Når denne bryteren står i øvre stilling, vil en ID-lampe bare lyse så lenge tilsvarende ID-bryter holdes nede.

De 32 "Special Number"-bryterne brukes når en vil sette opp et bestemt ord eller mønster på ID-lampene. Dette ordet eller mønsteret setter en da først opp på "Special Number"-bryterne, og overføres deretter til ID-lampene ved å presse "Special Number $\pm$ "-bryteren opp eller ned. De 32 "Special Number"-bryterne har hver to posisjoner: opp og ned, men ingen mellomstilling. Disse bryterne virker på følgende måte: Når "Special Number $\pm$ "-bryteren presses ned (-), tenner de ID-lampene som står rett overfor de "Special Number"-bryterne som er i øvre posisjon. Når "Special Number $\pm$ "-bryteren presses opp (+), tenner de ID-lampene som er rett overfor de "Special Number"-bryterne som står i nedre posisjon. Denne merkelige motsetningen har ingen annen betydning enn at det kanskje er bedre å huske. Uansett hvordan de 32 "Special Number"-bryterne står, vil en få tent alle ID-lampene ved først å presse "Special Number $\pm$ "-bryteren opp (+) og deretter ned (-).

1.4.1. Input via kontrollbordet

Når det ikke er kortbevegelse i IBM 528, står ID-lampene i forbindelse med Source 0 (Input Dynamiciser). Ordren "O-M" vil da overføre ordet på ID-lampene til lagerplassen M i hurtigminnet. Det som skjer, er i korthet følgende: Impulsene fra ID-lampene går via Input Dynamiciser (ID) til lagerplassen M i hurtigminnet. ID har til oppgave å omforme parallellrepresentasjonen på ID-lampene til et DEUCE-ord av standard størrelse og form.

Ofte vil det være en fordel om ordren "O-M" er stoppordre, slik at DEUCE stopper når denne ordren er kommet inn i TS COUNT, men før den er utført.

Operatøren kan da sette opp ordet på ID-lampene og deretter sette DEUCE i gang igjen med en Single-Shot.

1.5. Styring av maskinens hastighet

Til en viss grad kan en styre maskinens hastighet ved hjelp av: Stopp-bryteren, "Single-Shot"-bryteren, "Release"-bryteren og "One-Shot"-skiven.

1.5.1. Stopp-bryteren

Stopp-bryteren kan stå i en av tre stillinger - øvre, nedre eller mellomstilling:

- a) Når bryteren står i øvre stilling (merket "N o r m a l"), går maskinen med vanlig hastighet og vil ikke stoppe før den kommer til en stoppordre.
- b) Med bryteren i mellomstilling (merket "S t o p") oppfatter maskinen alle ordrer som stoppordrer, dvs. at DEUCE ikke kan skille mellom vanlig ordre og stoppordre.
- c) Når bryteren står i nedre stilling (merket "A u g. S t o p"), oppfatter maskinen alle ordrer som stoppordrer, dvs. som når bryteren står i "Stop"-stilling. Forskjellen ligger i at når bryteren står i nedre stilling, blir ikke ekte stoppordre ($P\ 32 = 0$) utført selv om en bruker "Single-Shot"-bryteren, fordi denne ikke er effektiv før "Release"-bryteren er presset ned. I slike tilfelle når "Single-Shot"-bryteren ikke er effektiv, lyser en rød lampe rett over "Stop"-bryteren.

"Release"-bryteren står normalt i mellomstilling; den kan imidlertid presses ned (med den virkning som er nevnt foran), men vil da springe tilbake til mellomstillingen så snart den slippes.

1.5.2. "Single-Shot"-bryteren

"Single-Shot"-bryteren står normalt i mellomstilling; den kan presses ned, og det vil da bli sendt ut ett "Single-Shot"-signal. Bryteren kan ikke stå i denne stillingen, men går tilbake så fort en slipper den. "Single-Shot"-bryteren kan imidlertid også presses opp, og i denne øvre stillingen kan bryteren stå så lenge en måtte ønske. Når bryteren står slik, blir det sendt ut en serie av Single-Shots (ca. 600 i minuttet).

Ved hjelp av "One-Shot"-skiven kan en produsere opptil 10 "Single-Shots". Slår en f.eks. 5 på skiven, blir det sendt ut 5 Single-Shots. (Slår en 0, blir det sendt ut 10 Single-Shots.)

Hverken "Single-Shot"-bryteren eller "One-Shot"-skiven er effektive når det er kortbevegelse i IBM 528.

1.6. "Go"-lampen

Når denne lampen lyser, betyr det at DEUCE utfører en serie av såkalte "Go"-ordrer ($P\ 32 = 1$). Så snart en stoppordre kommer inn i TS COUNT eller "Stop"-bryteren blir satt på "Stop" eller "Aug. Stop", vil lampen slukne.

1.7. "TCA"- og "TCB"-lampene

TCA- og TCB-lampene på kontrollbordet vil lyse når henholdsvis TCA eller TCB er "på".

1.8. "Instruction-Staticiser"-lampene

IS-lampene er en rekke med 15 lamper som viser $P\ 1\ (A)$, NIS, Source, Destination og $P\ 15$ i den ordren som befinner seg i TS COUNT, men som ennå ikke er utført. En vil ikke kunne lese av disse lampene unntatt når DEUCE står på "Stop" eller "Aug. Stop" eller når ordren i TS COUNT er stopp-ordre.

1.9. "Transtim"- og "Timci"-bryterne

1.9.1. "Transtim"-bryteren

Når "Transtim"-bryteren (merket CONT TT) står i nedre posisjon, blir det sendt ut en serie Transtim-signaler. Transtim-signalet sørget, som før nevnt, for at den operasjonen som er spesifisert ved Source og Destination i ordren i TS COUNT, blir utført.

1.9.2. "Timci"-bryteren

Når "Timci"-bryteren (merket TCI) står i nedre stilling, blir en serie med TCI-signaler sendt ut. TCI-signalet sørger for at en ordre går inn i TS COUNT fra den DL som NIS i ordren i TS COUNT spesifiserer eller fra det register som Source i Destination O-ordrer spesifiserer. Så lenge Timci-bryteren står nede, slik at en serie TCI-signaler blir sendt ut, vil en rekke ordrer bli sendt til TS COUNT. Den siste ordren som blir overført til TS COUNT før Timci-bryteren blir satt i mellomstilling, blir stående i TS COUNT, og en vil også finne denne ordren igjen på IS-lampene.

1.10. "External Tree"-bryteren

Normalt står denne bryteren i mellomstilling, den har imidlertid også en øvre og en nedre stilling. Når bryteren står i nedre posisjon, er det utstyret som kalles "External Tree" i virksomhet, mens "Request Stop"-utstyret er aktivt når bryteren står i øvre posisjon.

1.10.1. External Tree

Dette utstyret virker på følgende måte: Den ordren som står i TS COUNT blir ignorert, i stedet blir den ordren utført som er satt opp på et sett med 15 IS-brytere. Disse IS-bryterne står rett under IS-lampene på kontrollbordet.

Når en skal benytte dette utstyret, er det nødvendig at ordrene i TS COUNT hele tiden blir behandlet som stopp-ordrer; dette oppnår en ved å sette Stop-bryteren på "Stop". Framgangsmåten blir da slik:

- a) Sett Stop-bryteren på "Stop" (mellomstilling).
- b) Sett opp den ordren på IS-bryterne som skal erstatte ordren i TS COUNT.
En må her være oppmerksom på at en ikke kan få forandret Wait og Time i TS COUNT ved hjelp av IS-bryterne.
- c) Sett "External Tree"-bryteren i nedre stilling.
- d) Gi en Single-Shot eller press Transtim-bryteren (CONT TT) ned.
- e) Sett "External Tree"-bryteren i mellomstilling. Når en gjør dette, blir P 16 i TS COUNT satt til 0.

I punkt d) er det nevnt at en kan bruke enten "Single-Shot" eller CONT TT. Hvis Single-Shot blir brukt, blir neste ordre hentet inn i TS COUNT på normal måte, dvs. fra den delay-linje som NIS på IS-bryterne angir og Time fra TS COUNT. Blir derimot CONT TT benyttet, blir ikke den opprinnelige ordren i TS COUNT ødelagt av ordren på IS-bryterne (unntatt når det er en Destination 0-ordre), slik at så snart ordren på bryterne er utført ved hjelp av External Tree, kan programmet fortsette. I visse tilfelle kan imidlertid ikke denne metoden benyttes, nemlig:

- a) når ordren på IS-bryterne er lang og operasjonslengden er av betydning,
- b) "trigger"-ordrer (dvs. ordrer med Destination 24),
- c) ordrer med Destination 0.

I disse tilfellene må Transtim- og TCI-signalene komme samtidig. (Dette kan bare gjøres når en bruker External Tree med lang karakteristikk ($C = 1$) og en Single Shot.)

E k s.:

Nullstilling (tømming) av TS COUNT kan gjøres slik:

- a) Sett "0,30-0,1" på IS-bryterne.
- b) Sett "External Tree"-bryteren i nedre stilling.
- c) Gi en Single-Shot.
- d) Sett "External Tree"-bryteren i mellomstilling.

E k s.:

En kan undersøke innholdet av et bestemt spor på trommelen uten å miste noe av programmet eller annet i maskinen, forutsatt at det er et spor på trom-

melen som ikke er benyttet av programmet. Framgangsmåten blir slik:

- a) Noter hvilke blokker (X.Y) som henholdsvis lese- og skrivehodene er innstilt på.
- b) Overfør innholdet av DL 11 til et spor (A/B) som ikke blir benyttet av programmet, ved å utføre ordrene: "O,A-31,1" og "O,B-30,1". Bruk External Tree og CONT TT.
- c) Hent ned det sporet (C/D) som en er interessert i å undersøke. Dette gjør en ved å få utført ordrene "O,C-31,0" og "O,D-30,0" ved hjelp av External Tree og CONT TT.
- d) I DL 11 står nå det sporet som en er interessert i å undersøke.
- e) Når en er ferdig med undersøkelsen, hentes det gamle innholdet av DL 11 tilbake ved ordrene

O,A-31,0 External Tree, CONT TT

og O,B-30,0 " " , " "

- f) Sett lese- og skrivehodene tilbake ved ordrene

O,X-31,0 External Tree, CONT TT

og O,Y-31,1 " " , " "

Deretter kan en fortsette programmet.

1.10.2. Request Stop

Ved hjelp av dette utstyret kan en få maskinen til å stoppe på en ordre med en spesiell NIS, Source, Destination eller Characteristic eller kombinasjon av disse. Framgangsmåten er vanligvis slik:

- a) Stopp maskinen (Stopp-bryteren på "Stop").
- b) Sett den NIS, Source, Destination og/eller Characteristic på IS-bryterne som en ønsker å stoppe på. Sett en eller flere av de fire "Request Stop"-bryterne i nedre posisjon. "Request Stop"-bryterne står rett over de første IS-lampene og er betegnet som NIS, S, D og L. DEUCE vil undersøke de delene av ordrene i TS COUNT hvis tilsvarende "Request Stop"-bryter står i nedre stilling. Hvis f.eks. bare "Request Stop"-bryteren som er merket S, er i nedre stilling, undersøker DEUCE bare Source i de ordrene som blir overført til TS COUNT. Den vil da sammenlikne Source i TS COUNT med den Source som er satt opp på IS-bryterne, og vil stoppe når en ordre kommer inn i TS COUNT som har samme Source som på IS-bryterne.
- c) Sett "External Tree"-bryteren i øvre stilling (merket Request Stop).
- d) Sett maskinen i gang igjen ved å sette Stop-bryteren til NORMAL eller ved stadige Single-Shots. DEUCE vil da utføre ordrene i programmet i rekkefølge inntil den kommer til en ordre som tilfredsstiller de kravene som er satt.
- e) Når DEUCE stopper på en ordre som tilfredsstiller kravene for Request Stop, kan en ikke få maskinen i gang igjen før "Request Stop"-betingelsene blir opphevet eller endret.

Hvis en har satt opp ordren "4,13-25,1" på IS-bryterne og alle "Request Stop"-bryterne er i nedre posisjon, vil ikke DEUCE stoppe før en ordre kommer inn i TS COUNT som ser slik ut: "4,13-25,1". Hadde bare de to "Request Stop"-bryterne merket S og D vært i nedre stilling, ville DEUCE stoppe på første "13-25"-ordre uansett NIS og Characteristic. En ser altså at "Request Stop" blir mer selektivt når flere "Request Stop"-brytere er i nedre stilling.

En må være oppmerksom på at Trigger-ordrene (ordrer med Destination 24) ofte ikke kan brukes til å stoppe DEUCE på ved hjelp av Request Stop.

Når DEUCE har stoppet ved en Request Stop, kan en komme videre i programmet på flere måter:

- a) En kan enten oppheve Request Stop ved å sette "External Tree"-bryteren i mellomstilling eller
- b) sette alle "Request Stop"-bryterne i mellomstilling.
- c) Hvis ikke alle "Request Stop"-bryterne er i nedre stilling, kan en eller alle de bryterne som er i mellomstilling, settes i nedre stilling. (Dette vil imidlertid ikke alltid bringe DEUCE noe videre i programmet.)
- d) En kan forandre oppsettet på IS-bryterne.

I alle disse tilfellene bør en gjøre forandringen på bryterstillingene på kontrollbordet mens Stopp-bryteren står på "Stop". Når en så gjør en av de forandringene som er nevnt foran, vil den ordren som står i TS COUNT, og som DEUCE stoppet på, bli utført, men neste ordre vil bli oppfattet som en stopp-ordre.

1.11. Tvunget vegvalg

På kontrollbordet er det to brytere som en kan bruke når en vil tvinge DEUCE til å gå bestemte veger i programmet. Disse bryterne er merket TIL og DISCRIM.

1.11.1. TIL-bryteren

Denne bryteren har tre stillinger:

- a) mellomstilling: ordren 2-24 blir utført på normal måte,
- b) nedre stilling: TIL-signalet sendes kontinuerlig, slik at hvis ordren "2-24" er lagret i m.c.(m) og har Time = t, blir neste ordre hentet fra mc. (m + t + 3),
- c) øvre stilling: TIL-signalet sendes ikke, dvs. ordren "2-24" som er lagret i mc. (m) og har Time = t, sørger for at neste ordre blir overført i mc. (m + t + 2).

1.11.2. DISCRIM-bryteren

DISCRIM-bryteren har også tre stillinger:

- a) mellomstilling: ordrer med D 27 og D 28 utføres på vanlig måte,
- b) nedre stilling (merket ON): etter en ordre med D 27 eller D 28 blir neste ordre valgt som om et negativt tall blir sendt til D 27 eller et tall forskjellig fra null til D 28,
- c) øvre stilling (merket OFF): etter en D 27- eller D 28-ordre blir neste ordre valgt som om et positivt tall blir sendt til D 27 eller null blir sendt til D 28.

1.12. "Clear Store"-bryteren

Ved hjelp av "Clear Store"-bryteren kan en få tømt eller nullstilt hele hurtigminnet i DEUCE. Denne bryteren er laget slik at den først må trekkes ut og deretter presses ned før den får virkning. Når bryteren slippes, vil den springe tilbake til normalstilling.

1.13. Alarm

Trigger-ordren "7-24" setter i gang en alarm (summing) og tenner samtidig en lampe på kontrollbordet. Hvis en ønsker å stoppe alarmen, presses Alarm-bryteren (merket "Clear Alarm") ned. Samtidig med at summingen slutter, vil også alarm-lampen (like over Alarm-bryteren) slukne. Alarm-bryteren har tre stillinger: normalt står den i mellomstilling, i nedre stilling stoppes alarm og lampen slukkes. Fra denne nedre stilling vil bryteren automatisk springe tilbake til mellomstillingen. Settes bryteren i øvre stilling, vil alarmen stanse, men bare så lenge bryteren står i denne stillingen. Denne bryterstillingen slukker ikke alarmlampen.

1.14. "Read"- og "Punch"-bryterne

På kontrollbordet er det tre brytere hvormed en kan gi eller annullere lese- og puncheordrer. Disse er merket Read, Single Read og Punch.

En kan gi en leseordre (tilsvarende "12-24") fra kontrollbordet ved å presse Read-bryteren ned. Hvis en presser denne bryteren opp, blir en eventuell leseordre annullert. Hvis en presser Single Read-bryteren ned, får en lest et kort.

Punch-bryteren har tilsvarende virkning som Read-bryteren. Presses den ned i nedre stilling, blir det gitt en puncheordre ("10-24"). Presses derimot denne bryteren opp i øvre stilling, blir eventuell puncheordre annullert.

Alle disse tre bryterne er laget slik at de vil springe tilbake til mellomstilling så snart de slippes.

En lampe rett over Read-bryteren blir tent når en leseordre blir gitt

enten i programmet eller manuelt ved hjelp av Read-bryteren. En tilsvarende lampe over Punch-bryteren lyser når en puncheordre er gitt.

Lampene som er merket "Paper Tape Read" og "Paper Tape Punch" er ikke aktive så lenge papirbånd-utstyr ikke er koplet til.

1.15. Programme Display

Ved hjelp av dette utstyret kan en få punchet ordrene i et program i den rekkefølge som de blir utført av DEUCE. Det blir punchet en ordre pr. rad, altså 12 ordrer i hvert kort.

Når en skal bruke dette utstyret, kan en gå fram slik:

- a) En stopper maskinen i et punkt i programmet som ligger like foran de ordrene en vil undersøke.
- b) Stopp-bryteren settes til AUG.STOP.
- c) Blanke kort kjøres inn i punchebanen i 528.
- d) En presser ned Programme Display-bryteren, en lampe over bryteren blir tent og lyser så lenge Programme Display varer.
- e) Punching av ordrer i kort vil da fortsette inntil
 - i) en stoppordre. For å komme videre med Programme Display må først Release- og deretter "Programme Display"-bryteren presses ned.
 - ii) Operatøren stopper Programme Display ved å presse bryteren opp.
 - iii) Puncheordren blir annullert ved "9-24" eller manuelt. Hvis en ønsker å fortsette, kan en bare presse Punch-bryteren ned.
 - iv) Punchingen stopper på grunn av tomt kortmagasin, full kortlomme eller stoppknappen på 528 har vært brukt. I slike tilfelle vil Display fortsette så snart kort kjøres inn i punchebanen igjen.

Når en benytter Programme Display, bør en alltid være oppmerksom på at den ordren som står i TS COUNT når en starter Display, ikke blir punchet. Den første ordren som punches, er neste ordre. Den ordren som står i TS COUNT når en starter, bør ikke være lang (P 15 = 1), fordi det første kortet som da blir punchet og som skulle inneholde de 12 første ordrene, blir helt uforståelig. Når Programme Display slutter, vil den siste ordren som er punchet, være den som står i TS COUNT. Denne ordren er imidlertid ikke utført.

1.16. Magnettrummel-lampene

På kontrollbordet er det to sett á 4 lamper som angir hvilke blokker lese- og skrivehodene er innstilt på. Dessuten er det et sett med 4 lamper som gir Source i siste trommel-overførings-ordre, og 2 lamper som forteller om denne ordren var en skrive- eller leseordre.

Lese- og skrivehodenes plassering kan avleses binært på to sett á 4 lamper som er betegnet som "Read Position" og "Write Position". De 4 lampene som angir Source i siste trommeloverføringsordre, er betegnet "Head". Det er dessuten to lamper som er kalt "Write" og "Read". "Write"-lampen lyser hvis siste trommeloverføring var skriving på trommelen (P 15 = 1), mens "Read"-lampen vil lyse hvis siste trommeloverføring var lesing fra trommel (P 15 = 0).

1.17. Diverse lamper

1.17.1. Plug Board Alarm

En lampe som er merket "Plug Board Alarm" vil lyse så lenge det ikke er standardpluggen pluggbord i 528, dvs. i virkeligheten lyser lampen når det ikke er pluggen kontakt mellom to bestemte posisjoner på pluggbordet.

1.17.2. Not Ready

Lampen som er merket "Not Ready" lyser når DEUCE står i stopp-stilling, dvs. at Stopp-bryteren står på STOP eller AUG. STOP eller "External Tree"-bryteren står på REQ. STOP.

1.17.3. Marginal Check

"Marginal Check"-bryteren og lampen brukes under det daglige vedlikeholdet av DEUCE og under feilsøking av teknikeren. Når "Marginal Check"-bryteren blir satt i øvre eller nedre stilling, har dette til virkning at det elektroniske systemet i DEUCE blir belastet i den ene eller andre retningen. Hvis DEUCE arbeider tilfredsstillende enten bryteren står i øvre eller nedre stilling, betyr det at en kan vente at den vil arbeide helt pålitelig med bryteren i mellomstilling. "Marginal Check"-lampen lyser når bryteren ikke står i mellomstilling.

1.17.4. Clear Sync Alarm

Lampen rett over "Clear Sync Alarm"-bryteren lyser når magnettrommelen ikke vil la seg synkronisere med hurtigminnet eller når den faller ut av synkroniseringen. Samtidig høres en summende lyd fra en såkalt "Buzzer".

Når "Clear Sync Alarm"-knappen settes i øvre posisjon, stopper summingen, men lampen vil fortsatt lyse. En kan få slukket lampen og stoppet summingen samtidig ved å presse bryteren ned, forutsatt at magnettrommelen er kommet i synkronisering med hurtigminnet igjen. Bryteren kan ikke stå fast i nedre stilling; så fort den slippes, springer den tilbake til mellomstillingen.

1.17.5. Mag. Alarm

Når magnettrommelen ikke er koplet til eller hvis noen av bryterne i magnettrommel-kabinettet står i gal stilling, lyser den lampen som er merket

"Mag. Alarm". Lampen slukkes automatisk når bryterne i trommel-kabinettet blir satt riktig; disse bryterne må imidlertid bare opereres av teknikeren.

1.17.6. Fan

De bryterne og lampene som er merket "FAN", kontrollerer gjennomblåsing av luft i DEUCE. En av lampene skal alltid lyse, den grønne lampen hvis ventilasjonen er i gang, mens den røde indikator-lampen skal lyse hvis ventilasjonen ikke er koplet til. En setter gjennomblåsing i gang ved å trykke på knappen merket ON, og slår den av ved å trykke på OFF-knappen.

1.17.7. Heaters

Når en trykker på ON-knappen, tennes den grønne lampen og viser dermed at glødningen er på og at viftene i IBM 528 er startet. En stopper glødningen og viftene i 528 ved å trykke på knappen merket "OFF & RESET", da vil den røde lampen bli tent.

1.17.8. H.T.

Bryterne og lampene som er merket H.T., dirigerer tilførsel av høyspenning til DEUCE (unntatt "IBM 528"). Bryterne og lampene virker på tilsvarende måte som for "Heaters" og "Fan".

1.17.9. Hollerith

Benevnningen her er litt misvisende, fordi tidligere utgaver av DEUCE har benyttet engelske Hollerith-maskiner som input- og output-enhet. Disse bryterne og lampene dirigerer imidlertid tilførselen av høyspenning til IBM 528, og virker på tilsvarende måte som for "H.T.".

1.17.10. Overbelastningsindikatorer

Disse lampene vil lyse hvis det har vært overbelastning et eller annet sted i maskinen. Når operatøren ser at en eller flere av disse lampene blir tent, må dette meldes fra til teknikeren.

2. IBM 528

Inn- og utlesing i DEUCE foregår helst ved hjelp av hullkort. Program og data som skal inn i maskinen, må først overføres til hullkort og deretter gjennom en hullkortleser til DEUCE. Resultater og andre opplysninger som en ønsker ut av DEUCE, punches i hullkort i en hullkortpunch som er koplet til DEUCE. Disse hullkortene kan senere behandles i vanlige hullkortmaskiner eller sendes inn igjen i DEUCE.

DEUCE bruker I B M 5 2 8 som kombinert hullkort-leser og -punch.

2.1. Generell beskrivelse av IBM 528

IBM 528 er egentlig en reproducerer med to helt uavhengige kortbaner. Lesebanen har to lesestasjoner, hvorav bare den første er i bruk; det er ved denne stasjonen at opplysningene blir avfølt og overført til DEUCE. Den andre kortbanen, punchebanen, har en punche- og en lesestasjon, men bare punche-stasjonen er aktiv.

De to kortbanene i 528 kan arbeide helt uavhengig og med forskjellig hastighet. Kortene i lesebanen blir normalt, dvs. når det foregår bare lesing i 528, avlest med en hastighet av 200 kort/min. Ved hjelp av en bryter på 528 kan denne hastigheten reduseres til det halve. Punchingen blir alltid utført med en korthastighet av 100 kort/min. I de tilfellene det er lesing og punching samtidig (dual operation), dvs. at et kort blir avlest i lesebanen samtidig med at et kort blir punchet i punchebanen, blir hastigheten i lesebanen automatisk redusert til 100 kort/min., slik at operasjonene i de to kortbanene blir fullstendig synkroniserte.

Hullkortene som brukes i 528 er av standard størrelse med 80 kolonner og 12 rader. Kolonnene nummereres fra 1 til 80 fra venstre, mens radene kalles for Y, X, 0, 1, 2, 3, 4, 5, 6, 7, 8 og 9, regnet fra toppen av kortet. Kortene plasseres i kortmagasinene med klisjésiden ned og Y-raden inn mot kortmunnen.

Innlesing og punching av hullkortene kan foregå etter to helt forskjellige prinsipper som en gjerne kaller for den *binære* og den *desimale* metoden. Programmene blir normalt lest inn etter den binære metoden, mens data leses inn og punches etter den desimale metoden. Det er imidlertid intet i veien for at program kan leses inn desimalt og at data kan leses og punches binært. Det er endog mulig, om ønskelig, å lese inn opplysninger fra et kort etter begge metodene samtidig; noe tilsvarende er umulig ved punching.

2.2. Den binære metoden

Etter binær-metoden kan bare 64 av kortets kolonner avleses eller punches samtidig. Disse kolonnene kan imidlertid velges helt fritt innen kortet ved plugging; det er heller ingen sammenheng mellom de kolonnene som avleses og punches.

Hullkortene er som nevnt delt inn i 12 rader, og disse radene passerer lesebørstene og puncheknivene i denne rekkefølge: Y, X, 0, 1, 2, 3, 4, 5, 6, 7, 8 og 9. Kortene blir avlest eller punchet radvis. De 64 kolonnene i en rad inneholder opplysninger tilsvarende to ord á 32 bits, idet punchede kolonner og blanke representerer henholdsvis binære enere og nuller. En har altså her en tilsvarende parallell-representasjon som på ID- og OS-lampene på kontrollbordet.

En kaller de to feltene på kortet, som de to ordene leses fra eller punches i, for α -feltet og β -feltet. Det står en fritt å velge hvilke kolonner som skal danne de to feltene. Hvis en måtte ønske det, kan de samme kolonnene danne begge feltene eller deler av disse.

2.2.1. Binær innlesing

Ordren "12-24,0" setter kortene i lesebanen i bevegelse slik at en strøm av kort vil passere lesebørstene, men denne ordren alene vil ikke resultere i noen overføring av opplysninger fra kortene til DEUCE. Opplysningene må komme fra børstene som avføler kortene via Input Dynamiciser (Source 0). Denne har som nevnt til oppgave å omforme parallell-representasjonen av et ord til serie-representasjon.

Så snart kontrollorganet i DEUCE har mottatt leseordren "12-24,0", blir forbindelsen mellom ID og ID-lampene brutt; i stedet blir ID satt i forbindelse med lesebørstene. Ordren "0-M" vil da bevirke at et ord fra den kortraden som for øyeblikket ligger under lesebørstene, blir overført til lagerplass M i hurtigminnet. Ønsker en å lese begge ordene i en rad, dvs. både fra α -feltet og fra β -feltet, må en ha to ordrer:

$O_{\alpha} - M$

$O_{\beta} - N$

Den første ordren vil da overføre ordet fra α -feltet til lagerplassen M, og den andre ordren vil overføre ordet fra β -feltet til lagerplassen N i hurtigminnet.

Kortene beveger seg stadig under lesebørstene; derfor må "0-M"-ordrene bli utført på de tidspunktene da de enkelte radene ligger i en slik posisjon at avlesing kan foretas. For å kunne klare det, er det ordnet slik at DEUCE sender ut en serie av Single-Shots, én for hver rad. De enkelte Single-Shots blir sendt ut like før radene er klare for avlesing. Disse Single-Shots er det samme signalet som en kan produsere ved å bruke Single Shot-bryteren på kontrollbordet.

De to ordrene for avlesing av α - og β -feltene i en rad, blir da:

$O_{\alpha} - M, x$

$O_{\beta} - N$

Den første ordren er en stoppordre og blir ikke utført før kontrollorganet mottar en Single-Shot som forteller at raden er klar for avlesing. Den andre ordren må ikke være en stoppordre, idet den ordren da vil bli utført først ved Single-Shot for neste rad.

Etter Single-Shot for en rad vil den første ordren med Source 0 (unntatt de som har Destination 24, 30 eller 31) alltid gi innholdet i α -feltet. Den første ordren med Source 0 (med de unntak som er nevnt foran) vil dessuten sørge for at forbindelsen til ID blir flyttet fra α -feltet til β -feltet; denne flyttingen av forbindelsen starter 4 m.c. etter at den ordren som

etterfølger Source 0-ordren, er kommet inn i TS COUNT, og er avsluttet 19 m.c. etter denne tid. En vet da at hvis den etterfølgende ordren kommer inn i TS COUNT i m.c. (m), vil m.c. (m+4) være den siste minor-cycle som en kan ha en pålitelig overføring av ordet i α -feltet, og m.c. (m+20) er den første minor-cycle som en kan overføre ordet fra β -feltet. Hvis en forsøker å overføre mellom m.c. (m+5) og m.c. (m+19), kan en ikke være sikker på resultatet. En kan overføre opplysninger fra en rad inntil 4 1/2 m.s. etter Single-Shot for raden. Etter et intervall på 10 m.s. fra S.S. blir forbindelsen til ID automatisk flyttet tilbake til α -feltet.

E k s. 1: Les α -feltet og β -feltet så raskt som mulig.

l_0	0 - 14.x	(1.0 - 14.0.0.0.x)
l_2	0 - 15	(1.0 - 15.0.18.18)
l_{22}	neste ordre	

I dette eksempelet blir α -feltet overført til TS 14 i m.c. 2. Ordren "0-14.x" gir dessuten ordre om flytting av forbindelsen til ID; denne flyttingen starter i m.c. (2+4) = 6 og avsluttes i m.c. (2+19) = 21, slik at overføringen fra β -feltet tidligst kan foregå i m.c. 22.

E k s. 2: Les α - og β -feltet så tidlig og så sent som mulig.

l_{30}	0 - 14.x	(1.0 - 14.0.0.14.x)
l_{14}	0 - 17 ₂	(1.0 - 17.0.2.2)
l_{18}	0 - 18 ₂	(1.0 - 18.0.14.31)
l_{19}	1 - 1	(1.1 - 1.0.0.31)
l_{20}	1 - 1	(1.1 - 1.0.0.31)
l_{21}	0 - 1 ₁₅	(1.0 - 1.0.24.25)
l_{16}	neste ordre	

α -feltet blir i dette eksempelet lest inn to ganger, til TS 14 og QS 17₂ i henholdsvis m.c. 0 og 18. Flyttingen av forbindelsen til ID starter i m.c. 19 og varer til og med m.c. 1; derfor blir β -feltet første gang overført i m.c. 2. Siste gang blir β -feltet overført til DL 1₁₅ nøyaktig 4 1/2 m.s. etter S.S.

Hvis en ønsker å lese bare β -feltet fra en rad, må en ha en såkalt "dummy"-ordre med Source 0 slik at forbindelsen blir flyttet fra α - til β -feltet. En kan gjerne bruke ordren "0-0", men en må da sette $W \neq T$ i denne ordren, ellers vil α -feltet gå inn i TS COUNT som neste ordre.

E k s. 3: Innlesing av β -feltet.

l_0	$0 - 0.x$	$(W \neq T)$
l_3	$0 - 15$	$(W = 18)$
l_{23}	neste ordre	

Forbindelsen til ID blir her flyttet i m.c. 8 til og med m.c. 22, slik at β -feltet først kan overføres i m.c. 23.

Hittil har vi bare sett på innlesingen fra en rad i kortet. Skal en lese fra alle radene i kortet, må ordrer av typen "O-M" utføres på hver rad.

På grunnlag av den serien av Single-Shots som blir sendt ut under avlesingen av et kort, er det ikke mulig å skille mellom de enkelte radene i kortet. Normalt er det ikke nødvendig å kunne identifisere alle radene i kortet, men 9-raden, dvs. den siste raden, er det av stor interesse og helt nødvendig å kunne skille ut fra de andre. For å kunne klare dette sender DEUCE ut et signal som kalles T I L (Twelfth Impulse Line).

TIL starter like før Single-Shot for 9-raden og vedvarer en stund etter. Ved hjelp av ordren "2-24" kan en undersøke om TIL-signalet er til stede eller ikke. "2-24" som er en såkalt "Trigger", har som virkning at TIL blir sendt til Destination 28, slik at hvis "2-24" går inn i TS COUNT i m.c. (m) og har Time = T, blir neste ordre hentet inn i TS COUNT i m.c. (m+2+T) når TIL ikke er til stede (TIL = 0), mens neste ordre kommer fra m.c. (m+3+T) i de tilfellene TIL er til stede (TIL $\neq$ 0). Dette gjør en i stand til å velge en annen veg i programmet etter at siste raden i kortet er avlest.

E k s. 4: For å illustrere bruken av TIL kan en tenke seg følgende enkle oppgave: I et kort er det punchet 24 binære tall, $a_1, a_2, a_3, a_4, \dots, a_{12}$ i α -feltene og $b_1, b_2, b_3, b_4, \dots, b_{12}$ i β -feltene. En er interessert i å finne $\sum a, \sum b$ og $\sum (a+b)$ og plassere disse summene i DL 10_{0-2} .

Eks. 4 (forts.)

		<u>DL 1:</u>
l_{21}	12 - 24	m.c. 21: 1.12-24.0.31.31
l_{22}	5 - 24	" 22: 1. 5-24.0.0.0
l_{24}	30 - 21 (2)	" 23: 1. 0-22.0.30.0
l_{27}	0 - 22 ₂ ,x	" 24: 1.30-21.2.0.1
l_{23}	0 - 22 ₃	" 25: 1. 2-24.0.0.0
$l_{25} = 0$	2 - 24	" 26: 1.21-13.0.0.1
	$\neq 0$	" 27: 1. 0-22.0.1.26.x
l_{28}	9 - 24	" 28: 1. 9-24.0.0.0
l_{30}	21 - 10 ₀ (2)	" 29: 1.21-25.0.0.0
l_{26}	21 ₂ - 13	" 30: 1.21-10.2.0.26
l_{29}	21 ₃ - 25	" 31: 1.13-10.0.1.1
l_{31}	13 - 10 ₂	" 2: neste ordre
l_2	neste ordre	

I dette eksempelet har en brukt ordren "9-24" (i l_{28}). Denne ordren stopper all kortgjennomgang i 528. Kortbevegelsen i kortbanene stopper alltid i normalstilling, dvs. at ikke noe kort blir liggende under lesebørstene eller puncheknivene. Hvis denne ordren blir gitt på et tidspunkt da det ligger kort under lesebørstene, vil kortet fortsette inntil det har passert lesestasjonen, men det vil ikke bli sendt ut noen Single-Shots for de radene som passerer børstene etter "9-24" og heller ikke noe TIL-signal for dette kortet. En vil derfor ikke få avlest flere rader fra dette kortet. Selv om en ny "12-24" blir gitt før alle radene har passert børstene, vil ikke denne ordren ha noen virkning for dette kortet; først på neste kort blir TIL og Single-Shots sendt ut på vanlig måte.

Kortene i lesebanene kan også stoppes på en annen måte, nemlig ved en punching i en bestemt kolonne. Denne kolonnen kan velges ved plugging og kan være hvilken som helst av kortets 80 kolonner. Denne punchingen kalles CR (Clear Read) og stopper kortene i lesebanen på tilsvarende måte som "9-24". Single-Shot blir sendt ut for den raden som CR står i, men for de etterfølgende rader kommer ikke noen Single-Shots. Hvis CR står i en annen rad enn 9-radene, blir heller ikke noe TIL-signal sendt ut.

De 12 Single-Shots som blir sendt ut for hvert enkelt kort, følger etter hverandre med intervall på 18 m.s. mellom hver. Etter at en rad er avlest, kan derfor programmet utføre andre operasjoner i inntil 18 m.s. før neste rad

må avleses. Fra Single-Shot for 9-radene i et kort og til første Single-Shot på neste kort er det et tidsrom på 57 m.s. Dette tidsrommet kan brukes til annet arbeid. De tidsintervallene som her er oppgitt, gjelder bare når kortene i lesebanen går med full hastighet (200 kort/min.). Når hastigheten er redusert (100 kort/min.), vil intervallene være større. De nøyaktige spesifikasjoner av disse tidsrommene og andre tider av interesse vil en finne under 2.2.3.

2.2.2. Binær punching

Ordren "10-24" setter kortene i punchebanen i bevegelse slik at en serie av kort passerer punchestasjonen, men denne ordren alene vil ikke resultere i punching i de radene som passerer puncheknivene. De opplysningene som skal punches i radene, må sendes til Output Staticiser eller Destination 29. OS har som før nevnt til oppgave å omforme serierepresentasjonen av DEUCE-ord til parallell-representasjon.

Når kortene i punchebanen er i ro, står OS bare i forbindelse med OS-lampene på kontrollbordet, men så snart puncheordren "10-24" blir gitt, blir OS også satt i forbindelse med puncheknivene.

Single-Shots blir sendt ut på samme måte som under innlesing, én for hver rad. Selv om det foregår både lesing og punching i 528 på samme tid, blir det likevel bare sendt ut en serie med Single-Shots; denne gjelder da både for lese- og punchebanen fordi kortbevegelsene i kortbanene er helt synkronisert.

Skal en punche i begge feltene i en rad, trenger en tre ordrer:

$M_{\alpha} - 29, x$

8 - 24, 1

$N_{\beta} - 29$

Den første ordren vil ikke bli utført før kontroll-organet mottar en Single Shot som forteller at en rad ligger klar for punching. Ordren " $M_{\alpha} - 29, x$ " vil sørge for at innholdet av lagerplassen M i hurtigminnet blir punchet i α -feltet. Den neste ordren, "8-24, 1", bevirker at forbindelsen til OS blir flyttet fra α - til β -feltet; dette skjer momentant. Den siste av de tre ordrene vil da sørge for at innholdet i lagerplassen N blir punchet i β -feltet.

Ek s. 5: Punching i α - og β -feltet så raskt som mulig.

l_0 $M_{\alpha} - 29, x$ (1.M - 29.0.0.0.x)

l_2 8 - 24, 1 (1.8 - 24.1.0.0)

l_4 $N_{\beta} - 29$ (1.N - 29.0.0.0)

l_6 neste ordre

En kan sende noe til punching, dvs. til Destination 29, inntil 4 m.s. etter Single-Shot for raden.

E k s. 6: Punching av bare α -feltet (forsinket).

l_0 1 - 1,x Stoppordre ("dummy")

program < 4 m.s.

l_1 M - 29 Forsinket punching av α -feltet

E k s. 7: Punching av bare β -feltet (forsinket).

l_0 1 - 1,x Stoppordre ("dummy")

l_2 8 - 24,1

program < 4 m.s.

l_1 M - 29 Forsinket punching av β -feltet

Skal en punche flere rader i kortet, må ordrer av typen "M-29" utføres for hver rad. Heller ikke under punching er det mulig å identifisere de enkelte radene på grunnlag av Single-Shots; derfor bruker en også her TIL-signalet for å kunne skille ut 9-raden fra de øvrige. TIL undersøkes ved hjelp av ordren "2-24" på samme måte som ved avlesing.

Som før nevnt kan det foregå samtidig lesing og punching i 528. Hastigheten vil da i begge kortbanene være 100 kort/min. Under visse betingelser kan det hende at kortene i de to banene kommer ut av takt, nemlig når lese- og/eller puncheordren er forsinket, f.eks. fordi en må gjøre en del beregninger før en kan avgjøre om dette skal være en såkalt "dual operation", dvs. både lesing og punching. I slike tilfelle kan det hende at kortene i den ene kortbanen blir satt i gang til tross for at det ikke forekommer kort i magasinet i den andre kortbanen eller at den av andre grunner skulle være "not ready".

En kan imidlertid gardere seg mot at slike ting skal forekomme ved å følge visse bestemte regler:

1. Lese- og puncheordren som beordrer "dual operation", må utføres innen et intervall på 1 m.s.
2. Ordren "9-24" må utføres innen 13 m.s. etter S.S. for 9-raden på foregående kort. Denne ordre må dessuten komme før lese- og puncheordren under pkt. 1.

E k s. 8: En har et kort med 24 binærtall, to binærtall i hver rad:

a_i i α -feltet og b_i i β -feltet. En ønsker å lese inn dette kortet, beregne summen av de to binærtallene i hver rad ($a_i + b_i$) og punche denne summen i tilsvarende rad på et annet kort.

Eks. 8 (forts.)

		DL 1:
l_{28}	12 - 24	m.c. 0: 1. 0-13.0.0.0.x
l_{30}	10 - 24	" 1: N. 9-24.0.0.T
l_0	0 - 13,x	" 2: 1. 0-25.0.18.18
l_2	0 - 25	" 22: 1.13-29.0.0.0
l_{22}	13 - 29	" 24: 1. 2-24.0.0.6
l_{24}	2 - 24	" 28: 1.12-24.0.0.0
= 0		" 30: 1.10-24.0.0.0
l_1	9 - 24	

Ordren "9-24" stopper all kortbevegelse i 528, også i punchebanen. Blir denne ordren gitt mens det ennå er et kort under puncheknivene, vil ikke kortet stoppe før alle radene har passert puncheknivene, men for de radene som passerer etter "9-24" blir ikke sendt ut Single-Shots. Kommer "9-24" før 9-raden har passert, vil heller ikke noe TIL-signal bli sendt ut. I de radene som kommer etter "9-24", kan en ikke få punchet noe fordi forbindelsen mellom OS og puncheknivene blir brutt. Selv om en ny puncheordre, "10-24", blir gitt før alle radene har passert puncheknivene, vil ikke dette ha noen virkning for dette kortet; først på neste kort blir Single-Shots og TIL-signal sendt ut på normal måte igjen.

Mellom hver Single-Shot for et kort under punching er det et intervall på 38 m.s.; dette intervallet kan brukes til annet program. Mellom to påfølgende kort, dvs. mellom 9-raden på et kort og Y-raden på neste, er det et intervall på 116 m.s. De helt nøyaktige tidene for disse intervallene og andre tidsspesifikasjoner finner en under 2.2.3.

2.2.3. Tidsspesifikasjoner ved binærmotoden

I figur 1 på neste side er en tabell over de tidsspesifikasjoner som det er nødvendig å kjenne til når en vil lage et program som benytter seg av binær lesing og/eller punching. Tabellen er delt opp i 5 kolonner. De to første kolonnene angir fra og til hvilke operasjoner som tiden i de enkelte tilfelle er regnet, de tre siste kolonnene oppgir tiden i Major-cycles.

I det følgende skal en komme med en del kommentarer til tabellen.

a) Når kortene i lesebanen går med full hastighet, er opplysningene i en rad tilgjengelig fra børstene inntil 4 1/2 m.s. etter stopp-ordren (dvs. Single-Shot) for raden. Hvis hastigheten er redusert til det halve (100 kort/min.) enten automatisk, fordi det skal utføres samtidig punching, eller manuelt ved

Fig. 1: Tidsspesifikasjoner ved binær-metoden

	FRA:	TIL:	Lesing		Punching
			200	100	100
a.	S.S. for vilkårlig rad i kortet	Avlesing eller punching med bra resultat	4 1/2	9	4
b.	S.S. for en av de 11 første radene	få tak i neste S.S.	18	38	38
		miste neste S.S.	25	46	46
c.	S.S. for 9-raden	få tak i neste S.S.	57	116	116
		miste neste S.S.	70	135	135
d.	S.S. for 8-raden	TIL starter (minimum)	13	28	28
	TIL starter	S.S. for 9-raden (min.)	2	2	2
	" "	miste S.S. for 9-raden	7	14	14
	S.S. for 9-raden	TIL forsvinner (min.)	18	35	35
	" " "	" " (max.)	23	50	50
	TIL forsvinner	første S.S. neste kort (min.)	34	70	70
e.	S.S. for vilkårlig rad i kortet	"9-24" uten å få neste S.S.	13	30	30
	CR eller "9-24" i vilkårlig rad	ID-bryterne kan brukes	13	13	
	CR eller "9-24" på 9-raden	TIL-signalet forsvinner	23	23	23
f.	CR	Ny leseordre ("12-24")	0	0	
g.	S.S. på 9-raden	"9-24" uten ekstra kort	15	33	60
	" " "	"9-24" med "Missed Card Protection"	54	86	
h.	"9-24"	Ny "10-24" og/eller "12-24"	2	2	2
i.	S.S. for vilkårlig rad	punching i X-feltet på neste rad			20
j.	S.S. for vilkårlig rad	"9-24" uten punching på neste rad			15

hjelp av bryteren på 528, er den tilsvarende tiden 9 m.s. Under punching kan en overføre opplysninger til punching inntil 4 m.s. etter Single-Shot for raden.

b) En stoppordre som kommer inn i TS COUNT inntil 18 m.s. etter en Single-Shot (for en av de 11 første radene), vil bli utført på neste rads Single-Shot. Kommer derimot denne ordren fra 18 til 25 m.s. etter S.S., kan en ikke være sikker på om en S.S. har passert. En stoppordre som kommer mer enn 25 m.s. etter en av de første 11 radenes S.S., vil alltid komme etter at neste rads S.S. har passert. Under punching og lesing med redusert fart er de tilsvarende tider henholdsvis 38 og 46 m.s.

c) Etter S.S. for 9-raden i et kort må en sørge for at stoppordren for første raden i neste kort kommer inn i TS COUNT innen 57 m.s. for å være sikker på å få tak i S.S. for første raden. Kommer stoppordren fra 57 til 70 m.s. etter 9-radens S.S., kan en ikke være sikker på om en får tak i S.S. for første raden eller ikke. Går det mer enn 70 m.s. fra siste S.S. i et kort og til stoppordren, kan en være helt sikker på å miste første S.S. i neste kort. De tilsvarende tider ved 100 kort/min. er 116 og 135 m.s.

d) Fra S.S. for 8-raden går det minst 13 m.s. før TIL-signalet starter (28 m.s. ved 100 kort/min.).

Fra TIL starter er det garantert at det går minst 2 m.s. før S.S. for 9-raden; en stoppordre som kommer innenfor denne tiden vil altså få tak i S.S. for 9-raden. Kommer imidlertid denne stoppordren mer enn 7 m.s. etter starten av TIL, vil en ikke få tak i 9-radens S.S. Mellom 2 og 7 m.s. kan en ikke være sikker på hva som skjer. De tilsvarende tidene ved redusert fart er 2 og 14 m.s.

Regnet fra S.S. for 9-raden varer TIL-signalet minst 18 m.s., men aldri mer enn 23 m.s. (35 og 50 m.s. ved 100 kort/min.).

Fra TIL-signalet forsvinner går det minst 34 m.s. (70 m.s. ved 100 kort/min.) før S.S. for Y-raden på neste kort.

e) Kommer en "9-24"-ordre innen 13 m.s. etter S.S. for en av de 11 første radene i kortet, blir det ikke sendt ut Single-Shots for de etterfølgende rader i kortet. Går det lengre tid enn 13 m.s., vil Single-Shot komme for neste rad, men ikke for de øvrige. Den tilsvarende tid ved en hastighet á 100 kort/min. er 30 m.s.

Det går 13 m.s. før ID-bryterne kan brukes når en har stoppet kortene ved en CR eller "9-24,x" på en vilkårlig rad i kortet. Denne tiden er den samme uansett om kortene leses med full eller redusert hastighet.

Blir kortene i lese- og/eller punchebanen stoppet ved en "9-24,x" eller CR på 9-raden, vil TIL-signalet være borte etter maksimum 23 m.s.

f) En CR i hvilken som helst rad i et kort kan annulleres umiddelbart med en "12-24"-ordre, slik at det ikke blir noen stopp i kortgjennomgangen, men det vil ikke komme noen flere Single-Shots eller TIL for det kortet som inneholder CR.

g) En kan risikere at et ekstra kort passerer lesebørstene hvis en gir en "9-24"-ordre senere enn 15 m.s. etter S.S. for 9-raden. Hvis dette skulle hende, er det imidlertid en innretning som kalles "Missed Card Protection" som signaliserer dette og stopper all innlesing. Denne innretningen vil være aktiv inntil 54 m.s. etter S.S. for 9-raden. Dette er imidlertid minimumsgrensen, slik at det godt kan hende at kortene vil stoppe på grunn av "Missed Card" selv om det går over 54 m.s. fra S.S. fra 9-raden til "9-24". De tilsvarende tidene ved redusert fart i lesebanen er 33 og 86 m.s.

For å unngå ekstra kortmating i punchebanen bør "9-24" komme senest 60 m.s. etter S.S. for 9-raden. Overstiger denne tiden 60 m.s., kan en risikere å få et ekstra kort. Det er naturligvis ikke så alvorlig å få ekstra kortgjennomgang under punching som under innlesing; derfor er det ikke noen "Missed Card Protection" i punchebanen. Det kan imidlertid være ubehagelig å få ekstra kort i de tilfellene en benytter seg av kortnummer- og "job"-nummerpunching.

h) Hvis en etter å ha punchet og/eller avlest et kort gir ordren "9-24", må en ikke gi ordrer av typen "10-24" og/eller "12-24" før tidligst 2 m.s. etter. Hvis ordrene kommer nærmere hverandre, vil det kunne oppstå feil, idet ordren "9-24" ikke får noen virkning.

i) En har sett i 2.2.2. at overføringen av de opplysningene som skal punches i en rad, kan forsinkes inntil 4 m.s. etter S.S. for raden. Opplysninger kan også sendes til D 29 før S.S. og likevel bli punchet i α -feltet i neste rad. Hvis " M_{α} -29" kommer mer enn 20 m.s. etter S.S. for en rad, blir M_{α} punchet i α -feltet i neste rad. I slike tilfelle opptreer derfor D 29 som en lagerplass, men fra dette lageret kan bare overføres til puncheknivene.

j) Hvis en ikke ønsker punching i en rad, må ordren "9-24" gis innen 15 m.s. etter foregående rads S.S.

2.3. Den desimale metoden

Ved innlesing av data og punching av resultater bruker en som regel den desimale metoden. Denne metoden har mange fordeler framfor binærmotoden. De viktigste er:

a) Alle 80 kolonnene i hullkortet kan avleses og punches.

b) Beregninger og andre operasjoner i DEUCE kan foregå samtidig med og uavhengig

av innlesingen og punchingen. Dette er mulig på grunn av et såkalt
b u f f e r - m i n n e.

- c) Under desimalinnlesing blir desimalsifre og bokstaver automatisk omkodet til binærkodete desimaltall (6 bits pr. binærsiffer eller bokstav). Under punchingen skjer det motsatte; binærkodete desimaltall og bokstaver blir omkodet og punchet som desimaltall og bokstaver.

Desimalmetoden fører naturligvis også med seg noen ulemper, bl.a. blir hele DL 12 brukt som buffer-minne og kan derfor ikke brukes som vanlig minne. Dertil trenger en også som regel en del plass i hurtigminnet for midlertidig lagring av de kortene som skal punches eller nettopp er innlest. Dette vil vi se nærmere på senere i dette kapitlet.

2.3.1. Buffer-minne

I DL 12 som brukes som buffer-minne, lagres de binærkodete opplysningene som er avlest eller skal punches. Når en leseordre er gitt, vil innlesingen foregå helt automatisk. Etter hvert som opplysningene i kortet blir avlest, blir disse omkodet til binærform og lagret som 80 6-bits koder i den ene halvparten av DL 12. Programmet må siden sørge for at de enkelte binærkodete desimalsifrene blir samlet og kombinert slik at de danner fullstendige binær-tall. Dette kan imidlertid ikke gjøres før etter at 9-raden er avlest og registrert, fordi da først er alle opplysningene fra kortet plassert i sine endelige binærkoder i DL 12. Det normale er at disse binærkodene overføres til en annen delay-linje, f.eks. DL 10, for mellomlagring like etter at 9-raden er avlest, slik at innlesingen av et nytt kort kan settes i gang. Etter desimalmetoden må en nemlig gi ny leseordre for hvert kort som skal avleses.

Før punching kan settes i gang ved en puncheordre, må alle de opplysningene som skal punches, omregnes eller omkodes til 6-bits binærkoder og plasseres i DL 12. DEUCE vil da, etter å ha mottatt puncheordre, automatisk omkode disse binærkodene og punche dem som desimaltall eller bokstaver. Likesom leseordren gjelder også puncheordren bare for ett kort av gangen.

Buffer-minnene i DL 12 gjør det mulig å nytte nesten hele kort-cyclen til program. Under innlesing med full hastighet kan en bruke nærmere 300 m.s. pr. kort, mens nesten 600 m.s. er tilgjengelig for program på hvert kort når hastigheten er 100 kort/min.

I de kortene som avleses eller punches, blir opplysningene som allerede nevnt, representert ved 6-bits binærkoder i DL 12; den ene halvparten av DL 12 lagrer binærkodene fra innlesingen, mens den andre halvparten brukes til lagring av binærkodene til punching. Lese- og punchebuffer består hver alltid av 16 påfølgende minor-cycles, men det står en fritt å velge hvor DL 12 skal deles.

I motsetning til binærmetoden hvor ordrene "12-24" eller "10-24" setter i gang en kontinuerlig kortbevegelse i 528 og denne kortbevegelsen først kan stoppes ved ordren "9-24" eller CR, gjelder lese- og puncheordren i desimalmetoden for bare én kortcycle av gangen. Hvis en vil opprettholde en kontinuerlig kortbevegelse, må derfor "10-24,1" (som er leseordren) og/eller "12-24,1" (som er puncheordren) komme på hver kortcycle.

Oppdelingen av DL 12 i de to buffer-minnene blir definert på følgende måte: I hver kortcycle blir lesebuffer definert av den lese- eller puncheordre som blir utført først. Hvis denne ordren er lagret i m.c. (m) og har Wait = W, er m.c. (m+2+W) første ord i lesebuffer. De etterfølgende 15 minor-cycles, til og med m.c. (m+17+W), danner resten av lesebuffer. De resterende 16 minor-cycles, fra m.c. (m+18+W), utgjør punchebuffer. Hvis neste kortcycle skal være en såkalt "dual operation", dvs. at det forekommer både en lese- og en puncheordre, har Wait i den andre ordren ingen betydning for inndelingen av DL 12. Inndelingen av DL 12 må gjøres i hver kortcycle, slik at hvis det av en eller annen grunn er ønskelig, kan inndelingen av DL 12 forandres fra kort til kort.

I DL 12 pakkes binærkodene slik at i hvert ordpar plasseres 10 binærkoder á 6 bits. Et ordpar består, som kjent, av 64 bits; derfor brukes ikke de 4 høyeste bits (P61 - P64 er nuller). I fig. 2 på neste side ser en hvordan pakkingen er.

2.3.2. 6-bits binærkoder

De 12 radene i kortet har sin bestemte verdi ved desimalmetoden. På grunnlag av disse radverdiene blir så punchingen i de enkelte kolonnene omformet til binærkoder.

Y-raden	er	numerisk	lik	16	(000010)
X-raden	"	"	"	32	(000001)
0-raden	"	"	"	48	(000011)
1-raden	"	"	"	1	(100000)
2-raden	"	"	"	2	(010000)
3-raden	"	"	"	3	(110000)
4-raden	"	"	"	4	(001000)
5-raden	"	"	"	5	(101000)
6-raden	"	"	"	6	(011000)
7-raden	"	"	"	7	(111000)
8-raden	"	"	"	8	(000100)
9-raden	"	"	"	9	(100100)

Innen hver kolonne blir binærkoden bygget opp ved at verdiene av de radene hvor det forekommer punching i kolonnen, blir kombinert ved en logisk "og"-operasjon, dvs. at den endelige binærkoden har enere i de posisjonene hvor en eller flere av de forekommende radverdiene har enere.

Fig. 2: Pakking av binærkodene i DL 12.

m.c.:	1	67	1213	1819	2425	303132
n	kol. 80	kol. 79	kol. 78	kol. 77	kol. 76	kol
n + 1	75	kol. 74	kol. 73	kol. 72	kol. 71	
n + 2	kol. 70	kol. 69	kol. 68	kol. 67	kol. 66	kol
n + 3	65	kol. 64	kol. 63	kol. 62	kol. 61	
n + 4	kol. 60	kol. 59	kol. 58	kol. 57	kol. 56	kol
n + 5	55	kol. 54	kol. 53	kol. 52	kol. 51	
n + 6	kol. 50	kol. 49	kol. 48	kol. 47	kol. 46	kol
n + 7	45	kol. 44	kol. 43	kol. 42	kol. 41	
n + 8	kol. 40	kol. 39	kol. 38	kol. 37	kol. 36	kol
n + 9	35	kol. 34	kol. 33	kol. 32	kol. 31	
n + 10	kol. 30	kol. 29	kol. 28	kol. 27	kol. 26	kol
n + 11	25	kol. 24	kol. 23	kol. 22	kol. 21	
n + 12	kol. 20	kol. 19	kol. 18	kol. 17	kol. 16	kol
n + 13	15	kol. 14	kol. 13	kol. 12	kol. 11	
n + 14	kol. 10	kol. 9	kol. 8	kol. 7	kol. 6	kol
n + 15	5	kol. 4	kol. 3	kol. 2	kol. 1	
n + 16	kol. 80	kol. 79	kol. 78	kol. 77	kol. 76	kol
n + 17	75	kol. 74	kol. 73	kol. 72	kol. 71	
n + 18	kol. 70	kol. 69	kol. 68	kol. 67	kol. 66	kol
n + 19	65	kol. 64	kol. 63	kol. 62	kol. 61	
n + 20	kol. 60	kol. 59	kol. 58	kol. 57	kol. 56	kol
n + 21	55	kol. 54	kol. 53	kol. 52	kol. 51	
n + 22	kol. 50	kol. 49	kol. 48	kol. 47	kol. 46	kol
n + 23	45	kol. 44	kol. 43	kol. 42	kol. 41	
n + 24	kol. 40	kol. 39	kol. 38	kol. 37	kol. 36	kol
n + 25	35	kol. 34	kol. 33	kol. 32	kol. 31	
n + 26	kol. 30	kol. 29	kol. 28	kol. 27	kol. 26	kol
n + 27	25	kol. 24	kol. 23	kol. 22	kol. 21	
n + 28	kol. 20	kol. 19	kol. 18	kol. 17	kol. 16	kol
n + 29	15	kol. 14	kol. 13	kol. 12	kol. 11	
n + 30	kol. 10	kol. 9	kol. 8	kol. 7	kol. 6	kol
n + 31	5	kol. 4	kol. 3	kol. 2	kol. 1	
	1	45	1011	1617	2223	2829 32

M e r k s:

- 1) M.c. (n), dvs. første minorycle i lesebuffer, defineres av første lese- eller puncheordre innen hver kortcycle på følgende måte: Er denne ordren lagret i m.c. (m) og har en Wait = W, er $n = m + 2 + W$.
- 2) I dette tilfelle er pluggbordet i 528 standard plagget, dvs. "rett over" slik at kol. 80 går til pos. 1, kol. 79 til pos. 2 osv.

E k s.: I kol. 1 er det punchet i Y-, 3- og 9-raden.

$$\begin{array}{r} Y = 000010 \\ 3 = 110000 \\ \hline 9 = 100100 \\ Y + 3 + 9 = 110110 \end{array}$$

Så snart 9-raden er avlest og radverdiene fra denne er kombinert med verdien av de foregående rader, blir alle binærkodene automatisk undersøkt, og de binærkodene som har en verdi lik 0 (blank eller ikke avlest kolonne) eller 48 (vanligvis en punching i 0-raden alene) blir modifisert. Binærkoder lik 0 blir omgjort til 15, og binærkoder lik 48 blir omgjort til 0.

På grunn av den måten hvorpå binærkodene blir bygget opp, er det ikke mulig å skille mellom binærkoden for en kolonne med punching bare i 0-raden og binærkoden for en kolonne hvor det er punchet både i 0-raden og i X- og/eller Y-raden, eller en kolonne hvor det er punchet i både X- og Y-raden. I alle tilfelle blir binærkoden lik 48 (eller 0 etter modifiseringen). Dette betyr at en ikke kan benytte det vanlige prinsippet med å la en overpunching i et felt markere fortegnet hvis den kolonnen hvor overpunchingen forekommer også kan inneholde 0. På tilsvarende måte er det ikke mulig å skille mellom en blank kolonne og en kolonne hvor det er punchet både i 7- og i 8-raden. I begge tilfellene blir binærkoden lik 15 etter modifiseringen.

Det er 2^{12} (4 096) forskjellige mulige kombinasjoner av punchinger som kan forekomme i en kolonne. Binærkoden består bare av 6 bits, slik at antall binærkoder som kan forekomme er 63 (binærkoden 48 kan ikke forekomme). Det er derfor helt innlysende at en hel rekke av kombinasjonene i en kolonne må bli representert av samme binærkode, f.eks. representerer binærkoden 110000 følgende kombinasjoner: 3, 3 + 2, 3 + 1, 3 + 2 + 1 og 2 + 1.

Under punching kan det ikke forekomme alle mulige kombinasjoner av punchinger; hver enkelt binærkode vil resultere i en bestemt punching. I fig. 3 på side 34 og 35 er det satt opp en oversikt over binærkodene og de tilsvarende punchekombinasjonene.

2.3.3. Desimal innlesing

Leseordren ved desimal innlesing er "12-24,1". Det er altså bare den lange karakteristikken (C = 1) som skiller denne ordren ut fra den binære innlesingsordren.

Etter at DEUCE har mottatt en leseordre, foregår hele innlesingen automatisk, og binærkodene blir lagret i den delen av DL 12 som leseordren definerer. Den innbyrdes plassering av binærkodene fra de enkelte kolonnene er avhengig av pluggingen på pluggbordet.

Fig. 3: 6-bits binærkode og tilsvarende punchekombinasjoner.

6-bits binærkode		Punche- kombinasjoner	Tilsvarende tegn på tabulator
Des.	Bin.		
0	000000	0	0
1	100000	1	1
2	010000	2	2
3	110000	3	3
4	001000	4	4
5	101000	5	5
6	011000	6	6
7	111000	7	7
8	000100	8	8
9	100100	9	9
10	010100	2, 8	
11	110100	3, 8	
12	001100	4, 8	
13	101100	5, 8	
14	011100	6, 8	
15	111100	Blank	Blank
16	000010	Y	/
17	100010	Y, 1	A
18	010010	Y, 2	B
19	110010	Y, 3	C
20	001010	Y, 4	D
21	101010	Y, 5	E
22	011010	Y, 6	F
23	111010	Y, 7	G
24	000110	Y, 8	H
25	100110	Y, 9	J
26	010110	Y, 2, 8	
27	110110	Y, 3, 8	
28	001110	Y, 4, 8	
29	101110	Y, 5, 8	
30	011110	Y, 6, 8	
31	111110	Y, 7, 8	

Fig. 3 (forts.)

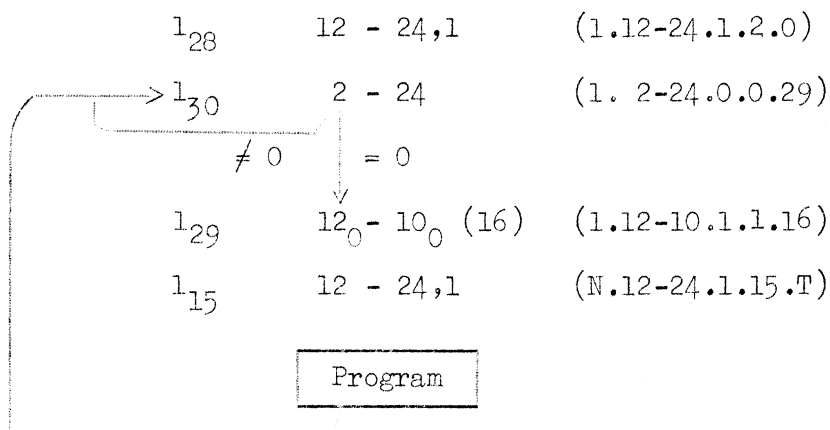
6-bits binærkode		Punche- kombinasjoner	Tilsvarende tegn på tabulator
Des.	Bin.		
32	000001	X	
33	100001	X, 1	K
34	010001	X, 2	L
35	110001	X, 3	M
36	001001	X, 4	N
37	101001	X, 5	P
38	011001	X, 6	Q
39	111001	X, 7	R
40	000101	X, 8	S
41	100101	X, 9	T
42	010101	X, 2, 8	
43	110101	X, 3, 8	
44	001101	X, 4, 8	
45	101101	X, 5, 8	
46	011101	X, 6, 8	
47	111101	X, 7, 8	
48	000011	O	O (numerisk)
49	100011	O, 1	O (alfabetisk)
50	010011	O, 2	U
51	110011	O, 3	V
52	001011	O, 4	W
53	101011	O, 5	X
54	011011	O, 6	Y
55	111011	O, 7	Å
56	000111	O, 8	Æ
57	100111	O, 9	Ø
58	010111	O, 2, 8	
59	110111	O, 3, 8	
60	001111	O, 4, 8	
61	101111	O, 5, 8	
62	011111	O, 6, 8	
63	111111	Blank	

Leseordren gjelder som før nevnt bare for en kortcycle av gangen. Hvis en derfor ønsker kontinuerlig kortbevegelse i lesebanen, må en passe på å gi ny leseordre så snart et kort er avlest.

Først når 9-raden i kortet har passert lesebørstene og er blitt registrert, er de fullstendige binærkodene fra kortet å finne i DL 12. Disse opplysningene kan da bearbeides mens et nytt kort blir innlest til DL 12. Før en setter i gang en ny innlesing, må en derfor overføre opplysningene fra lesebuffer til et midlertidig lagringssted, slik at de nye binærkodene kan lagres i DL 12. Når en har gitt en leseordre, blir lesebuffer automatisk nullstilt eller tømt like før avlesingen av den første raden (Y-raden) i kortet.

For at programmet kan være i stand til å avgjøre når et kort er ferdig avlest og binærkodene kan overføres fra DL 12, sender DEUCE ut et signal som starter når leseordren er gitt og varer inntil leseoperasjonen er ferdig. Dette signalet kalles `LONG TIL` og skiller seg ut fra det "binære" TIL-signalet, som bare sendes ut på slutten av kortet, ved å være under hele kortavlesingen og inntil 4 m.s. etter 9-radens Single-Shot. `LONG TIL` testes på samme måte som TIL ved ordren "2-24". Under desimal innlesing blir Single-Shots sendt ut på vanlig måte, men altså ikke det "binære" TIL-signalet.

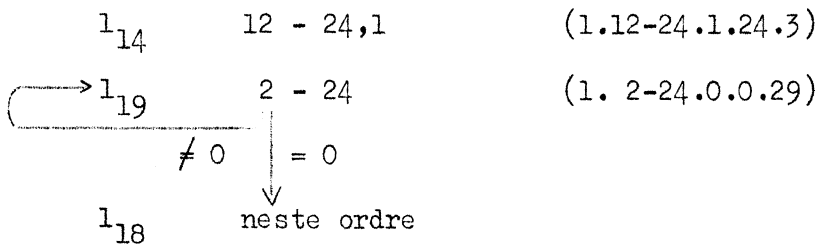
E k s.: Les kort til DL 12₀₋₁₅, overfør til DL 10₀₋₁₅



Den første ordren i eksempelet foran sørger for at det første kortet blir lest inn i DL 12. Den neste ordren undersøker om `LONG TIL` er borte, dvs. om innlesingen av kortet er ferdig. Så snart `LONG TIL` forsvinner, går DEUCE til den tredje ordren som overfører innholdet av lesebuffer til en mellom-lagringsplass i DL 10₀₋₁₅. Deretter gis en ny leseordre, og mens dette kortet leses inn, blir opplysningene fra foregående kort som er lagret i DL 10, bearbeidd i DEUCE. Når dette programmet er gjennomløpt, går en tilbake og undersøker `LONG TIL` igjen.

I begge eksemplene foran er DL 12₀₋₁₅ benyttet som lesebuffer, men det er som før nevnt ikke noe i veien for å bruke hvilke som helst 16 minor-cycles i DL 12 som lesebuffer bare disse ligger ved siden av hverandre.

E k s.: Innlesing av et kort til DL 12₈₋₂₃



2.3.4. Desimal punching

Ordren "10-24,1" gjelder som puncheordre etter den desimale metoden. Det er bare den lange karakteristikken som skiller den ut fra den binære puncheordren.

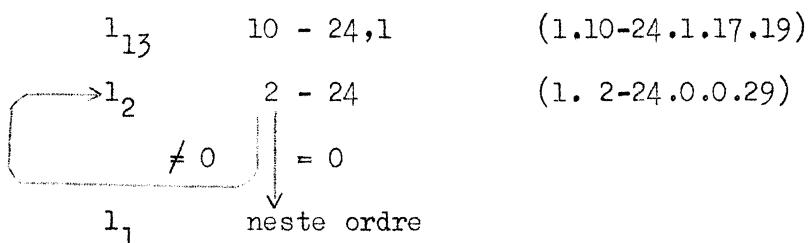
Når DEUCE mottar en puncheordre, foregår hele punchingen automatisk. De binærkodene som er lagret i punchebuffer i DL 12 blir punchet som desimaltall eller bokstaver i de kolonnene som de enkelte posisjonene i punchebuffer er plugget til på pluggbordet. "Wait" i puncheordren definerer også lesebuffer slik at det blir de øvrige 16 minor-cycles i DL 12 som danner punchebuffer.

Puncheordren gjelder akkurat som leseordren bare for en kort-cycle av gangen; derfor må en ny puncheordre gis så snart et kort er punchet hvis en ønsker å oppnå uavbrutt punching.

Når et kort skal punches, må opplysningene sendes til DL 12 i binærkodet form. Dette kan imidlertid ikke gjøres før det foregående kortet er ferdig punchet, ellers vil punchingen av dette kortet bli forstyrret. For at DEUCE kan være i stand til å avgjøre når et kort er punchet ferdig, benytter en seg av LONG TIL-signalet.

LONG TIL starter med en gang puncheordren er gitt og varer inntil punchingen er avsluttet, dvs. til 9-raden er punchet. Dette signalet undersøkes ved ordren "2-24". Under punchingen blir ikke det "binære" TIL-signalet sendt ut; derimot blir serien med Single-Shots sendt ut på vanlig måte.

E k s.: Punch et kort fra DL 12₁₆₋₃₁

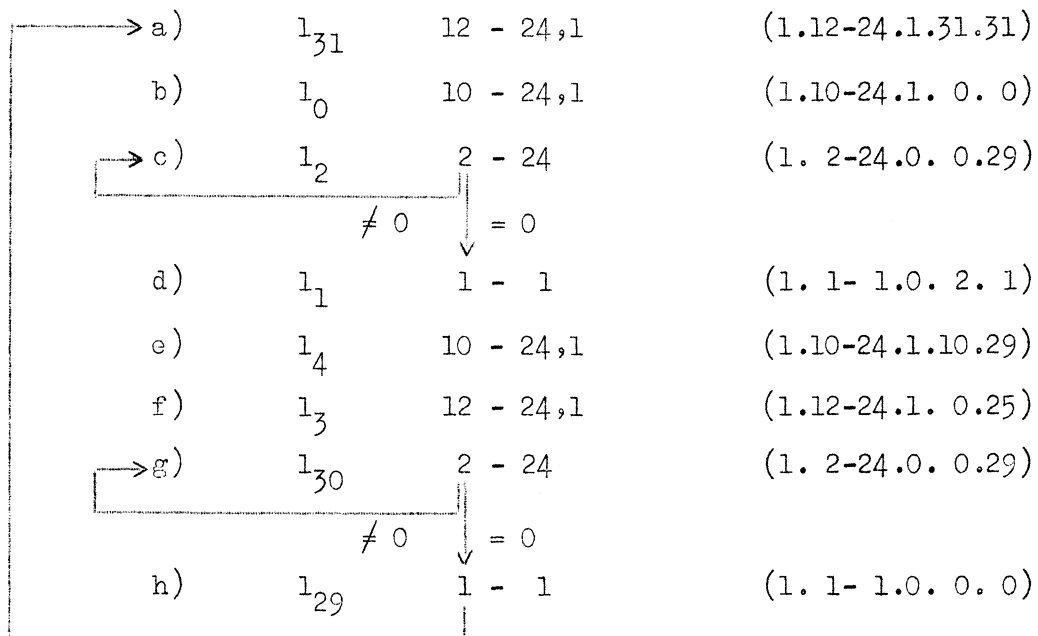


2.3.5. Samtidig punching og lesing ("Dual operation")

Lesing og punching kan foregå i samme kortcycle. Denne kortcyclen er da "langsom" (600 m.s.), og kortbevegelsene i de to banene er helt synkroniserte,

slik at det samme LONG TIL-signalet og den samme serien med Single-Shots gjelder for begge banene. Rekkefølgen av ordrene "12-24,1" og "10-24,1" er helt likegyldig, men tidsrommet mellom dem må ikke overskride 1 m.s. Dessuten gjelder den regelen at det alltid er den første av de to ordrene som definerer lesebuffer, uansett om dette er lese- eller puncheordren. Den andre ordrens "Wait" har derfor ingen betydning, men bør være lik 0 forat ikke tidsrommet mellom de to ordrene skal overskride 1 m.s.

E k s.: Bruk av 528 som reproduser



M e r k n a d e r:

- a) Denne ordren sørger for at et kort blir lest inn i DL l_{20-15}^{12} .
- b) Samtidig med lesingen blir et kort punchet fra den andre halvparten av DL 12. Tiden mellom utføringen av ordrene a) og b) er godt under 1 m.s.
- c), g) Undersøk om LONG TIL-signalet er til stede.
- d), h) Tiden mellom slutten av LONG TIL og første "10-24,1" eller "12-24,1" må være minst 2 m.s. En er derfor nødt til å legge inn en "dummy"-ordre.
- e), f) Et kort blir lest inn i DL l_{16-31}^{12} samtidig med at et kort blir punchet fra DL l_{0-15}^{12} . Tidsrommet mellom ordrene er mindre enn 1 m.s.

2.3.6. Tidsspesifikasjoner ved desimalmetoden

De tidsspesifikasjonene som er oppgitt i 2.2.3., gjelder også ved desimalmetoden; dessuten gjelder følgende regler som er spesielle for denne metoden:

- a) LONG TIL varer helt fra "10-24,1" eller "12-24,1" og inntil 4 m.s. etter 9-radens Single-Shot.

- b) Hvis en vil unngå å miste kortcycles, må ny lese- og/eller puncheordre gis innen en viss tid etter 9-radens S.S. Dette tidsrommet er avhengig av den nettopp avsluttede kortcycle. Hvis denne kortcycle har vært en rask kortcycle (300 m.s.), må ny lese- og/eller puncheordre gis innen 15 m.s. etter 9-radens S.S. Har derimot kortcyclen vært en punchecycle eller langsom lesecycle (600 m.s.), må ny "10-24,1" og/eller "12-24,1" komme innen 30 m.s. etter S.S. for 9-radens.
- c) Det må gå minst 2 m.s. mellom slutten av LONG TIL ("2-24") og første "10-24,1" eller "12-24,1".

2.4. Plugging på IBM 528

For at en skal kunne oppnå full fleksibilitet når det gjelder valg av kolonner under punching og lesing i 528, forekommer det på 528 et pluggbord.

Pluggbordet er delt inn i 4 soner:

- a) Binær innlesing, med CR (Clear Read).
- b) Binær punching, med "Job"- og kortnummer.
- c) Desimal innlesing.
- d) Desimal punching, med "Job"- og kortnummer.

Det må aldri forekomme plugging fra en sone til en annen. Videre bør en merke seg at når en puncher "Job"- og/eller kortnummer under desimal punching, kan tilsvarende mange posisjoner i DL 12 ikke punches.

2.4.1. Standard plugget pluggbord

Et standard plugget pluggbord vil være plugget slik at kol. 1 - 32 danner α -feltet og kol. 33 - 64 danner β -feltet under binær innlesing og punching. Videre er bordet plugget slik at kol. 65 er CR-kolonnen, og kortnummeret punches i kol. 77 - 80.

Standard plugging av desimal innlesing og punching resulterer i at kol. 1 går til pos. 80 i bufferminne, kol. 2 går til pos. 79, osv. slik som på fig. 2 side 32.

2.4.2. Kort- og "job"-nummer

På 528 finnes et telleverk på 4 posisjoner som automatisk teller antall kort som punches. Dette telleverket kan innstilles på et hvilket som helst tall (fra 0000 til 9999) ved hjelp av 4 brytere, én for hver telleverksposisjon. Hver bryter har 10 innstillinger, fra 0 til 9. Dessuten er det knapp hvormed en kan "nullstille" telleverket, dvs. at alle bryterne blir satt i 0-stilling.

Det er dessuten 8 andre brytere på 528; disse kalles "Job"-nummer bryterne og har også 10 innstillinger hver. "Job"-nummeret kan imidlertid bare forandres ved å forandre bryterinnstillingene, altså ingen automatisk endring.

Fra alle disse bryterne er det uttak på pluggbordet, 4 for kortnummer og 8 for "Job"-nummer. Under binær punching må disse uttakene plugges til kolonner som ikke er benyttet i noen av de to 32-kolonners feltene. Ved desimal punching må en ofre like mange posisjoner i punchebuffer som uttak en benytter, da en naturligvis ikke kan punche mer enn 80 kolonner i kortet.

2.5. Kontrollbrytere og alarmlamper

IBM 528 er utstyrt med en del kontrollbrytere og kontrolllamper.

2.5.1. Kontrollbrytere

- a) **I n i t i a l I n p u t**: Denne bryteren utfører følgende funksjoner: tømmes hurtigminnet, annullerer eventuell puncheordre, gir en binær leseordre og setter kortene i lesebanen i bevegelse.
- b) **R u n i n R e a d**: Når en trykker på denne knappen, blir kortene kjørt inn i lesebanen, slik at det første kortet blir liggende foran lesestasjonen og "Read Ready"-lampen vil lyse.
- c) **R u n o u t R e a d**: Denne bevirker at alle kortene i lesebanen blir kjørt gjennom slik at de blir liggende i kortlommen.
- d) **S t o p**: Denne bryteren stopper all bevegelse i 528 når inneværende kort-cycle er avsluttet.
- e) **R u n i n P u n c h**: Sørger for at kort blir kjørt inn i punchebanen slik at det første blir liggende like foran punchestasjonen og "Punch Ready"-lampen blir tent.
- f) **R u n o u t P u n c h**: Denne bryteren bevirker at de siste kortene som er punchet, men som ennå befinner seg i punchebanen, blir kjørt ut i kortlommen. Dette tar tre kortcycles.

2.5.2. Alarmlamper

- a) **R e a d R e a d y**: Det er bare når denne lampen lyser at kort kan leses inn ved ordrene "12-24" eller "12-24,1". Denne bryteren vil lyse når en bruker Run in Read-bryteren, slik at et kort blir liggende klar foran lesestasjonen.
- b) **P u n c h R e a d y**: Det er bare når denne lampen lyser at kort kan punches ved ordren "10-24" eller "10-24,1". En får denne lampen til å lyse ved å trykke på Run in Punch-knappen, slik at kort blir liggende klar foran punchestasjonen.
- c) **M i s s e d C a r d**: Denne lampen lyser når et kort har passert lese-

stasjonen uten å ha blitt lest inn. En får ikke slukket denne lampen uten å trykke på Stop-knappen.

- d) S t a c k e r: Denne lampen indikerer at en av kortlommene er full av kort. Lampen slukner når kortene fjernes fra lommen.

2.6. Kort-"crash"

Når en har "crasht" i 528 og ønsker å åpne dekslene for å få fjernet kort og kortrester, må en ta ut pluggbordet først, fordi i og med en gjør dette, blir den høyspente strømmen koblet ut fra valsene i lesebanen uten at en dermed kobler ut den høyspente strømmen fra resten av 528. Hvis en bare åpner side-dekslet, vil høyspenningen falle ut fra hele 528.

3. SKRIVING, PUNCHING, TESTING OG KJØRING AV PROGRAM

Et program bør lages slik at det foruten å være så raskt som mulig, også er lett å finne feil i (under testingen og seinere), lett å forandre og enkelt å bruke. Dette er sterke krav og ser kanskje ut til å trekke i forskjellige retninger, men disse kravene behøver ikke å komme i konflikt med hverandre. Det som programmeringsarbeidet blir forsinket på grunn av dette, blir mer enn oppveid av den tiden som spares ved å slippe strevet med retting og kjøring av dårlig planlagte program.

Dette kapitlet inneholder endel generelle retningslinjer og råd om hvordan hele programmeringsarbeidet bør legges opp. Etter hvert som en programmerer får erfaring, vil han finne fram til egne systemer eller versjoner som kan være like så gode eller bedre enn det som her er skissert. Hovedsaken må være at programmene lages og dokumenteres slik at de er lettfattelige for andre programmerere (og ikke minst for programmereren selv ett år seinere).

3.1. Skrivning av program

De fleste program er av en slik størrelse at det er umulig for den enkelte å ha fullstendig oversikt over programmet bare ved hjelp av hukommelsen. Det er derfor helt nødvendig at det blir tegnet opp diagrammer og skrevet detaljerte programbeskrivelser for alle program. Dette er særlig viktig når arbeidet med et program skal overlates til en annen; uten fullstendige diagrammer og beskrivelser vil den nye mannen ofte stå overfor store vanskeligheter og bruke uforholdsmessig lang tid til å finne ut av de kompliserte delene av programmet.

Den første fasen under planleggingen av et program er en inngående analyse av problemet. En slik analyse skal klargjøre alle punkter, f.eks. hva

er inngangsdata og hvorledes er disse gitt, hvilke resultater er det som ønskes og hvordan skal disse presenteres? Særlig i de tilfellene hvor programmereren har fått overlatt problemet fra en annen, er det av vesentlig betydning å få disse ting klarlagt på forhånd, fordi den andre som svært ofte ikke har kjennskap til elektroniske regnemaskiner og deres virkemåte, kan være svært uklar i sin formulering av problemet; ofte vil en finne at han ikke engang er helt klar over hva han ønsker.

3.1.1. Diagrammer

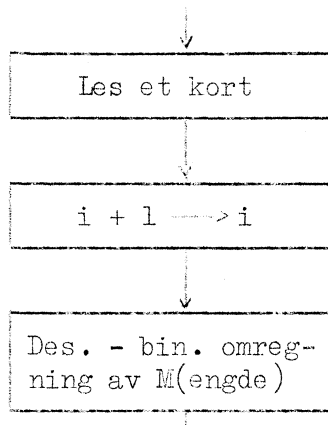
Et meget viktig hjelpemiddel for programmereren under planleggingen og utarbeidingen av program, er tegning av såkalte diagrammer. Et diagram er et forsøk på i en skjematisk framstilling å gi et bilde av gangen i problemløsningen, hva som punkt for punkt må gjøres med inngangsdata for at en skal komme fram til de størrelser som representerer resultatene.

Tegningen av et diagram er naturligvis ikke noen entydig operasjon. Et problem kan formuleres i et diagram på mange måter. Det kan være oversiktsdiagram som viser de store trekkene eller et detaljert diagram som prøver å gjengi detaljer i programmet. I alle tilfelle kan en si at et diagram er et visuelt hjelpemiddel for programmereren, dvs. hjelper ham å holde oversikten over programmet. Det er ingen gylden regel om hvor detaljerte diagrammene bør tegnes; det avhenger sterkt av den enkelte programmerers trening og erfaring.

For å lette lesingen av diagrammene bør en prøve å bruke visse standard symboler og betegnelser. Et diagram består i prinsippet av en samling bokser, linjer og notater. En skiller gjerne mellom tre typer av bokser: operasjonsbokser, valgbokser og fortsettelsesbokser. Størrelsen av boksene kan varieres etter behov. Boksene knyttes sammen i diagrammet ved hjelp av linjer; pilene på linjene angir rekkefølgen av boksene (operasjonene).

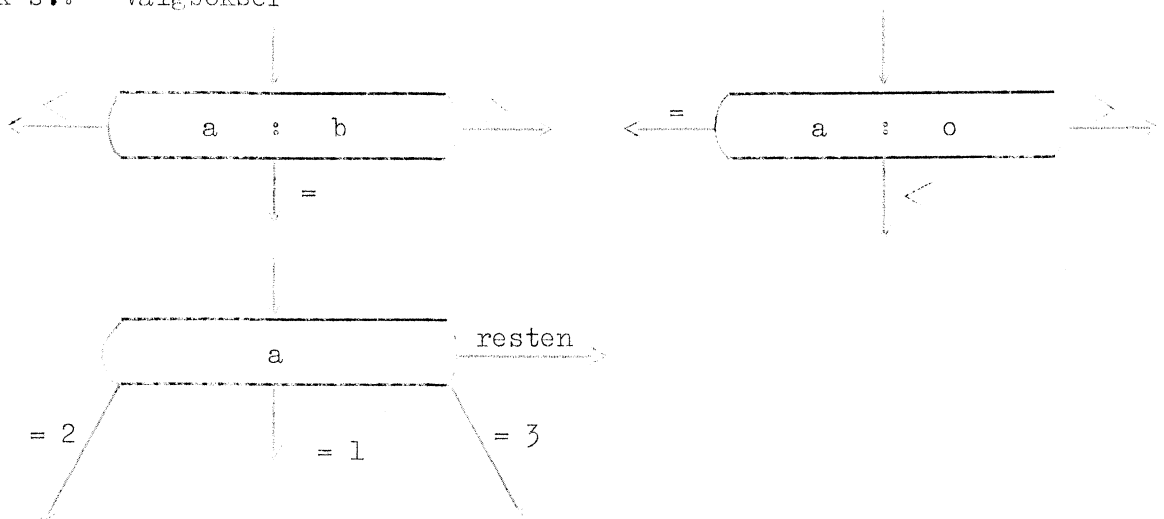
O p e r a s j o n s b o k s e r brukes over alt i diagrammet hvor en vil angi at det skal utføres en operasjon. Inne i boksen beskrives operasjonen ved hjelp av en kort tekst eller ligning. I de tilfellene hvor en bruker forkortelser eller andre betegnelser, bør diagrammet også inneholde en liste over disse og betydningen av dem. En operasjonsboks kan ha mange innganger, dvs. det kan føre flere linjer til samme operasjonsboks. Normalt har operasjonsboksene bare en utgang. (Som vi senere skal se gjelder ikke dette for hoveddiagrammer.)

E k s.: Operasjonsbokser



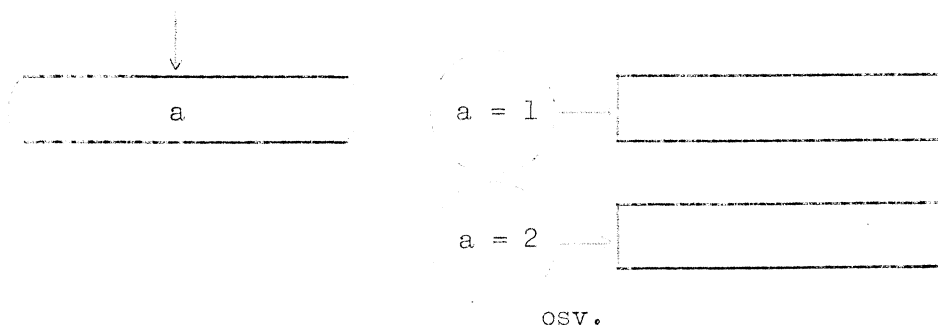
V a l g b o k s e r brukes i diagrammet når en på et punkt i programmet vil skille mellom to eller flere alternative veger; en valgboks har derfor alltid minst to utganger. Det som skal avgjøre vegvalget settes inne i boksen, og de betingelsene som skal oppfylles for de forskjellige vegene settes langs vedkommende veg. Hvis valget skal avgjøres på grunnlag av en sammenlikning mellom to størrelser, settes disse ett på hver side av sammenlikningstegnet ":".

E k s.: Valgbokser



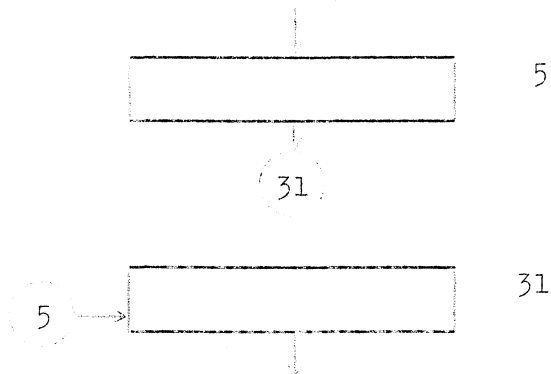
En valgboks kan ha så mange utganger at det ikke er praktisk mulig å tegne opp alle sammen. En kan da tegne opp valgboksen uten utgang og få fram sammenhengen i diagrammet ved å gjenta symbolet i valgboksen sammen med den verdien symbolet skal ha i en f o r t s e t t e l s e s b o k s foran den første boksen i hver veg.

E k s.: Fortsettelsesbokser



Fortsettelsesbokser brukes dessuten på de stedene i programmet hvor det ikke er mulig eller praktisk å tegne forbindelse mellom bokser ved hjelp av linjer og piler. I slike tilfelle brukes to fortsettelsesbokser, én etter den boksen hvor forbindelsen kommer fra og én foran den boksen forbindelsen skal til. For lettere å se forbindelsen mellom boksene bruker en boksnumrene. I alle diagrammene pleier en nemlig å nummerere boksene.

E k s.: Fortsettelsesbokser



I dette eksempelet er det en forbindelse mellom boksene 5 og 31 i diagrammet.

Foruten de typer av bokser som er nevnt foran, har en også en del andre standard symboler som brukes i diagrammene:

- : Sammenlikningstegn. Dette symbolet brukes i valgbokser mellom to størrelser som skal sammenliknes.
- =, >, < Når ett av disse tegnene brukes langs en av vegene fra en valgboks (dette er det normale), skal dette oppfattes slik at en ved å sette dette tegnet inn på sammenlikningstegnets plass i valgboksen, kan lese betingelsen som gjelder for denne vegen.
- Dette tegnet brukes i operasjonsbokser og betyr at uttrykket til venstre for tegnet skal erstatte det som står til høyre.
- < X > Adressen (lager-plassen) til X.
- (X) Innholdet av adresse X (lagerplassen X).

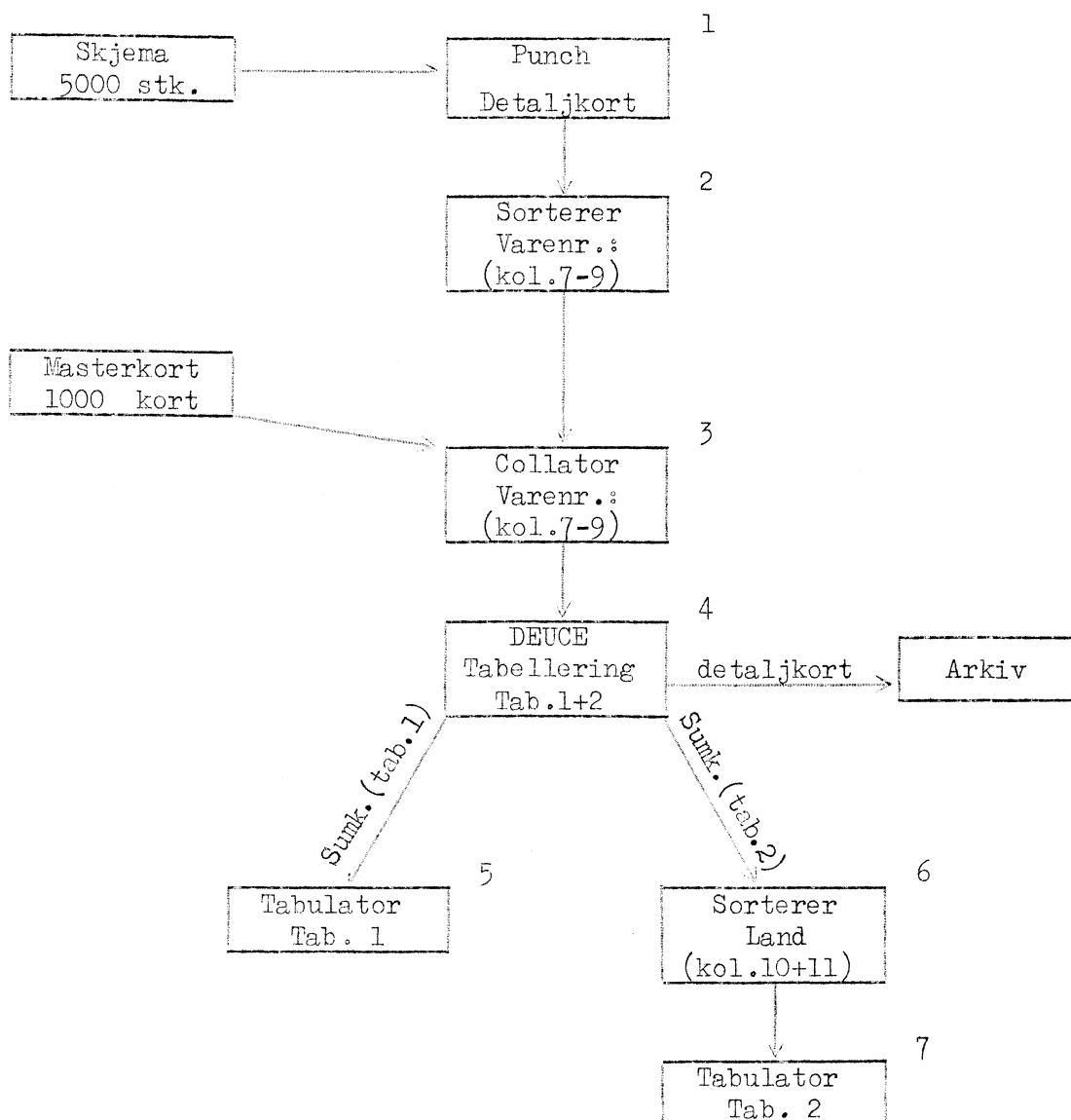
I alle diagrammer bør boksene nummereres, helst i den rekkefølge som operasjonene utføres. Teksten i diagrammene bør av hensyn til plassen være så kort som mulig, men bør samtidig gi tilstrekkelig opplysninger. Det hender ofte at en i diagrammene må bruke forkortelser og symboler som ikke er standard; i slike tilfelle bør en liste over disse føres opp på selve diagrammet eller som eget bilag.

H o v e d d i a g r a m m e t skal gi en oversikt over hele gangen i den maskinelle bearbeidingen av en oppgave fra punchingen av datakortene til tabelleringen av de siste resultatene. Dette diagrammet er meget grovt og

har til hovedhensikt å vise i hvilket omfang de enkelte maskinene blir benyttet under bearbeidingen.

Nedenfor er tegnet opp et eksempel på et meget enkelt hoveddiagram. I dette diagrammet er det særlig boks 4 som er av interesse for programmereren. Han må nemlig lage et program som utfører tabelleringen av detaljkortene til tab. 1 og tab. 2. Det må da først lages et diagram som gir oversikt over hva som skal gjøres i DEUCE for å komme fram til tabellene 1 og 2. Dette diagrammet kalles gjerne et funksjonsdiagram.

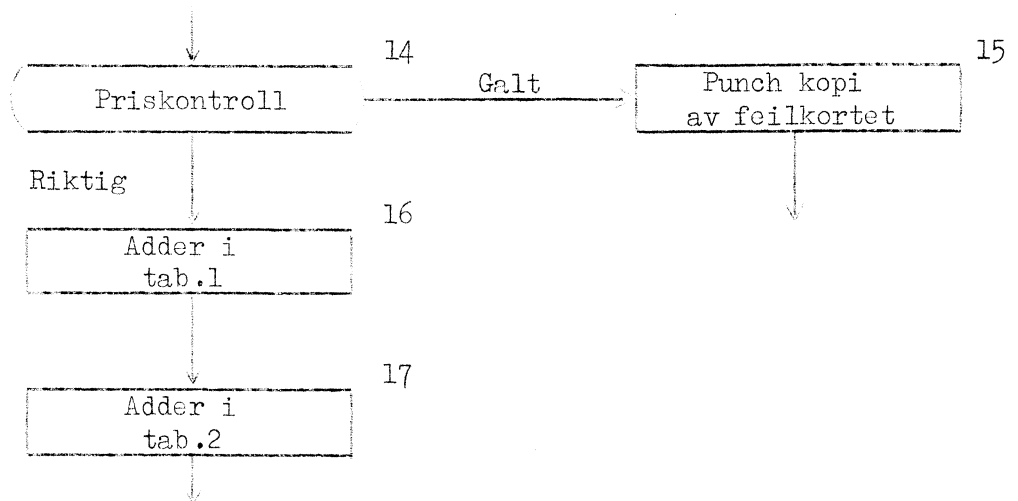
E k s.: Hoveddiagram



F u n k s j o n s d i a g r a m m e t bør tegnes slik at de spesielle eiendommelighetene ved DEUCE ikke kommer inn i bildet. Teoretisk skulle et funksjonsdiagram kunne lages uten spesiell kjennskap til DEUCE, men i praksis viser det seg at allerede under utarbeidingen av hoved- og funksjonsdiagrammet må en ta hensyn til denne maskinens egenskaper. En må imidlertid forlange at

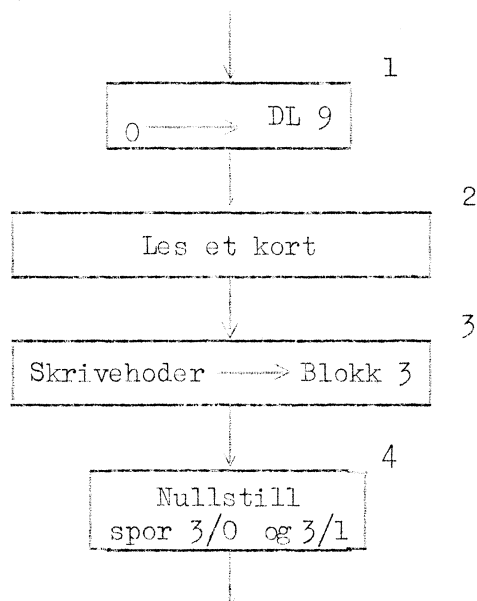
selv om det er tatt hensyn til dette under tegningen av funksjonsdiagrammet, bør det kunne leses og forstås av andre som ikke har kjennskap til DEUCE. For å unngå misforståelser bør funksjonsdiagrammet, sammen med en kortfattet p r o g r a m b e s k r i v e l s e , overleveres oppdragsgiveren til gjennomlesing og godkjenning. Programmereren bør være særlig interessert i dette fordi eventuelle feil og misforståelser da kan oppdages på et tidlig tidspunkt.

E k s.: Funksjonsdiagram



For hver DEUCE-kjøring som er tegnet inn på hoveddiagrammet, utarbeides videre et l o g i s k d i a g r a m. Det er ikke noen regel om hvor detaljert et logisk diagram skal tegnes, men i dette diagrammet bør ting som f.eks. magnettrommel-overføringer, desimal til binær omregning, binær til desimal omregning, programhenting osv., komme med. For ikke å ødelegge oversikten og plassen i diagrammet bør en imidlertid ikke lage det logiske diagrammet for detaljert; detaljer og spissfindigheter som en gjerne ønsker med, kan heller tegnes inn på u n d e r d i a g r a m m e r.

E k s.: Logisk diagram



Det neste trinn i programmeringsarbeidet er `flow-diagrammet`, som følger det logiske diagrammet meget nøye. Flow-diagrammet består av ordrer og subrutiner. Ofte vil programmereren oppdage under utarbeidingen av flow-diagrammet at det logiske diagrammet må revideres på visse punkter fordi den eller de metodene som er skissert i diagrammet, ikke fører fram eller fordi det er andre metoder som passer bedre for DEUCE. Det endelige logiske diagrammet kan derfor ofte ikke tegnes før flow-diagrammet er ferdig. De enkelte programmererne har sine egne metoder under utarbeidingen av diagrammene. Noen legger ned et svært arbeid på det logiske diagrammet og tegner det ene utkastet etter det andre før de endelig kommer fram til et som de er så tilfreds med at de kan fortsette med flow-diagrammet. Andre nøyer seg med å lage et grovt skissert logisk diagram og starter deretter umiddelbart på utarbeidingen av flow-diagrammet. Disse programmererne bygger da opp det endelige logiske diagrammet samtidig med flow-diagrammet. Denne siste metoden kan en ikke anbefale for nye og uøvde programmerere; erfaringsmessig har det nemlig vist seg at, uansett hvor store eller små programmene er, lønner det seg å legge ned mye arbeid i det logiske diagrammet og prøve å lage dette så fullstendig som mulig før en begynner med flow-diagrammet.

I flow-diagrammene bør helst ordrene nummereres slik at en har referanser til de enkelte ordrene; dette er særlig viktig under plasseringen i minnet. Ordrenummereringen bør dessuten bygges opp slik at det er lett å se forbindelsene mellom boksene i det logiske diagrammet og ordrene i flow-diagrammet. En enkel måte å ordne denne nummereringen på er å bruke boksnumrene i det logiske diagrammet sammen med et undernummer slik at f.eks. de ordrene som tilsvarende boks 3 får numrene 3.1, 3.2, 3.3 osv.

Når en har skrevet alle ordrene i flow-diagrammet og har nummerert disse, er neste trinn i arbeidet å plassere ordrene i minnet. Hvis en ikke da benytter et system for registrering av plasseringene, kan en komme til å plassere to ordrer i samme minne-posisjon. Slike feil kan være svært ubehagelig å rette opp hvis det er store program og hurtigminnet er godt utnyttet. Skjemaet på neste side har vist seg å være meget brukbart. På dette skjemaet har hvert enkelt ord i hurtigminnet sin egen rute. Når et ord i minnet blir opptatt av en ordre, konstant o.l., merker en bare av i tilsvarende rute. Noen nøyer seg med å krysse av i ruten, mens andre skriver ordrenummeret til den ordren som er plassert i tilsvarende ord. Den siste metoden er å foretrekke, i hvert fall når det gjelder større program, fordi en ofte vil komme ut for det tilfellet at en ønsker å lagre en ordre i et ord som allerede er opptatt, dvs. den ordren som allerede er plassert der bør helst flyttes til et annet sted. Hvis en har skrevet ordrenummeret i ruten, er det lett å finne den ordren som bør flyttes.

Dette er ikke så enkelt hvis en bare krysser av i ruten, en må da lete seg fram på flow-diagrammet til en finner den ordren som har den adressen.

På nederste del av skjemaet er det ruter for avmerking av spor-disponeringen på trommelen, én rute for hvert spor.

3.1.2. Koding

Når diagrammene er tegnet og ordrer, kataloger, konstanter etc. er plassert i minnet, begynner kodingen av programmet. En koder ordrene og konstantene etter hvert som de opptrer i flow-diagrammet. Under kodingen benytter en seg av spesielle kodeark. (Se neste side.) På hvert enkelt kodeark er det tre hovedkolonner; i hver hovedkolonne er det plass for alle ordrene innen en delay-linje, dvs. at det er plass til tre DL's på hvert ark.

Kodingen av programmet bør som regel utføres av programmereren selv; visse mindre program kan naturligvis overlates til mindre erfarne programmerere for koding. Ofte vil imidlertid logikken i programmet være slik at det kan være svært vanskelig for andre enn programmereren selv å kode riktig etter flow-diagrammet; særlig gjelder dette løkker, modifikasjoner, initialordre o.l. Det finnes flere lettkodingssystemer hvormed en kan spare kodingen; for de mest eleverte systemer kan en også forenkle flow-diagrammet ganske mye. Disse systemene blir behandlet i senere kapitler i dette hefte.

Da programmereren sjelden puncher sine egne programmer, må kodearket føres nøyaktig, og en bør følge visse standard regler under kodingen:

- a) Ordredelene NIS, S, D, W og T bør alltid skrives. En kan se eksempler på at noen programmerere unnlater å skrive W og T hvis disse er lik null. Etter forfatterens mening er dette uheldig fordi en da ikke vil kunne skille mellom ordrer med W og T lik null og de ordrene hvor en av en eller annen grunn har glemt å påføre W og T.
- b) Meningene er noe mer delte når det gjelder skrivingen av Ch og Go i ordrene. Mange skriver alltid disse ordredelene også, dvs. Ch som 0, 1, 2 eller 3 og Go som 0 eller 1, fordi de hevder at dette letter puncharbeidet. Andre programmerere, deriblant forfatteren, mener derimot at en får bedre oversikt hvis Ch bare skrives når den er forskjellig fra null og at en bør sette et kryss (x) i Go-kolonnen i de ordrene hvor $Go = 0$ (stoppordre), mens en ikke avmerker noe i Go-kolonnen når $Go = 1$.
- c) I initialordrer, "17-0"- eller "18-0"-ordrer og konstanter som skrives som ordrer, kan det ofte hende av P_1 , Joe og/eller P_{31} skal skrives. Disse skrives da i parentes henholdsvis foran NIS, mellom W og T og mellom T og Go.

- d) De fleste konstanter skrives ikke som ordrer, men som en serie med P-er. En konstant som skal inneholde enere i de 4 laveste posisjonene og resten nuller, kan skrives enten som $(15 \cdot P_1)$ eller $(P_1 - P_4)$ eller (P_{1-4}) . En konstant som består av enere i posisjonene 18, 20 og 21 kan på tilsvarende måte skrives som $(13 \cdot P_{18})$ eller $(P_{18} + P_{20-21})$. Ofte vil konstantene være noe mer kompliserte, f.eks. $(P_{1-3} + P_8 + P_{11-15} + P_{18} + P_{23-24})$ eller $(7 \cdot P_1 + P_8 + 31 \cdot P_{11} + P_{18} + 3 \cdot P_{23})$. Hvis en har en katalog eller andre konstanter som fyller en hel delay-linje, kan det ofte lønne seg å dele ordene opp i 6-bits grupper. En delay-linje med konstanter kan da skrives slik:

$$\begin{array}{cccccccc}
 a_1 \cdot P_1 + b_1 \cdot P_7 + c_1 \cdot P_{13} + d_1 \cdot P_{19} + e_1 \cdot P_{25} + f_1 \cdot P_{31} \\
 a_2 \cdot P_1 + b_2 \cdot P_7 + c_2 \cdot P_{13} + d_2 \cdot P_{19} + e_2 \cdot P_{25} + f_2 \cdot P_{31} \\
 a_3 \cdot P_1 + b_3 \cdot P_7 + c_3 \cdot P_{13} + d_3 \cdot P_{19} + e_3 \cdot P_{25} + f_3 \cdot P_{31} \\
 - \quad - \quad - \quad - \quad - \quad - \quad - \quad - \quad - \quad - \quad - \quad - \\
 - \quad - \quad - \quad - \quad - \quad - \quad - \quad - \quad - \quad - \quad - \quad - \\
 a_{32} \cdot P_1 + b_{32} \cdot P_7 + c_{32} \cdot P_{13} + d_{32} \cdot P_{19} + e_{32} \cdot P_{25} + f_{32} \cdot P_{31}
 \end{array}$$

Vanligvis pleier en å forenkle skrivemåten slik:

a_1	b_1	c_1	d_1	e_1	f_1
a_2	b_2	c_2	d_2	e_2	f_2
a_3	b_3	c_3	d_3	e_3	f_3
-	-	-	-	-	-
-	-	-	-	-	-
a_{32}	b_{32}	c_{32}	d_{32}	e_{32}	f_{32}

Under denne delay-linjen på kodearket må en da tydelig avmerke at konstantene er inndelt i 6-bits grupper. Denne metoden er særlig aktuell for de katalogene som inneholder 6-bits binærkodede desimaltall.

De 4 første linjene i hver hovedkolonne på kodearket gir plass for såkalte "filling instructions". Disse ordrene er laget slik at de øvrige 32 ordrene i hovedkolonnen kan leses direkte inn i én delay-linje uten bruk av spesielle innlesingsprogrammer. "Filling instructions" er noe forskjellig avhengig av hvor de etterfølgende 32 ordrene skal lagres, nemlig:

- a) i delay-linje 1-8,
- b) i " 9-12,
- c) på magnet-trommelen.

Disse ordrene er dessuten avhengig av hvorvidt en har benyttet bare α -feltet eller både i α - og β -feltet i programkortene. Under punchingen av programmene bør en bare bruke α -feltet, selv om en da får en noe større masse med programkort, fordi en dessverre alltid må regne med at det forekommer feil i programmet, og det vil da være enklere å rette opp eventuelle feilkort. Senere når programmet er ferdig testet og funnet i orden, kan en eventuelt få overført programmet til 64-kolonners.

Hvis en kaller den delay-linjen som skal fylles for "N", ser "filling instructions" for delay-linjene 1-8 slik ut:

1. linje	blank
2. "	N. 0 - N. 1. 26. 25. x
3. "	N. 0 - N. 0. 30. 31. x
4. "	1. 0 - N. 0. 30. 31. x

Det er imidlertid et unntak fra denne regelen. For den delay-linjen som leses inn sist (vanligvis DL 1), må den fjerde ordren forandres til å se slik ut:

$$M. 0 - N. 0. 30. (m - 1). x$$

hvor DL M_m er den første ordren i programmet.

"Filling instructions" for de øvrige delay-linjene, DL 9 - DL 12, ser slik ut:

1. linje	1. 0 - 7. 1. 29. 28. x
2. "	7. 0 - N. 0. 30. 31. x
3. "	7. 0 - 7. 0. 27. 28. x
4. "	1. 0 - N. 0. 30. 31. x

Ofte skal en del program, kataloger o.l. leses inn på trommelen, men en kan ikke få lest inn på trommelen ved hjelp av bare 4 ordrer. En er nødt til å bruke et av de programmene som er laget for dette bruk. Det mest hensiktsmessige programmet ser foreløpig ut til å være det programmet som går under betegnelsen "ZPl4 - Read to Drum". Når en benytter dette programmet behøver bare den første av de fire linjene fylles ut; på denne linjen skriver en: " $P_{15} + t \cdot P_{17}$ ", hvor "t" er spornummer. (Sporene nummereres fra 0 til 255.) Den første linjen i den siste kolonnen med ordre som skal til trommelen, må dessuten inneholde en P_{31} , slik at den blir: " $P_{15} + t \cdot P_{17} + P_{31}$ ".

3.2. Punching av program

Når en har ført opp alle ordrene, konstantene osv. på kodearket, er programmet klart for punching. Som regel overlater programmereren punchingen av programkortene til en øvet puncheoperator.

Ordrene punches binært i programkortene, én ordre på hver rad. Det er intet i vegen for at en kan punche to ordrer i hver rad, dvs. bruke både

α - og β -feltet, men av hensyn til testingen og særlig rettingen av eventuelle feil, lønner det seg å bruke bare α -feltet. I hver hovedkolonne på kodearket er det plass til inntil 36 ordrer (derav 4 "filling instructions"); disse 36 ordrene punches på tre kort:

Rad	1. kort	2. kort	3. kort
Y $\left. \begin{array}{l} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{array} \right\}$		D ₈	D ₂₀
x $\left. \begin{array}{l} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{array} \right\}$	"filling	D ₉	D ₂₁
0 $\left. \begin{array}{l} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{array} \right\}$	instructions"	D ₁₀	D ₂₂
1 $\left. \begin{array}{l} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{array} \right\}$		D ₁₁	D ₂₃
2	D ₀	D ₁₂	D ₂₄
3	D ₁	D ₁₃	D ₂₅
-	-	-	-
-	-	-	-
9	D ₇	D ₁₉	D ₃₁

Disse tre kortene danner en såkalt triade.

I programkortene opptar α -feltet kol. 1-32, β -feltet kol. 33-64. I kol. 65 punches CR (hvis den skal forekomme); under binær innlesing vil en CR (Clear Read) som kjent stoppe innlesingen. Programnummeret punches i kol. 66-72. Før programmet er testet har en funnet det hensiktsmessig å punche blokk- og spornummer i kol. 73-76. Dette gjelder naturligvis bare de triadene som skal til trommelen. I kol. 77 punches den delay-linjen som triaden skal leses til, enten direkte under programinnlesingen eller fra trommelen i programmet. Kortnummer punches i kol. 78-80.

Før programkortene kan kjøres inn, må en sørge for at DEUCE står i riktig utgangsstilling. Dette oppnås delvis ved hjelp av bryterne på kontrollbordet og delvis ved standard hjelpeprogrammer. De viktigste av disse hjelpeprogrammene er:

- a) Initialkort - 1 kort
- b) Set and Sync Clock Track (ZP 34) - 3 "
- c) Read to Drum (ZP 14) - 3 "

Når programmet skal leses inn, bør det leses inn i denne rekkefølge:

- a) Initialkort
- b) Set and Sync Clock Track
- c) Read to Drum
- d) Triader som skal til trommelen (ordnet etter stigende rekkefølge av sporene)
- e) Triader som skal til delay-linjene (ordnet etter synkende rekkefølge av delay-linjene).

I de tilfellene hvor en bare benytter hurtigminnet, går punktene c) og d) ut.

I det følgende skal en se litt nærmere på de viktigste hjelpeprogrammene under programinnlesing.

3.2.1. Initialkort

Minor-cycles i DEUCE har ikke noe absolutt nummer, men nummereringen er relativ i forhold til et visst punkt. Dette varierer fra gang til gang. DEUCE skiller imidlertid alltid mellom like og ulike minor-cycles. Når en starter opp en kjøring, dvs. før programmet leses inn, må en derfor identifisere like og ulike minor-cycles, slik at de ordrene som skal inn i like minor-cycles, kommer inn i like og omvendt. Dette klarer en ved hjelp av initialkortet.

Det forekommer mange versjoner av initialkortet fra de svært enkle hvor en nøyer seg med å identifisere like og ulike minor-cycles, til de ganske innviklede hvor en har kombinert dette med en del andre initialoperasjoner som kan være nødvendige.

Et av de absolutt enkleste initialkortene ser slik ut:

Y-raden	blank
X- "	blank
0- "	1. 4 - 24. 0. 0. 0. x
1- "	blank
2- "	1. 29 - 22. 0. 1. 28. x
3- "	blank
4- "	1. 21 - 28. 0. 0. 28. x
5- "	blank
6- "	1. 30 - 21. 1. 0. 28. x
7- "	blank
8- "	1. 30 - 16. 0. 0. 28. x
9- "	blank

Initialkortet kjøres inn i DEUCE ved hjelp av Initial Input-knappen på 528. Når en trykker på denne knappen har det som nevnt den virkningen at hele hurtigminnet blir tømt og kortene i lesebanen blir satt i gang. TS COUNT tømmes samtidig med hurtigminnet og inneholder derfor ved starten: "0. 0 - 0. 0. W. T x"; W og T kan ha hvilken som helst verdi mellom 0 og 31 fordi W og T ikke blir nullstilt. Som en husker vil en overføring til TS COUNT normalt bare bli utført hvis D 0-ordren er lang; ellers vil overføringen fra den oppgitte Source være slutt før TS COUNT er mottakelig for neste ordre slik at neste ordre kommer fra den DL som NIS i ordren angir. I de tilfellene W = T, blir overføringen til Destination 0 effektiv også for en kort ordre (C = 0). Den ordren

som opprinnelig står i TS COUNT blir i alle tilfelle utført ved Y-radens Single-Shot. Hvis W tilfeldigvis er lik T, blir Y-radens overført til TS COUNT som neste ordre; i de andre tilfellene ($W \neq T$) blir en minor-cycle fra DL 8 (NIS 0) overført til TS COUNT. I dette tilfelle spiller det ingen rolle hva som skjer, fordi Y-radens er blank og hele DL 8 er null: "0. 0 - 0. 0. 0. 0. x" går til TS COUNT som neste ordre. Denne ordren blir utført på X-radens Single-Shot og bevirker at det som står i Q-feltet i X-radens går til TS COUNT (0. 0 - 0. 0. 0. 0. x). Denne ordren sørger for at ordren "1. 4 - 24. 0. 0. 0. x" blir hentet inn som neste ordre osv. En anbefaler leseren å fortsette, og han vil da finne ut at følgende blir utført av initialkortet:

- a) identifisering av like og ulike minor-cycles,
- b) clear TCA,
- c) clear TCB.

Det mest generelle initialkortet er det programmet som benevnes ZP48 M, og ser slik ut:

Y-radens	P ₁₅₋₃₁	(identifikasjon)
X- "	1. 0 - 1. 1. 10. 22. x	
0- "	1. 0 - 1. 2. 0. 28. x	
1- "	1. 21 - 28. 0. 1. 31	
2- "	1. 27 - 23. 0. 0. 0	
3- "	1. 4 - 24. 0. 0. 28	
4- "	2. 17 - 0. 1. 0. 0	
5- "	1. 31 - 30. 1. 0 (15) 22	
6- "	1. 12 - 24. 0. 0. 6	
7- "	1. 1 - 17. 0. 2. 4	
8- "	1. 0 - 1. 0. 5. 0. x	
9- "	1. 31 - 29. 1. 0. 28	(punching i kol. 65)

En anbefaler leseren å gå nøye gjennom dette kortet og prøve å lage flow-diagram, både i de tilfellene starten er på en like minor-cycle og når starten er på en ulike. En vil da finne at dette initialkortet utfører følgende operasjoner:

- a) identifiserer like og ulike minor-cycles,
- b) nullstiller (tømmer) trommolen,
- c) setter TCB "off",
- d) kontrollerer at den automatiske ordremodifiseringen er i virksomhet.

Dette initialkortet sørger derimot ikke for å slå "av" TCA, flytte lese- og skrivehodene til blokk 0 og nullstille Q S 17, DS 21 og DL 1 etter at programmet har brukt disse plassene.

3.2.2. Clock Track

Når både hurtigminnet og magnettdrømmelen brukes i et program, må naturligvis disse være synkroniserte slik at det vi kaller mc. 0 i delay-linjene tilsvarer den minor-cycle som vi kaller 0 på trømmelen. Blir programmet til hurtigminnet og trømmelen lest inn samtidig, vil denne synkroniseringen alltid være til stede, forutsatt at det ikke er maskinfeil. Det kan imidlertid forekomme tilfelle da en ønsker å starte opp et program uten å være nødt til å lese inn på trømmelen program og data som allerede står der. I slike tilfelle benytter en seg av en såkalt "Clock Track" til å få synkronisert minor-cycles i hurtigminnet og magnettdrømmelen.

"Clock Track" kan defineres som et spor på trømmelen som inneholder en bestemt serie av bits. Dette mønsteret gjør en i stand til å finne tilbake til nummereringen av minor-cycles når denne av en eller annen grunn er gått tapt. Det er to hovedtyper av "Clock Tracks":

- a) et helt spor med ordrer. Disse ordrene er slik at uansett hvilken ordre som først blir utført, blir utgangen alltid den samme.
- b) " P_{31} - Clock Track" hvor det eneste som forlanges av sporet, er at det inneholder P_{31} i et bestemt ord (nemlig minor-cycle 16). For øvrig kan dette sporet inneholde ordrer eller data bare det ikke forekommer flere P_{31} .

Av disse to prinsippene er det siste å foretrekke. Det forekommer et program som kalles "ZP 34 - Clock Track, Set and Sync", hvormed en kan sette "Clock Track" eller hvis "Clock Track" allerede forekommer, sørge for at hurtigminnet blir synkronisert med denne. Det er spor 15/15 som blir brukt som "Clock Track".

Programmet er på 3 kort og operasjonene utføres etter følgende system:

Hvis 15/15 er en " P_{31} - Clock Track", dvs. at det forekommer P_{31} i mc. 16, blir programmet i hurtigminnet lagret i takt med denne. Hvis det ikke forekommer P_{31} i 15/15, adderes P_{31} til mc. 16 i 15/15. Er det flere enn en P_{31} i 15/15, nullstilles hele sporet og P_{31} settes inn i mc. 16.

3.2.3. Read to Drum

Ved programinnlesing til trommelen strekker ikke de 4 første ordrene i triaden til som "filling instructions". En må i stedet benytte et spesielt program som sendes foran de triadene som skal fylle spor på trommelen. Det finnes flere slike program; det beste ser foreløpig ut til å være 2-korts programmet "ZP 14 - Read to Drum".

Selve "read to Drum"-programmet består av følgende to kort:

	Kort 1:	Kort 2:
Y-raden	1. 0 - 1. 1. 8. 27. x	1. 14 - 25. 0. 0. 4
X- "	1. 0 - 1. 0. 30. 29. x	1. 24 - 14. 2. 0. 4. x
0- "	1. 0 - 14. 0. 0. 11. x	1. 0 - 30. 1. 0. 19
1- "	1. 25 - 25. 1. 12. 27	1. 24 - 13. 0. 0. 2
2- "	1. 23 - 14. 1. 7. 24	1. 25 - 26. 0. 0. 2
3- "	1. 13 - 14. 0. 0. 25	1. 0 - 31. 1. 0. 16
4- "	0. 13 - 0. 0. 0. 0	1. 1 - 13. 0. 2. 13
5- "	1. 14 - 28. 0. 0. 0. x	1. 1 - 13. 0. 0. 1
6- "	1. 0 - 11. 0. 1 (15) 14. x	1. 28 - 25. 0. 0. 12
7- "	1. 16 - 27. 0. 0. 5	1. 24 - 16. 0. 0. 6. x
8- "	1. 1 - 13. 0. 0. 2	1. 0 - 15. 0. 0. 16. x
9- "	1. 27 - 25. 0. 0. 0	P ₁₋₄

Leseren anbefales også her å utarbeide et flow-diagram på grunnlag av ordrene i de to kortene. En vil da finne ut at programmet er ordnet slik at på Y-raden i det første kortet innen hver triade som skal til trommelen, punches $P_{15} + t \cdot P_{17}$ hvor t i spornummeret (fra 0 til 255). De tre neste radene i det første kortet innen triaden, X-, 0- og 1-raden, kan inneholde hva som helst, men en bør la dem være blanke. I den siste triaden som skal til trommelen må den første raden punches litt annerledes, nemlig: " $P_{15} + t \cdot P_{17} + P_{31}$ ". Hvis det forekommer triader hvor den første raden er blank, blir denne triaden lest inn på det sporet som følger etter det spor som foregående triade ble lest inn på. En anbefaler imidlertid alltid å punche den første raden i hver triade selv om triadene skal lagres i etterfølgende spor.

3.3. Program-testing

Erfaringsmessig viser det seg at det er svært vanskelig å gjøre selv de enkleste programmer uten feil, fordi det er så mange detaljer å ta hensyn til. Feilene i programmene kan deles inn i tre hovedgrupper:

- a) enkle feil, det kan være rene skrivefeil, regnefeil eller punchefeil,
- b) logiske feil i diagrammene,
- c) feil i problemdefinisjonen.

Testingen av program kan være en meget langsom og kostbar operasjon; derfor bør programmereren alltid ha plan for testingen. En skal i det følgende gi litt veiledning i hvordan programtestingen kan legges opp.

3.3.1. Kontroll før testing på DEUCE

Mange feil i programmet kan finnes før testingen på DEUCE, hvis en gjennomfører en nøyaktig gransking av programmet. En slik kontroll vil vanligvis kreve lang tid, men dette vil likevel lønne seg framfor å bruke verdifull DEUCE-tid for å finne feilene.

En slik gjennomgåelse av programmet kan selvsagt gjøres av programmereren selv, men bør helst overlates til en annen programmerer fordi det har vist seg at en programmerer som går gjennom sitt eget program, har tendens til å tenke i de samme banene som når programmet opprinnelig ble laget. En annen fordel med dette systemet er at en blir tvunget til å dokumentere programmet ordentlig fordi den som skal kontrollere programmet, vil trenge programbeskrivelse, oversikt over input-data, tabellspesifikasjoner osv. sammen med en muntlig orientering.

Det første trinnet i kontrollarbeidet er å gå nøye gjennom flow-diagrammet og se etter at programmet utfører det som det skal i henhold til funksjonsdiagrammet og det logiske diagrammet. Deretter kontrolleres kodingen på kodearkene mot flow-diagrammet; særlig må en være påpasselig når det gjelder initialordrer, modifiseringsordrer, lese- og puncheordrer, multiplikasjon og divisjon.

En annen metode som riktignok er ganske tidkrevende, er rekonstruksjon av flow-diagrammet og det logiske diagrammet på grunnlag av kodingen. Denne metoden er, hvis den blir utført av en annen programmerer, ypperlig når det gjelder å finne logiske feil, men krever så lang tid at den sjelden er brukbar i praksis unntatt kanskje som et ledd i utdanningen av nye programmerere.

3.3.2. Testing på DEUCE

Det er liten hensikt i, og ofte svært tidsødende, å starte testing av program på DEUCE uten å ha kjennskap til de hjelpemidlene som står til rådighet under testingen. Disse hjelpemidlene kan være innebygd i maskinen (Programme Display, Request Stop osv.) eller det kan være spesielle hjelpeprogrammer, f.eks. POST MORTEM.

En bør gjøre til regel alltid å legge opp en plan for testingen før en forsøker programmet på DEUCE, slik at en vet hva en skal se etter, når en skal se etter dette og hvilke aksjoner som må gjøres i de forskjellige tilfellene. Denne planen bør skrives ned i form av en instruks; det høres kanskje pussig ut at en skal skrive instruks til seg selv, men det er på grunn av at mye

testetid har vært bortkastet eller dårlig utnyttet nettopp fordi test-instruks har manglet.

Før testingen kan starte, må en også ha laget seg et sett med prøvekort. Dette prøvekortsettet bør lages slik at en med så få kort som mulig kan få testet alle de kritiske punktene i programmet. De første prøvekortene kan være enkle slik at en får testet den gjennomgående logikken i programmet, mens de andre må lages slik at en får med grensetilfellene.

Under testingen på DEUCE kan en få to slags informasjoner, visuelle eller punchede. De visuelle informasjonene får en på lampene og monitorene på kontrollbordet; disse kan imidlertid normalt bare leses av når DEUCE har stoppet. Når programmet går for fullt er det oftest svært vanskelig å få noe ut av lampene og monitorene, men særlig under programinnlesingen bør en følge nøye med på IS-lampene fordi de fleste innlesingsrutinene danner et eget karakteristisk mønster på IS-lampene. Et avvik fra mønsteret vil da kunne indikere at korttrekkfølgen er gal, at det mangler kort eller at en triade har gale "filling instructions".

Det er uhyre vanskelig å si hvor et program går galt når det kjøres for full fart. En må derfor sørge for at programmet stopper ved bestemte punkter slik at programmet blir oppdelt, og på den måten teste hver del for seg. En kan få maskinen til å stoppe på en bestemt ordre ved å lage denne om til en stoppordre i programkortene, eller en kan benytte "Request Stop". "Request Stop"-utstyret er nærmere beskrevet i kapitelet om kontrollbordet. Hvis en lager stoppordrer ved å forandre i programkortene, bør disse stoppordrene velges slik at når en kommer til en slik ordre, må en kunne, ved å iaktta noen lamper eller monitors, fastslå om den delen av programmet som har passert, har vært riktig eller gal. Det er imidlertid en del ulemper ved dette systemet med midlertidige stoppordrer:

- a) En midlertidig stoppordre kan være svært ubehagelig hvis den skal gjentas flere ganger før en kommer til det punkt som skal testes.
- b) En midlertidig stopp vil ikke stoppe DEUCE like før eller mens et kort blir avlest eller punchet fordi Single-Shot da blir sendt ut fra 528.

"Request Stop" skiller seg fra midlertidige stoppordrer på følgende måte:

- a) Enhver ordre kan forandres til midlertidig stoppordre, og DEUCE vil da stoppe før denne ordren blir utført. En "Request Stop" kan en derimot ikke ha på en bestemt ordre, men på en kombinasjon av en eller flere av P_1 , NIS, Source, Destination og P_{15} . Alle ordrene som inneholder denne bestemte kombinasjonen, vil stoppe DEUCE. Videre kan bare én slik kombinasjon spesifiseres av gangen; riktignok kan denne kombinasjonen forårsake stopp på flere ordrer i programmet (ofte også uventede steder).

- b) "Request Stop" på disse Trigger-ordrene: 0-24, 1-24, 2-24, 10-24 og 12-24 kan influere på resultatet av disse ordrene.
- c) "Request Stop" må settes opp eller forandres mens programmet er i maskinen; på denne måten kan en "Request Stop" annulleres etter at DEUCE har gått gjennom en løkke noen ganger.
- d) "Request Stop" kan settes opp på en ordre som en ikke vet adressen til. Hvis en for eksempel er klar over at innen en viss del av programmet blir noe uventet overført til TS 16, kan en sette opp "Request Stop" på D 16. DEUCE vil da stoppe på alle ordrer som har Destination 16.

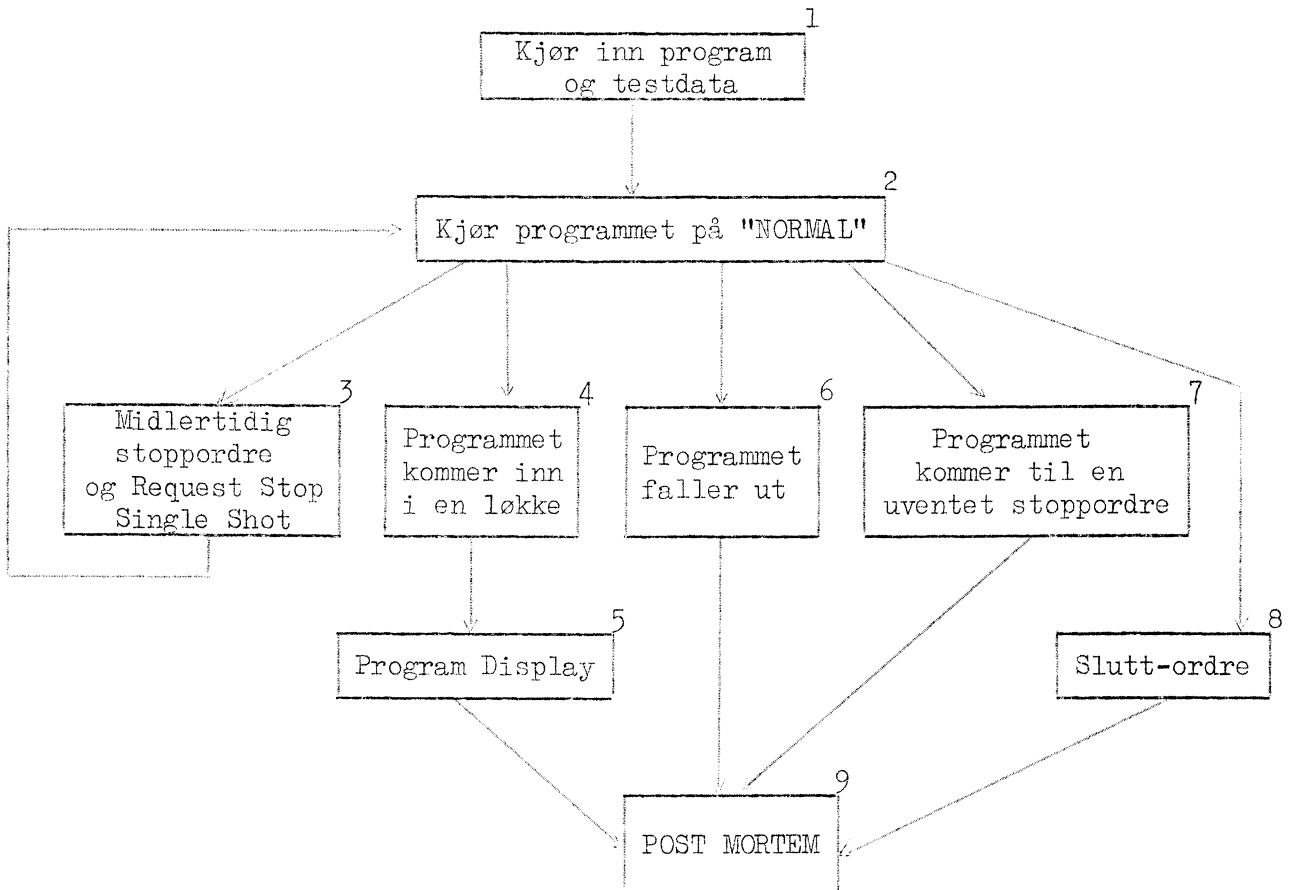
En annen metode for programtesting ved hjelp av visuelle opplysninger er å ha stopp-bryteren på STOP (eller AUG. STOP) og la DEUCE utføre en eller noen få ordrer av gangen. Denne metoden brukes ofte av nye og uøvde programmerere, men metoden kan sjelden forsvares. Det kan hende at erfarne programmerere også bruker denne metoden, men da fordi alle andre metoder ikke har ført fram.

Erfaringen viser at informasjonen som blir punchet i kort, har mange fordeler framfor visuelle opplysninger. En slipper å oppta verdifull DEUCE-tid mens en leser av lamper eller monitors; dessuten blir selve avlesingen sikrere fordi den kan foregå utenom DEUCE. Testing på DEUCE er ofte en ganske hektisk affære på grunn av den snau tiden som blir en tildelt. De vanligste og viktigste hjelpemidlene for punching av informasjoner under testing er PROGRAM DISPLAY (spesialutstyr i DEUCE) og POST MORTEM (et bestemt program).

PROGRAM DISPLAY er beskrevet nærmere under kapitlet om kontrollbordet. PROGRAM DISPLAY-kortene kan kontrolleres mot flow-diagrammet. Hvis programmet har "falt ut", dvs. kommet til ordren "0. 0 - 0. 0. 0. 0. x" under Program Display, bør en forsøke å finne fram til feilen ved å undersøke det siste kortet som ble punchet og arbeide seg framover i Program Display-kortene.

POST MORTEM-programmet er laget slik at hele innholdet av DEUCE, både hurtigminnet og magnettrommelen, blir punchet ut. Hver gang en avslutter testingen av et program bør en ta POST MORTEM. Ved å studere dette kan en finne ut mye om hvordan programmet har gått.

En kan sette opp en generell metode for testing av program i et diagram:



M e r k n a d e r:

- Boks 1: Programmet bør lages slik at den første ordren alltid er en stopp-ordre.
- " 3: Ofte er en interessert i etter en slik stopp å få punchet resten av programmet som Program Display.
- " 5: Det er vanligvis bortkastet tid å punche flere PROGRAM DISPLAY-kort.
- " 9: En bør alltid sørge for å ta POST MORTEM før en forlater DEUCE. Hvis en ikke gjør dette, vil en finne at det inntreffer mye oftere at en ikke har POST MORTEM, men trenger den, enn at en har tatt POST MORTEM unødvendig.

3.4. Kjøring av programmet

Det arbeidet som blir gjort med programmet før en kommer så langt som til prøve- eller produksjonskjøring, blir i det alt vesentligste utført av programmereren selv. Selve kjøringen av programmet med aktuelle data blir imidlertid oftest overlatt DEUCE-operatøren. Før operatøren kan starte kjøringen, må han ha tilstrekkelig materiale både om programmet og kjøringen. Han må ha følgende:

- a) Operatørinstruks,
- b) pluggbord, ferdig plugget eller pluggetegning/beskrivelse,

- c) programkort,
- d) datakort,
- e) blanke resultatkort.

Operatøren må dessuten ha klare opplysninger om:

- a) hvem som er ansvarlig for arbeidet og kan svare på spørsmål,
- b) hva som skal gjøres i tilfelle maskinfeil,
- c) hva som skal gjøres i tilfelle kjøringen ikke kan fortsette på grunn av feil i programmet eller uorden eller ubrukbare detaljkort,
- d) hvor lang tid DEUCE kan disponeres til arbeidet,
- e) hvem som skal ha beskjed i tilfelle kjøringen avsluttes før den disponible tiden er oppbrukt slik at maskinen er ferdig,
- f) hvor data- og resultatkort skal leveres.

3.4.1. Operatørinstruks

For alle programmer som skal kjøres på DEUCE, skal det lages en operatørinstruks. Denne skal inneholde så mange opplysninger at en øvet operatør kan kjøre programmet uten tilsyn. Instruksene bør inneholde følgende punkter:

1. Beskrivelse av kortene

1.1 Programkort

Program nr. og antall kort

1.2 Datakort

- a) Hvordan eskene er merket
- b) Hvilke korttyper som forekommer og hvilke klisjéer de er punchet på
- c) Rekkefølgen (sorteringen) av kortene
- d) Sluttkort

1.3 Resultatkort

- a) Hvordan eskene skal merkes
- b) Hvilken klisjé det skal punches på og hvilke korttyper som kan forekomme
- c) Rekkefølgen av kortene

2. Pluggbord

En pluggbordtegning må følge med som bilag til instruksene. Hvis en har stående fast pluggbord for kjøring av programmet, bør det stå i instruksene hvordan dette er merket.

3. Innstilling av brytere for gangpunching og kortnummerering.

4. Innlesing av program og start

For de fleste programmene kan en her skrive på NIS, S-D, Ch (dvs. en bestemt ordre), Single-Shot. Operatøren skal da gjøre følgende:

- a) sette riktig pluggbord i 528,
- b) sette alle bryterne på kontrollbordet i riktig utgangsstilling,

- c) legge programkortene med en passende mengde datakort bak, i 528,
- d) legge blanke kort i punchebanen og trykke på "Run in Punch" på 528,
- e) trykke på "Initial Input",
- f) vente til programkortene er lest inn, ta programkortene ut, kontrollere at programmet stopper på riktig ordre og gi et Single-Shot.

Hvis denne framgangsmåten ikke kan brukes, må en gi en nøyaktig beskrivelse av hva som skal gjøres.

5. Kjøring av datakortene

Her må en gi nøyaktige opplysninger om hva som kan inntreffe under kjøringen av programmet og hva operatøren skal gjøre i hvert enkelt tilfelle. En må også gi beskjed om hva som skal gjøres når noe uforutsett inntreffer. De forskjellige typer av stopp må beskrives.

Følgende form for oppstilling anbefales:

a) Programmerte stopp

I de fleste programmer vil det være en del programmerte stopper. Disse stoppene må spesifiseres på følgende måte: Hvilken ordre maskinen stopper på, hvilke OS-lamper som eventuelt skal lyse, hva dette betyr, hva som skal gjøres hvis det som lampene indikerer virkelig har skjedd og hva som skal gjøres hvis det som lampene indikerer ikke har skjedd (maskinfeil, f.eks. lesefeil).

b) Andre stopp

En må her gi klar og tydelig beskjed om hva som skal gjøres hvis maskinen får en uforutsett stopp eller kommer inn i en løkke. Det må her spesifiseres om og i hvilket tilfelle det skal tas "POST MORTEM", hvor mange ganger en skal prøve å kjøre om igjen før en gir seg osv.

4. LETTKODINGSSYSTEMER FOR DEUCE

4.1. Behov for lettkodingssystemer

Programmering for DEUCE krever et inngående kjennskap til maskinen og dens virkemåte. For at programmeringen ikke skal bli altfor tidkrevende og kostbar, kreves derfor programmeringsspesialister som daglig holder sine kunnskaper vedlike. Erfarne programmerere vil trolig i lange tider være en mangelfaktor. For engangsarbeider vil dessuten programmeringen i svært mange tilfelle bli så kortbar at den kan gjøre elektronisk bearbeiding direkte ulønnsom.

For å gjøre DEUCE tilgjengelig også for brukere som har engangsproblemer hvor vanlig programmering vil bli uforholdsmessig kostbar eller når det ikke

er mulig å få hjelp av en erfaren programmerer, er det utarbeidd en rekke lettkodingssystemer hvor en større eller mindre del av programmeringsarbeidet er overlatt til maskinene. Disse er som regel lettere å lære og eliminerer en rekke fallgruber og feilkilder i programmeringen. Dette oppnås imidlertid som regel på bekostning av programmets effektivitet.

4.2. Gruppering av forskjellige lettkodingssystemer

Det finnes forskjellige typer av lettkodingssystemer for DEUCE. Figur a gir en oversikt over de forskjellige systemer og deres oppbygning.

Systemets oppbygning	Interpretive systemer	Translative systemer
Systemer som nytter elementær-operasjoner	-	Samleprogram Oversettingsprogram
Systemer som nytter masteroperasjoner	Samleprogram Interpretivt program	Samleprogram Oversettingsprogram

Fig. a.: Gruppering av forskjellige typer lettkodingssystemer og deres oppbygning.

Det skilles her mellom systemer som nytter elementæroperasjoner, dvs. de samme operasjoner som DEUCE, men hvor ordrones struktur ellers kan være forskjellig. Disse systemer krever kjennskap til vanlig programmering for DEUCE. Den annen hovedgruppe er systemer som nytter masteroperasjoner, dvs. operasjoner som ikke korresponderer med maskinens operasjoner. Disse systemer er som regel enkle å lære og tilpasses bestemte problemtyper.

Innenfor hver av disse hovedgrupper av systemer kan en videre skille mellom interpretive og translative systemer. De interpretive systemene leser inn lettkodingsprogrammet og oversetter det skrittvis til maskinens språk ved å innkalle lagrede subrutiner hver gang det nyttes til produktiv kjøring. De translative systemer derimot foretar oversettingen en gang for alle og før maskinen tar fatt på den produktive kjøringen. Disse systemene er derfor langt mer komplisert å lage enn de interpretive, men gir også langt mer effektive program, idet en slipper å kaste bort tid på oversetting av hvert trinn i programmet og for hver gang det skal nyttes.

Felles for alle lettkodingssystemer er at de må ha et systemprogram som maskinen arbeider etter når den skal utføre sin del av programmeringen. Systemprogrammet består ofte av flere deler:

- a) Samleprogrammet hvis oppgave er å oversette de symboler som nyttes i lettkodingen til en for maskinen høvelig form, ta vare på eventuelle subrutiner som kan nyttes og påvise punchefeil, ulogiske masterordrer etc.
- b) Et interpretivt program som må nyttes til den skrittvis og gjentatte oversetting av lettkodingsprogrammet under den produktive kjøring. Innebygd i det interpretive programmet finnes gjerne spesielle rutiner med sikte på å gjøre påvisning av feil så enkel som mulig.
- c) Translative programmer som en gang for alle oversetter lettkodingsprogrammet og vanligvis puncher ut det ferdige DEUCE-program før den produktive kjøringen finner sted. Innebygd i det translative programmet finnes også ofte rutiner med sikte på å forenkle programtestingen.

Samleprogrammet kan ofte være bygd sammen med henholdsvis det interpretive eller det translative program. For interpretiv-systemer er det imidlertid særlig en fordel at det er skilt ut som en selvstendig del slik at en kan unngå å kaste bort tid på å utføre denne samme prosessen hver gang programmet skal kjøres.

Figur b viser hvilke forskjellige systemer som er utviklet for DEUCE og hvilken av de ovenfornevnte typer de hører til.

Systemets navn	Interpretive systemer	Translative systemer
Systemer som nytter elementær-operasjoner	-	Detailed Coding Scheme STAC (Storage Allocation and Coding Programme)
Systemer som nytter masteroperasjoner	Alfasystemet GIP-systemene TIP-systemene	Alfasystemets Translator Easicode SODA

Fig. b.: De forskjellige lettkodingssystemer for DEUCE.

4.3. Detailed Coding Scheme

Detailed Coding Scheme er et translativt system som bygger på elementær-operasjoner. Systemet overlater til DEUCE å utføre all den rutinemessige "timing" av ordrene. Ordrene punches desimalt, en ordre pr. kort omtrent på den form de har på flow-diagrammet før en tar fatt på kodingen.

Systemet tillater innlesing av såvel binære som desimalt punchede konstanter og DEUCE-kodede subrutiner, men kan vanskelig nyttes i programmer med automatisk ordremodifisering. DEUCE bearbeider det desimalt punchede arbeids-

program m.m. ved hjelp av et kombinert samle- og translativt program, og puncher ut et binært DEUCE-program eller eventuelt feilkort som markerer logiske feil som er påvist under oversettingen.

Systemet har vært i bruk i England i flere år, men da det opprinnelig er laget for tidligere modeller av DEUCE, er det ikke så effektivt for vår DEUCE selv om det er fullt brukbart.

4.4. Storage Allocation and Coding Programme

Et nyere og mer nyttig system er STAC som også er et translativt system som nytter elementøperasjoner. Mens en for Detailed Coding Scheme var nødt til å allokere ordrene og de forskjellige variable til adresser i maskinen, er dette unødvendig i STAC-systemet hvor både adresseallokering og detalj-koding utføres automatisk av et kombinert samle- og translativt program.

I STAC er de alvorligste svakheter ved Detailed Coding Scheme fjernet, og systemet skulle kanskje være til nytte for oss.

4.5. Alfasyttemet

Alfasystemet er et interpretivt lettkodingssystem for DEUCE som bygger på masteroperasjoner. Det har en meget enkel ordrestruktur av typen "A er lik funksjon av B og C", hvor A, B og C er symbolske adresser. Det krever ikke kjennskap til oppbyggingen av DEUCE eller maskinens ordresystem. Hver masterordre representerer gjennomsnittlig ca. 10 DEUCE-ordrer. Systemet arbeider utelukkende med flytende aritmetikk som gjør det helt unødvendig å tenke på over-flow. Det er derfor vel egnet til programmering av mindre engangsarbeider som krever rask gjennomføring og hvor det ikke betyr noe at programmet er 5-10 ganger senere enn et tilsvarende DEUCE-program.

Alfasystemet består av et adskilt samleprogram og interpretert program. Samleprogrammet leser og omformer de alfa-numerisk punchede masterordrer til en binær representasjon, samtidig som det kontrollerer at alfaprogrammet ikke inneholder punchefeil eller ulogiske masterordrer. Under den produktive kjøringen omsetter så DEUCE skrittvis dette binært representerte program til serier av DEUCE-ordrer ved hjelp av det interpretive program.

Data kan leses inn enten som desimaltall med flytende komma eller som standard flytende binærtall. Alfaprogrammet er nærmere beskrevet i Statistisk Sentralbyrås håndbøker nr. 16.

4.6. General Interpretive Schemes

GIP heter en serie med interpretive systemer som opprinnelig ble utviklet spesielt med sikte på behandling av matrix-problemer o.l., men som senere

har vist seg å være av generell verdi. GIP-ordrene er av formen "A B C R", og er av to typer. Den ene typen består av administrative masterordrer og innlesings- og utpunchingsordrer. Den andre typen refererer seg til operasjoner som er nærmere definert ved visse standardprogrammer kalt brikker som kan inkluderes i det interpretive program etter behov. Dersom det ikke er nødvendig å lage en ny brikke, forutsetter ikke bruk av GIP vesentlig kjennskap til DEUCE-programmeringen.

Systemet består av et kombinert samle- og interpretivt program. Systemet kan som nevnt suppleres etter behov med brikker, og oversetter en for en av GIP masterordrene til en serie med administrative DEUCE-ordrer eller til et brikkeprogram. Det er lagt stor vekt på at hver brikke skal være mest mulig effektiv, og GIP-programmer vil derfor stort sett være meget raske til tross for at de utføres via et interpretivt system som kaller inn brikkene fra trommelen etter hvert som de skal nyttes.

4.7. Tabular Interpretive Schemes

TIP er et annet sett av interpretive systemer som opprinnelig ble utviklet for demonstrasjonsformål ved en engelsk flyfabrikk, men som senere har vist seg å være til nytte for ingeniører og andre som er vant til å sette opp sine regneproblemer på en tabellarisk form. Ved hjelp av TIP kunne disse få løst sine regneproblemer uten å sette seg inn i den krevende DEUCE-programmeringen. Ved programmering etter TIP-systemene tenker en seg maskinen inndelt som et ark med en rekke linjer og kolonner hvor alle operasjoner gjentas for hver linje. Strukturen i masterordrene er av formen "A B C D", hvor A og B er tall som refererer seg til de kolonner operandene skal hentes fra, C til hvilken kolonne resultatene skal skrives og hvor D gir operasjonsnummer.

Det kan inkluderes spesielle brikker i det interpretive program på samme måte som for GIP. I motsetning til de fleste andre lettkodingssystemer kan masterordrene i TIP også punches binært.

4.8. Alfasytemets translator

Om kort tid vil English Electric offentliggjøre et translativt system som bygger på de samme masterordrene som det interpretive alfasytemet. Dette translativt system nytter også det samme samleprogrammet som det interpretive systemet. Det vil derfor være mulig å dra nytte av alle automatiske tester av ordrer som samleprogrammet gir, og om det er ønskelig, kan en altså først kjøre prøver i det interpretive system før en foretar oversettingen. Dette innebærer at programmereren kan teste og gjøre korreksjoner i sitt program bare ved å kjenne til masterordrene og uten å sette seg inn i DEUCE-programmet.

Alfasystemets translator oversetter resultatet av samleprogrammet til et DEUCE-program etter en inngående analyse med sikte på å få et så effektivt som mulig DEUCE-program.

4.9. Easicode

Easicode er et translativt system som nå er under utvikling. Det bygger på masterordrer av omtrent samme type som alfasystemet, men nytter fast komma-aritmetikk.

4.10. SODA

SODA er et translativt system som nylig er utarbeidd i Australia. I motsetning til alle de andre lettkodingssystemer for DEUCE nytter SODA en såkalt en-adresse masterordre; det har et kombinert samle- og translativt program, og oversettingen fra masterordre til DEUCE-program foregår således i en kjøring.

Deuce Program nr.

D.L.		Spor		Kort nr.		D.L.		Spor		Kort nr.		D.L.		Spor		Kort nr.					
mc	NIS	S	D	C	W	T	mc	NIS	S	D	C	W	T	mc	NIS	S	D	C	W	T	
																					Y
																					X
																					0
																					1
0							0							0							2
1							1							1							3
2							2							2							4
3							3							3							5
4							4							4							6
5							5							5							7
6							6							6							8
7							7							7							9
8							8							8							Y
9							9							9							X
10							10							10							0
11							11							11							1
12							12							12							2
13							13							13							3
14							14							14							4
15							15							15							5
16							16							16							6
17							17							17							7
18							18							18							8
19							19							19							9
20							20							20							Y
21							21							21							X
22							22							22							0
23							23							23							1
24							24							24							2
25							25							25							3
26							26							26							4
27							27							27							5
28							28							28							6
29							29							29							7
30							30							30							8
31							31							31							9

DEUCE

Dato		Program											Ark. nr.	
m.o.	DL 1	DL 2	DL 3	DL 4	DL 5	DL 6	DL 7	DL 8	DL 9	DL 10	DL 11	DL 12	m.o.	
0													0	
1													1	
2													2	
3													3	
4													4	
5													5	
6													6	
7													7	
8													8	
9													9	
10													10	
11													11	
12													12	
13													13	
14													14	
15													15	
16													16	
17													17	
18													18	
19													19	
20													20	
21													21	
22													22	
23													23	
24													24	
25													25	
26													26	
27													27	
28													28	
29													29	
30													30	
31													31	

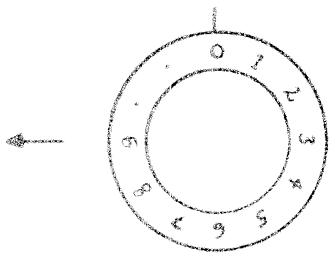
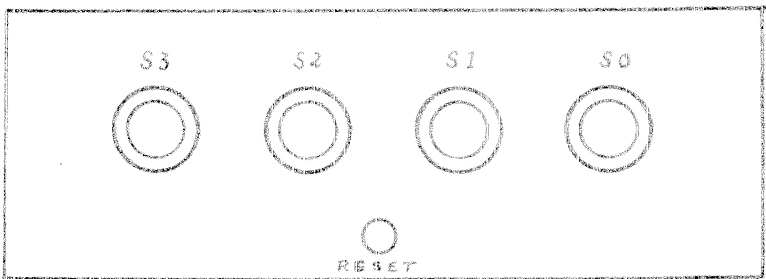
Blokk

Spor	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0																
1																
2																
3																
4																
5																
6																
7																
8																
9																
10																
11																
12																
13																
14																
15																

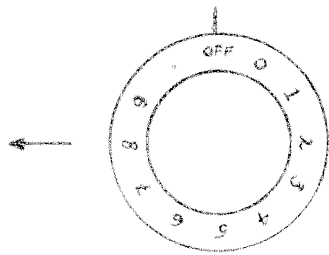
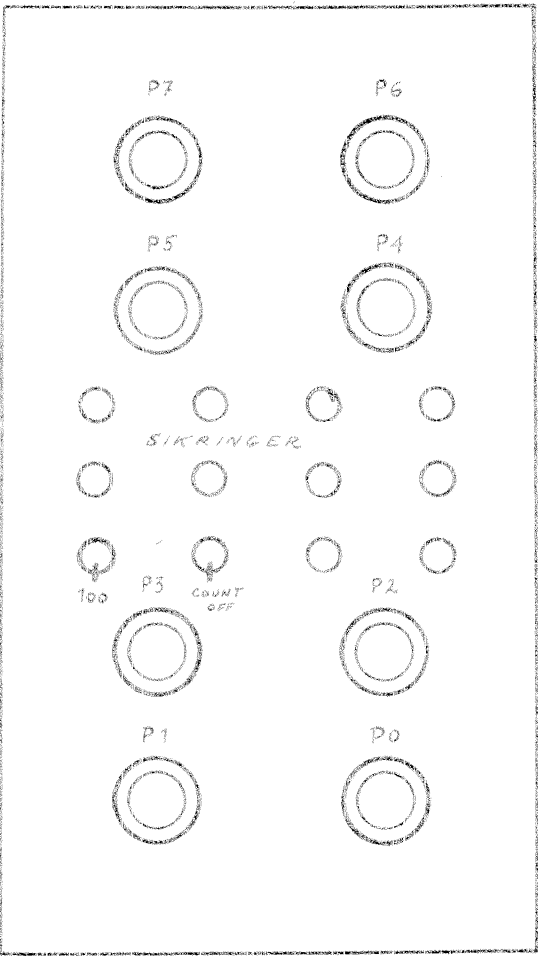
BRYTERE PÅ IBM 528

INITIAL INPUT	RE AD DY	PR E AD Y	MC AR D S S E D	ST AC K E R	RUN IN READ	RUN OUT READ	STOP	RUN OUT PUNCH	RUN IN PUNCH
------------------	----------------	--------------------	-----------------------------------	-------------------------	----------------	-----------------	------	------------------	-----------------

MANØVRERINGSBRYTERE OG LAMPER



KORTNUMMERERING



GANGPUNCHING OG HASTIGHET

PLUGGBORDET PÅ IBM 528

10										20										30										40										50										60										70										80										90										100																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																													
1										2										3										4										5										6										7										8										9										10										11										12										13										14										15										16										17										18										19										20										21										22										23										24										25										26										27										28										29										30										31										32										33										34										35										36										37										38										39										40										41										42										43										44										45										46										47										48										49										50										51										52										53										54										55										56										57										58										59										60										61										62										63										64										65										66										67										68										69										70										71										72										73										74										75										76										77										78										79										80										81										82										83										84										85										86										87										88										89										90										91										92										93										94										95										96										97										98										99										100									
1										2										3										4										5										6										7										8										9										10										11										12										13										14										15										16										17										18										19										20										21										22										23										24										25										26										27										28										29										30										31										32										33										34										35										36										37										38										39										40										41										42										43										44										45										46										47										48										49										50										51										52										53										54										55										56										57										58										59										60										61										62										63										64										65										66										67										68										69										70										71										72										73										74										75										76										77										78										79										80										81										82										83										84										85										86										87										88										89										90										91										92										93										94										95										96										97										98										99										100									
1										2										3										4										5										6										7										8										9										10										11										12										13										14										15										16										17										18										19										20										21										22										23										24										25										26										27										28										29										30										31										32										33										34										35										36										37										38										39										40										41										42										43										44										45										46										47										48										49										50										51										52										53										54										55										56										57										58										59										60										61										62										63										64										65										66										67										68										69										70										71										72										73										74										75										76										77										78										79										80										81										82										83										84										85										86										87										88										89										90										91										92										93										94										95										96										97										98										99										100									
1										2										3										4										5										6										7										8										9										10										11										12										13										14										15										16										17										18										19										20										21										22										23										24										25										26										27										28										29										30																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																					

25	— FIELD ALPHA —	32	— FIELD BETA —	7	— FIELD BETA —	14	
0	0	0	0	0	0	0	0
41	— READ BRUSHES —	50	— READ BRUSHES —	60	—	62	
0	0	0	0	0	0	0	0
41	— READ BRUSHES —	50	— READ BRUSHES —	60	—	62	
0	0	0	0	0	0	0	0
40	— READ STORAGE LOCATION —	31	— READ STORAGE LOCATION —	21	—	19	
0	0	0	0	0	0	0	0

7	FIELD ALPHA	15	FIELD ALPHA	24
23	READ BRUSHES	31	READ BRUSHES	40
23	READ BRUSHES	31	READ BRUSHES	40
58	READ STORAGE LOCATION	50	READ STORAGE LOCATION	41

15	FIELD BETA	23	FIELD BETA	32
0	FIELD BETA	0	FIELD BETA	0
63	READ BRUSHES	71	READ BRUSHES	80
0	READ BRUSHES	0	READ BRUSHES	0
63	READ BRUSHES	71	READ BRUSHES	80
0	READ BRUSHES	0	READ BRUSHES	0
18	READ STORAGE LOCATION	10	READ STORAGE LOCATION	1
0	READ STORAGE LOCATION	0	READ STORAGE LOCATION	0

0	0	0	P7	0	P5	P4	P3	P2	P1	0	S0	S2	S1	S0	1	PH	+
1	—	—	PUNCH MAGNETS	—	—	—	10	—	—	—	PUNCH MAGNETS	—	—	—	—	—	20
1	—	—	PUNCH MAGNETS	—	—	—	10	—	—	—	PUNCH MAGNETS	—	—	—	—	—	20
80	—	—	PUNCH STORAGE LOCATION	—	—	—	—	—	—	—	70	—	—	—	—	—	60

5	FIELD ALPHA	14	FIELD ALPHA	24
21	PUNCH MAGNETS	30	PUNCH MAGNETS	40
21	PUNCH MAGNETS	30	PUNCH MAGNETS	40
60	PUNCH STORAGE LOCATION	50	PUNCH STORAGE LOCATION	24

25	FIELD ALPHA	32	1	FIELD BETA	10	7
41	PUNCH MAGNETS	50		PUNCH MAGNETS	60	3
44	PUNCH MAGNETS	50		PUNCH MAGNETS	60	3
40	PUNCH STORAGE LOCATION	30		PUNCH STORAGE LOCATION	21	

[illegible]

