

STATISTISK SENTRALBYRÅS HÅNDBØKER

Nr. 16

Oslo, 15. januar 1959

A L F A S Y S T E M E T

E T L E T T K O D I N G S S Y S T E M F O R D E U C E

Tilføyelser til Statistisk Sentralbyrås Håndbok nr. 16.

Mens håndboken har vært til innbinding, er det kommet nye opplysninger om Alfsystemet som gjør det nødvendig med følgende endringer i heftet:

- s. 5, linje 16: "kortsett av" strykes.
- s. 8, linje 10: Setningen "I tilknytting til ..." og resten av avsnittet strykes.
- s.16, fotnoten: Her føyes "T punches som V" til
- s.17, linje 15: Her føyes til "X-konstanter punches med fortegn i kol. 47, mantissen i kol. 48-56, eksponentens fortegn i kol. 58 og eksponenten i kol. 59-61".
- s.25,subrutiner: Instruksjonene "SUBROUTINE" og "END OF subroutine" skal ha henholdsvis 19 og 20 som operasjonsnummer (r-nummer), ikke 60 og 61.

A L F A S Y S T E M E T

Et lettkodingssystem for DEUCE

Statistisk Sentralbyrå

Oslo, 1959

F o r o r d

Statistisk Sentralbyrå har tidligere utarbeidd en håndbok i Programmering for DEUCE. Med dette sender Byrået ut en oversikt over et lettkodingssystem, Alfsystemet, med sikte på å gi mer tilfeldige brukere av DEUCE et middel for å programmere sine oppgaver selv.

Håndboken er utarbeidd ved Planleggingskontoret for maskinell databearbeiding, og som vanlig har Norsk Regnesentral bidratt med merknader til manuskriptet.

Statistisk Sentralbyrå, Oslo, 15. januar 1959

Petter Jakob Bjerve

Svein Nordbotten

I n n h o l d

	Side
1. INNLEDNING	4
2. ALFAINSTRUKSER	5
3. GENERELLE INSTRUKSJONER	6
3.1. Input- og outputinstruksjoner	6
3.2. Elementære regneinstruksjoner	9
3.3. Administrative instruksjoner	10
3.4. Telleinstruksjoner	11
3.5. Modifiseringsinstruksjon	12
3.6. Sammenlikningsinstruksjoner	12
4. SPESIELLE INSTRUKSJONER	13
5. SUBRUTINER	15
6. REGLER FOR PUNCHING	16
6.1. Punching av instruksjoner	16
6.2. Punching av data	17
7. ALFAPROGRAMMET OG BETJENING AV DEUCE	17
7.1. Samleprogrammet	17
7.2. Det interpretive program	19
8. KAPASITETSBEGRENSNINGER	22
9. OVERSIKT OVER INSTRUKSJONENE	24

1. INNLEDNING

Forskere og andre med spesielle regneproblemer har ofte behov for å kunne nytte de muligheter som en elektronisk siffermaskin innebærer. Men det kan være vanskelig for den enkelte forsker å få en erfaren programmerer til å utarbeide et program. Det kan dessuten ta uforholdsmessig lang tid for programmereren å sette seg inn i spesielle problemstillinger. På den annen side kan det kreve for meget av forskerens tid om han må sette seg ned for å lære de nødvendige programmeringsfinesser selv. Han vil bare av og til ha bruk for en slik lærdom og vil derfor ikke få den rutinemessige erfaring som er nødvendig for effektiv programmering. Det er derfor ønskelig med kodesystemer som er enklest mulig å lære og hvor det i størst mulig utstrekning er overlatt til maskinen selv å utarbeide alle de detaljerte finesser som et fullstendig maskinprogram består av.

Som ved de fleste andre elektroniske siffermaskiner, er det for DEUCE utarbeidd flere slike lettkodingssystemer. GIP (General Interpretive Programme) og TIP (Tabulator Interpretive Programme) er systemer som særlig er egnet for henholdsvis matriksberegninger og problemer som lett lar seg stille opp i tabellarisk form. Alfasytemet er et tredje system som tar sikte på en mer generell anvendelse og kan derfor ikke konkurrere med de to andre innen de områder disse er spesielt beregnet på. Det skulle imidlertid være nyttig for alle som fra tid til annen har behov for å få utført spesielle beregninger uten å kunne disponere erfarne programmerere. Systemet er meget enkelt og forutsetter at problemet settes opp nesten analogt med en instruks til en beregner. Det er derfor lett å lære og krever i alminnelighet bare spesifisering av omkring en tiendedel av det antall operasjoner som er nødvendig i et vanlig DEUCE-program for samme problem. Dette betyr en betraktelig reduksjon av mulige feilkilder. På den annen side vil et arbeid utført ved hjelp av Alfasytemet kreve mer maskintid enn utført med et vanlig DEUCE-program. Et program som bygger på Alfasytemet må en derfor regne med kan kreve 5-10 ganger så lang kjøretid som et håndkodet DEUCE-program. Alfasytemet legger dessuten selv beslag på en stor del av lagerkapasiteten i maskinen som ikke vil være tilgjengelig for selve beregningen.

I et hvert tilfelle bør det således foretas en avveining av de foreliggende muligheter. Selv etter et Alfakodet program vil DEUCE utføre en mengde beregninger i løpet av en time, mens det kan kreve dager av kostbar programmerertid å utarbeide et fem minutters DEUCE-program for samme problem.

Bruk av Alfasystemet krever ikke kjennskap til oppbygningen av DEUCE eller maskinens ordresystem. For å unngå forvekslinger med DEUCE-programmering har en i framstillingen av Alfasystemet unnlatt å bruke betegnelsene program, ordre, lagerposisjon, adresse etc. I forbindelse med Alfasystemet nyttes istedet instruks, instruksjon, variable og parametre m.m. I stedet for lagring av et tall eller et resultat i en bestemt posisjon, nytter en formuleringen "definisjon av en variabel eller parameter".

2. ALFAINSTRUKSER

En Alfainstruks er en serie av instruksjoner som er skrevet i en alfanumerisk form fastlagt slik at de skal ha størst mulig likhet med formen i tradisjonelle regneinstruksjoner. Alfaprogrammet er et interpretivt program som på forhånd lagres i maskinen. Hver instruksjon er utformet på engelsk. De blir punchet i hullkort og ved hjelp av Alfaprogrammet omsetter DEUCE instruksjonene til en eller flere DEUCE-operasjoner. Skal instruksjonen nyttes flere ganger, kan en spare maskintid ved å la operatøren ta vare på kortsett av instruksens binære utgave.

I hver instruksjon kan det være opptil tre spesifiserte variable eller parametre. De variable og parametrene er betegnet henholdsvis med X_i , ($0 \leq i \leq 2.383$) og N_i , ($0 \leq i \leq 63$). De variable er samlet i blokker a 32 i rekkefølgene $X_1 \dots X_{32}$, $X_{33} \dots X_{64}$, osv. Av hensyn til regnehastigheten er det fordelaktig å innrette seg slik at de variable som det skal arbeides med i sammenheng er samlet i samme blokk.^{x)}

Alle variable og parametre må skrives med flytende desimalkomma, dvs. på formen $A \cdot 10^B$ hvor tallområdet for de variable er bestemt ved $1 \leq |A| < 10$ og $0 \leq |B| \leq 999$. For parametrene er tallområdet dessuten begrenset til hele tall med tallverdi $|N_i| \leq 2 \cdot 10^9$. Null er definert ved at både $A=0$ og $B=0$, og tallenes nøyaktighet er begrenset ved 9 desimalsiffer i A.

Instruksjonene er bygd opp etter et bestemt mønster som vist i Fig.1. Bortsett fra betegnelsen i kolonnen FUNCTION, som skrives med store bokstaver, kan ordlyden i instruksjonene formuleres fritt, også på norsk.

x) For å øke regnehastigheten, har Alfasystemet også 8 hjelpevariable T_i , ($i=1 \dots 8$) som er meget raskt tilgjengelig. Disse hjelpevariable kan nyttes som x-variable i alle instruksjoner unntatt input- og outputinstruksjoner.

Instruksark

Nr.	r	R	Tekst	A	Tekst	B	FUNCTION	C	Tekst	D	P	O/S
1	03	17		X33	is	X35	MULTIPLIED by	X36				
2	01			X34	is	X34	PLUS	X33	Punch		P	
3	10				Count	N5	UP TO	N3	Jumping to	R17		
4	14						STOP					

Figur 1

Betydningen av feltene er:

- Nr. : Fortløpende nummerering av instruksjonene. Kan utelates.
- r : Instruksjonens kodennummer. Kan utelates etter avtale med operatøren.
- R : Referansenummer som må nyttes om en andre steder i instruksjonen ønsker å vise til en bestemt instruksjon. Skrives vanligvis som Ri, ($1 \leq i \leq 99$).
- A : Vanligvis resultatet av en operasjon. Betegnes symbolsk som Xa, Na eller a for henholdsvis variabel, parameter eller konstant.
- B og C: Vanligvis to operander som symbolsk betegnes med Xb eller Xc, Nb og Nc, eller b og c.
- D : Henvisning til et referansenummer.
- P : Resultatet av instruksjonen skal punches.
- S : Maskinen stopper alltid og må ha en spesiell impuls fra operatøren før den utfører instruksjonen.
- O : Etter ønske kan maskinen stoppes og må da ha en spesiell impuls fra operatøren før den utfører instruksjonen.

Den første instruksjon som er skrevet i Fig. 1 definerer et produkt som betegnes med X33 og sier at det framkommer ved å multiplisere to faktorer X35 og X36. Den andre instruksjonen sier at X34 framkommer som summen av den verdi denne variabel tidligere hadde, pluss verdien av X33. Samtidig punches summen på et hullkort. Den tredje instruksjon sier at hver gang den passerer skal det adderes 1 til parameteren N5 og hoppes til den instruksjon som har referansenummer R17, nemlig instruksjon nr. 1. Dette skal gjentas inntil parameteren N5 har fått samme verdi som parameteren N3. Deretter stopper maskinen.

3. GENERELLE INSTRUKSJONER

3.1. Input- og outputinstruksjoner

I instruksene skiller en mellom tre forskjellige typer av data:

- a. variable, Xi, er de størrelser som bearbeides,

b. parametrene, N_i , er hele tall som spesifiserer problemet og bearbeidingsmetoden, men som også selv kan endres under bearbeidningen.

c. konstanter er størrelser som spesifiserer problemet, men er uendret under bearbeidningen. Konstantene kan være både hele tall og brøker.

3.1.1. Input

Data kan leses inn i maskinen på to forskjellige måter i Alfa-systemet. Mest alminnelig er å lese inn datakort. Data som skal punches og leses inn som datakort, settes opp på et dataark som vist i Fig. 2.

Dataark

Felt	Nr.	X_i/N_i	Fast desimalkomma	Flytende desimalkomma	
				A	B
3	1	X35	37.568	+3.7568	+1
	2	X36	+39071	+3.9071	+4
	3	X37	0.056	+5.6	+2
4	1	N1	+87	+8.7	+1
5	1	N7	3	+3.0	0

Figur 2

Det er bare tallene i de to kolonner under flytende desimalkomma som er nødvendig å føre opp og punche. De andre kolonnene kan nyttes når det letter oversikten.

Maskinen leser inn de variable etter følgende instruksjon:

(1) "Read b DATA into X_c onwards"

En minner her om at tekstformuleringen av instruksjonen er helt fri når en ser bort fra funksjonsordet som her er DATA.

Parametrene kan bare leses inn en om gangen etter denne instruksjonen:

(2) "Read l DATA into N_c "

De data som er ført opp på dataarket i figur 2 kan således leses inn i maskinen ved hjelp av følgende instruksjoner:

"Read 3 DATA into X35 onwards

Read 1 DATA into N1

Read 1 DATA into N7"

I instruksjonene ovenfor er tallet på data som skal leses inn bestemt av en konstant. Ofte vil en imidlertid ha behov for å la resultatet av foregående beregninger være avgjørende for hvor mange data en vil lese inn. For de variable kan dette oppnås ved at en gir en parameter, N_b , den ønskede verdi og nytter instruksjonen:

(3) "Read Nb DATA into Xc onwards"

En annen måte å lese data inn i maskinen på er å spesifisere data som konstanter i selve instruksjonen. Størrelsene 0 og 1 kan f.eks. nyttes direkte i B og C kolonnene i de fleste instruksjoner. Videre vil XO og NO alltid tolkes som null og kan brukes uten videre i A, B og C kolonnene. Andre data kan leses inn som konstanter ved hjelp av instruksjonene:

(4) "Xa is the CONSTANT (x).

or

(5) "Na is the CONSTANT c.

I den første instruksjonen må konstantens verdi skrives i en parentes med flytende komma. I tilknytting til disse to instruksjonene har en dessuten muligheten til å bryte ut av instruksjonsrekkefølgen og fortsette med den instruksjon som har det oppgitte referansenummer Rd. Skal den naturlige rekkefølge fortsettes, settes referansenummeret lik R0 som betyr neste instruksjon.

3.1.2. Output

Maskinen kan gi fra seg resultatene ved å punche dem i hullkort som deretter kan skrives ut ved hjelp av en tabulator. Som for input er det også flere måter å få maskinen til å punche ut data på. Det vanligste er å punche ut flere data i et felt (batch). Feltene kan om det ønskes, nummereres fra 1 til 999. Data er dessuten nummerert fra 1 og oppover innen hvert felt. Det punches alltid et kort for hvert datum. En har valget mellom å få data i standardisert form, dvs. slik at det flytende desimalkomma tilfredsstiller de betingelser som er gitt i avsnitt 2, eller i ustandardisert form. De to typer karakteriseres som henholdsvis Type 1 og Type 0. Type 1 må alltid nyttes dersom de punchede data senere skal nyttes som input.

Maskinen puncher ut et bestemt antall variable etter følgende instruksjon:

(6) "Batch a. Punch b RESULTS from Xc onwards. Type d"

På samme måte som for input kan det være ønskelig å la antall variable som punches ut være bestemt av en parameter:

(7) "Batch a. Punch Nb RESULTS from Xc onwards. Type d"

Parametrene kan bare punches ut en om gangen og alltid i standardisert form etter instruksjonen:

(8) "Batch a. Punch 1 RESULT from Nc"

Under avsnitt 2 ble det antydnet en annen måte å få maskinen til å punche ut resultater på. Ved å føye til en P etter regneinstruksjonene, vil resultatet bli punchet i standardisert form og med feltnummer 0. De data

som er gjengitt i figur 2 og som ble lest inn ovenfor, puncher maskinen ut etter følgende instruksjoner:

"Batch 3. Punch 3 RESULTS from X35 onwards. Type 1
Batch 4. Punch 1 RESULT from N1
Batch 5. Punch 1 RESULT from N7"

Når kortene listes på en tabulator, vil de i alminnelighet bli skrevet på samme form som i de to kolonnene lengst til høyre i figur 2.

3.2. Elementære regneinstruksjoner

En har ovenfor sett hvordan arbeidsmateriale kan leses inn i og punches ut av maskinen på forskjellige måter. Her skal en se hvordan maskinen instrueres til å utføre de fire elementære regneoperasjoner.

3.2.1. Addisjon

Instruksjonen for addisjon av variable kan skrives slik:

(9) "Xa is Xb PLUS Xc"

og for parametre:

(10) "Na is Nb PLUS Nc"

Resultatet av operasjonen er her spesifisert i A kolonnen, mens operandene er gitt i B og C kolonnene.

De tre variable og de to parametre i figur 2 kan til eksempel adderes hver for seg, og resultatene betegnes som henholdsvis X38 og N5 ved følgende instruksjoner:

"X38 is X35 PLUS X36

X38 is X38 PLUS X37

N5 is N1 PLUS N7"

3.2.2. Subtraksjon

Instruksjonene for subtraksjon er analoge med addisjonsinstruksjonene og skrives slik:

(11) "Xa is Xb MINUS Xc"

og for parametre:

(12) "Na is Nb MINUS Nc"

3.2.3. Multiplikasjon

Maskinen utfører multiplikasjon av henholdsvis to variable eller to parametre etter følgende instruksjoner:

(13) "Xa is Xb MULTIPLIED by Xc"

og

(14) "Na is Nb MULTIPLIED by Nc"

3.2.4. Divisjon

Instruksjonene for divisjon skrives:

(15) "Xa is Xb DIVIDED by Xc"

og

(16) "Na is Nb DIVIDED by Nc"

For den siste instruksjonen må en huske at Na bare kan være et helt tall og derfor bare vil utgjøre den hele delen av kvosienten. Ellers må både Nb og Nc i tallverdi være mindre enn 16 384 i denne instruksjonen.

3.3. Administrative instruksjoner

I tillegg til input- og outputinstruksjonene og de enkelte regneinstruksjonene, trenger en forskjellige administrative instruksjoner for å få en effektiv organisasjon av instruksene.

3.3.1. Overføring

For å omdefinere eller kopiere variable og parametre, har en 4 overføringsinstruksjoner til disposisjon.

Instruksjonene skrives slik:

(17) "Xa is Xb MOVED.

(18) "Na is Nb MOVED.

(19) "Xa is Nb MOVED.

(20) "Na is Xb MOVED.

Betegnelsen i A-kolonnen er den nye betegnelsen på den størrelse som er spesifisert i B-kolonnen. Den gamle betegnelsen beholdes fortsatt. Dersom en overførings- eller regneinstruksjon gir en ny størrelse en betegnelse som allerede er brukt, er det den siste definisjon som er gyldig.

I forbindelse med instruksjon (20) må en igjen huske at Na bare kan være et helt tall. En overføring etter denne instruksjon gir Na en verdi som svarer til hele tall som er nærmest Xb.

3.3.2. Ubetinget hopp

Tidligere har en sett at i tilknytting til innlesning av konstanter og ved overføringer, kan instruksjonsfølgen brytes ved et ubetinget hopp. En slik mulighet er også knyttet til visse andre instruksjoner som en skal komme inn på senere. Dessuten finnes det 3 spesielle ubetingede hoppinstruksjoner, nemlig:

- (21) "JUMP to the reference Rd"
- (22) "JUMP to the subroutine Sd"
- (23) "JUMP to the reference Nd"

Maskinen hopper etter den første instruksjonen direkte til den instruksjon som har referansenummer Rd. I det andre tilfellet hopper den til den subrutine (se 5) som har subrutinennummer Sd. Ved den siste instruksjonen hopper maskinen til den instruksjon som har det referansenummer som parameteren Nd angir.

3.3.3. Stopp m.m.

Maskinen vil stoppe etter følgende instruksjon:

- (24) "STOP"

Denne instruksjon må en bl.a. alltid nytte mellom hovedinstruksjonen og etterfølgende subrutiner. Maskinen vil også stoppe midt i instruksjonen dersom det skrives en S eller en O etter instruksjonen. Maskinen stopper da før den utfører instruksjonen, men kan settes i gang av operatøren fra kontrollbordet. Kombinert med P-utpunching er dette en verdifull mulighet for testing av instruksjonen.

En kan også få maskinen til å gi et lydsignal når den stopper, og gjøre et ubetinget hopp når den settes i gang ved å nytte følgende instruksjon:

- (25) "BUZZ, stop and jump to Rd"

Til slutt er det en instruksjon som alltid må utgjøre siste instruksjon i enhver instruks. Det er:

- (26) "FINISH"

Denne instruksjon må komme etter subrutiner, parametertabeller og andre ting som måtte følge etter selve hovedinstruksjonen.

3.4. Telleinstruksjoner

Når en beregning eller del av en instruks skal gjentas flere ganger, kan delinstruksjonen skrives tilsvarende mange ganger. Men som i en vanlig instruks kan en også i Alfasetmet be maskinen gjenta et utsnitt av instruksjonen et bestemt antall ganger før den fortsetter.

Dette gjør en ved å gi den første instruksjonen i instruksjonen som skal gjentas, et hvilket som helst ledig referansenummer mellom 1 og 99, og etter den siste instruksjonen skriver en telleinstruksjonen:

- (27) "Count Nb UP TO c. Jumping to Rd"

hvor c er det antall ganger instruksjonen skal gjentas. Dette tallet må være mindre enn 4 096. Parameteren Nb settes automatisk lik 0 ved start og

øker altså til c med 1 for hver gjennomgang av instruksdelen, hvorefter parameteren automatisk settes lik null og maskinen går videre til neste instruksjon.

Spesielt når instruksen skal nyttes for flere oppgaver, er det ønskelig å innrette seg slik at tallet på gjennomganger er bestemt av en parameter. En nytter da istedet instruksjonen:

(28) "Count Nb UP TO Nc. Jumping to Rd"

Parameteren Nc kan da enten være lest inn fra et parameterkort eller være resultatet av foregående beregninger.

3.5. Modifiseringsinstruksjon

Det er sjelden at en ønsker å gjenta den samme instruksjonsdel med de samme data for hver gjennomgang, som oftest ønsker en å modifisere bestemte spesifikasjoner i instruksdelen hver gang den gjentas. I Alfa-systemet finnes en spesiell instruksjon for dette formål. Skrives instruksjonen

(29) "Na Nb MODIFY Nc"

vil neste instruksjon tolkes som om parameterverdiene i modifiseringsinstruksjonen var addert til variabel- eller parameterspesifikasjonene i de respektive kolonner.

Følgende instruksjoner vil for eksempel få maskinen til å beregne produktsummen, X4, med leddene (X6+i).(X16+i) for (i=0.....9):

```
" 12          N3 MODIFY          N3
      X5 is    X6 MULTIPLIED by X16
      X4 is    X4 PLUS             X5
      Count N3 UP TO             10. Jumping to R12"
```

3.6. Sammenlikningsinstruksjoner

Foruten de ubetingede hopp har en også muligheter for å la maskinen utføre betingede hopp i instruksjonen. Kriteriene for hopp er i denne sammenheng resultatet av sammenlikning av bestemte størrelser. I alt finnes det åtte sammenlikningsinstruksjoner, fire for sammenlikninger mellom variable og fire for parametre. Felles for alle er at maskinen fortsetter til etterfølgende instruksjon dersom betingelsen for hopp ikke er oppfylt..

(30) "If Xb EQUALS Xc Jump to Rd"

(31) "If Nb EQUALS Nc Jump to Rd"

(32) "If Xb is UNEQUAL to Xc Jump to Rd"

(forts.)

- (33) "If Nb is UNEQUAL to Nc Jump to Rd"
- (34) "If Xb is AS BIG as Xc Jump to Rd"
- (35) "If Nb is AS BIG as Nc Jump to Rd"
- (36) "If Xb is BIGGER than Xc Jump to Rd"
- (37) "If Nb is BIGGER than Nc Jump to Rd"

I (34) og (35) betyr "AS BIG as" større eller lik.

En ønsker for eksempel å finne hvor mange, N1, av de variable X1....Xn som må kummuleres for at summen minst skal bli like stor som verdien av X2001.

Følgende instruksjoner får maskinen til å kumulere de variable, undersøke etter hver addisjon om summen er blitt lik eller større enn X2001. Maskinen hopper i så fall til puncheinstruksjonen og puncher ut tallet på addender. Hvis ikke, går den gjennom telleinstruksjonen inn i løkken igjen. Dersom summen ikke når opp i verdien av X2001 etter at alle addender er addert, vil N1 i siste utførelse av tellinstruksjonen bli satt lik null som deretter punches ut.

```
      " 2                N1          MODIFY
          X2000 is XL          PLUS          X2000
              If X2000 is AS BIG as X2001 Jump to R1
              Count N1          UP TO          n          Jumping to R2
1 Batch a Punch 1          RESULTS from N1 "
```

4. SPESIELLE INSTRUKSJONER

I tillegg til de mer generelle instruksjoner som er behandlet foran, omfatter Alfssystemet en rekke spesielle instruksjoner som forenkler instrukskrivning av forskjellige problemer.

4.1.1. Overføring av største verdi

Følgende instruksjoner får maskinen til å kopiere den av to størrelser som har den største verdi:

(38) "Xa is Xb MAXIMUM Xc"

og

(39) "Na is Nb MAXIMUM Nc"

4.1.2. Rotutdraging

Rotutdraging foretas etter instruksjonen:

(40) "Xa is the ROOT of Xc"

4.1.3. Modulen

Modulen $X_a = \sqrt{X_b^2 + X_c^2}$ beregnes ved:

(41) "Xa is Xb MODULUS Xc"

4.1.4. Logaritmer og eksponentialfunksjon

(42) "Xa is LOG of Xc"

(43) "Xa is EXP of Xc"

begge med gruntall e.

4.1.5. Trigonometriske o.a. funksjoner

(44) "Xa is SIN Xc"

(45) "Xa is COS Xc"

(46) "Xa is TAN Xc"

(47) "Xa is SIN-1 Xc"

(48) "Xa is COS-1 Xc"

(49) "Xa is TAN-1 Xc"

(50) "Xa is SINH Xc"

(51) "Xa is COSH Xc"

(52) "Xa is TANH Xc"

(53) "Xa is SINH-1 Xc"

(54) "Xa is COSH-1 Xc"

(55) "Xa is TANH-1 Xc"

4.1.6. Rekkesum

Summen av rekken $\sum_0^{b-1} p_i \cdot (Xc)^i$ hvor $p_0 \dots p_{b-1}$ er definert som $X(c+1) \dots X(c+b)$, beregner maskinen etter instruksjonen:

(56) "Xa is sum b term SERIES. Argument Xc"

4.1.7. Grafisk avlesning og tabelloppslag

Funksjoner som skal nyttes under beregningene kan leses inn i tabellform. En leser da inn som variable $X(c+1) \dots X(c+b)$ det ønskede antall argumenter, som ikke nødvendigvis må være ekvidistante, i stigende rekkefølge. Deretter leses de tilhørende funksjonsverdier.

Når en skal ha tak i en bestemt funksjonsverdi svarende til X_c , nytter en instruksjonen:

(57) "Read Xa from b pt. GRAPH at Xc"

4.1.8. Integrasjon

Dersom en funksjon er lest inn i maskinen med b funksjonsverdier svarende til ekvidistante argumentverdier, beregnes dens integral ved Simpsons regel etter følgende instruksjon:

(58) "Xa is the b step INTEGRAL with interval Xc"
hvor X_c er intervallet mellom to argumenter.

4.1.9. Simultane differensiallikninger

For b simultane differensiallikninger av formen $\frac{dy_i}{dx} f_i(x, y_1 \dots y_b)$ med gitte x^0 og y_i^0 , beregnes deres nye verdier for $x^0 + h$ ved først å lage en subrutine S_d som beregner alle f_i for de gitte verdier, og deretter nyttes instruksjonen:

(59) "Solve b DIFF.EQUATIONS at X_c intervals, using S_d "

hvor

$X_c = x^0$

$X(c+1+i) = y_i^0$

$X(c+b+1+i) = f_i$

$X(c+2b+2)$ til

$X(c+3b+1)$ nyttes til mellomresultater.

Etter beregningen er x^0 og alle y_i^0 erstattet av de nye verdier. Beregningen foregår etter Runge-Kuttas 4-trinns metode.

5. SUBRUTINER

I mange instruksjoner vil en ofte komme tilbake til samme beregningsrutine. Det er da hensiktsmessig å ordne seg ved hjelp av såkalte subrutiner for vedkommende beregning. I hver instruksjon kan en ha opptil 27 slike subrutiner. En subrutine kjennetegnes ved at den første og den siste instruksjon i rutinene er henholdsvis:

(60) "Sr SUBROUTINE"

og

(61) " END OF subroutine S_d "

I den første instruksjon må rutinens nr. skrives på formen S_r i instruksjonens R-kolonne. Dette referansenummer må også avslutte siste instruksjon i D-kolonnen. Ved bruk av subrutiner må en være spesielt merksam på at en ikke kan bruke referansenummer som er brukt i hovedinstruksjonen eller i andre subrutiner.

Når en i hovedinstruksjonen kommer til et punkt hvor en ønsker at maskinen skal hoppe inn i en subrutine, skrives en vanlig ubetinget hoppinstruksjon som vist i (22). Når maskinen har fullført subrutinen, hopper den automatisk inn i hovedinstruksjonen til instruksjonen som følger etter hoppinstruksjonen. En instruksjon kan ha flere ordener med subrutiner, dvs. subrutinene kan selv ha subrutiner.

Subrutinene skrives etter hovedinstruksjonen og skilles fra denne med en stoppinstruksjon (24).

6. REGLER FOR PUNCHING

6.1. Punching av instruksjoner¹⁾

Det punches et kort for hver instruksjon. Instruksjonen punches i kol. 1-32, Alfainstruksens nummer i kol. 66-72, eventuelt serienummer i kol. 73-76 og instruksjonens nummer i 77-80. Dette avviker fra det system Alfa-kortene er redigert etter i England, men svarer på den annen side til de vanlige programkort som vil bli brukt av Statistisk Sentralbyrå.

Instruksjonens enkelte deler punches slik:

kol.	1- 2	r
"	3- 5	R
"	6-10	A
"	11-15	B
"	16-21	FUNCTION
"	22-26	C
"	27-29	D
"	30	P
"	31	S/O (O er bokstav, ikke tall)
"	32	Y-raden skal alltid punches.

D a t a f e l t e r. Datafeltene A, B og C må alltid punches i alle kolonner. Variable, parametre og konstanter markeres med henholdsvis X, N og O i kol. 6/11/22. Dersom feltene ikke brukes i instruksjonen, f.eks. i instruksjonen STOP, må de fylles ut med nuller.

R e f e r a n s e f e l t e r. I R-feltet punches et vanlig referansenummer i kol. 4-5 med O i kol. 3. Når referansen gjelder en subrutine, punches dessuten en S i kol. 3. Referansenummer i D-feltet må alltid begynne med R, S eller N i kol. 27. En unntakelse er i "RESULTS" hvor det punches O i kol. 27 foran typenummer. Dersom det ikke forekommer referansenummer i instruksjonen, må referansefeltene være blanke.

F u n k s j o n s f e l t. Dette felt kan være blankt. Dersom en imidlertid finner det ønskelig også å punche inn funksjonsordet, må det

1) Da de norske punchmaskiner er modifisert for vårt alfabet, må en inntil videre foreta omkodning av enkelte bokstaver før de punches. Lar en FUNCTION-feltet være blankt, vil det bare være nødvendig med følgende omkodninger:

N	punches	som	P
O	"	"	Q
P	"	"	R
R	"	"	T
S	"	"	U
X	"	"	Å

være i overensstemmelse med koden i r-feltet. Punching av funksjonsordet begynner i kol. 16 og omfatter de første seks bokstaver inklusive mellomrom. Eventuelle ledige kolonner etter funksjonsordet skal være blanke.

A n d r e f e l t e r. I kol. 30 punches bokstaven P dersom resultatet av instruksjonen skal punches. I kol 31 punches eventuelt bokstaven O eller S dersom instruksjonen krever det. Når p, O eller S ikke forekommer i instruksjonen, skal kol. 30 og 31 være blanke. Y-raden i kol. 32 må alltid punches for at maskinen skal kunne skille mellom instruksjoner og konstanter.

6.2. Punching av data

Både variable, parametre og konstanter som skal leses inn fra datakort, må punches med flytende desimalkomma på standardisert form (3.1.1) og i følgende kolonner:

A kol. 7-15

B " 24-26

F o r t e g n. Fortegn er markert med overpunching i kol. 7 og 24 i Y- eller X-raden for henholdsvis + og -.

F e l t- o g k o r t n u m m e r. Feltnummer kan punches i kol. 27-29, mens kortnummeret punches i kol. 30-32.

Konstanter som skal leses inn i forbindelse med instruksjonen:

"Xa is the CONSTANT (x)",

punches etter reglene ovenfor og kortet legges inn umiddelbart etter instruksjonene som definerer konstantene. Merk at N-konstanter punches i instruksjonens C-felt.

7. ALFAPROGRAMMET OG BETJENING AV DEUCE

Alfaprogrammet sørger for at DEUCE omsetter de desimale instruksjoner og kontrollerer at de blir utført. Alfaprogrammet består av to deler, samleprogrammet og det interpretive program.

7.1. Samleprogrammet

7.1.1. Samleprogrammet er punchet på 123 programkort. Det legges foran de desimalt punched instruksjonskort i kortmagasinet og gjør at de sistnevnte blir lest inn, kontrollert, omformet til binære kodeord som det interpretive program krever og til slutt at de binære kodeord blir punchet ut. Maskinen settes i gang ved bryteren INITIAL INPUT på 528.

7.1.2. Hvert binært kodeord som svarer til en instruksjon legger beslag på et ordpar, en like og en ulike m.c., i DEUCE. Ordparet er disponert slik:

	P1-P12	P13-P24	P25-P30	P31	P32
Like m.c.	A	C	r	P	O
Ulike m.c.	B	D	Markering	F	S

F er her en spesiell markering som viser at r-operasjonen gjelder en parameter og ikke en variabel. Markeringen settes automatisk.

Kodeordene lagres på trommelen i sporene 0/0 til 1/15 og får adresser fra 1-511 idet første ordpar alltid er null. Samleprogrammet reserverer dessuten et ordpar for hver parameter og X-konstant som blir definert i instruksen. Dersom adressen til det kodeord som første gang definerer parameteren eller konstanten er "n", vil parameterens adresse være "n+1".

Samleprogrammet koder om følgende r-verdier:

Opprinnelig r:	Instruksjon	Ny r:
0	Xa is Nb MOVED	14
0	Na is Xb MOVED	17
4	Na is Nb DIVIDED by Nc	16
5	JUMP to Nd	18
5	JUMP to Sd	19
10	Count Nb UP TO Nc Rd	15
11	Xa is the CONSTANT (X)	0

Etter at maskinen har lest inn og omformet de desimale instruksjoner til binære kodeord, blir den binære utgave av programmet automatisk punched ut i tilfelle en skal nytte programmet flere ganger. De ni første kort som punches ut, inneholder en referanse katalog. Det første kodeord punches ut på Y- og X-radene av det tiende kortet. Kodeordene punches i kol. 1-32 og er disponert som oppgitt ovenfor.

Konstanten 1 i B- eller C-feltet representeres binært ved 12 enere i alle instruksjoner unntatt "UP TO", "CONSTANT", "DATA" og "RESULTS" hvor den representeres på vanlig måte.

"SUBROUTINE" framkommer som et blankt ordpar, mens "STOP" og "FINISH" blir representert ved enere i alle ordparets posisjoner. Ved utpunching blir imidlertid "FINISH" punched ut ved enere bare i første ord.

7.1.3. Samleprogrammet er laget slik at det kontrollerer instruksjonene og indikerer visse typer feil:

1. P u n c h e f e i l. Maskinen stopper med 4,8-24 X i TS COUNT. OS-lamoene indikerer i hvilken kolonne feilen forekommer. Når flere lamper i serie lyser, er det den lengst til venstre som indikerer feilkolonnen. Lyser alle OS-lampene er det lest inn en konstant i stedet for en instruksjon eller omvendt.

Etter en feil av denne type må det sist innleste kort rettes og leses inn på nytt.

2. L o g i s k e f e i l. Maskinen vil lese alle kortene, sette alarmen på og punche ut kort som indikerer feilene. Disse feilkortene har r-nummer punchet i kol. 1-2 og instruksjonsnummer i kol. 30-32. Kolonner med feil indikeres med punching i feilkortets x-rad. Etter at alle feilkortene er punchet ut, vil maskinen stoppe på 2,6-24.

7.2. Det interpretive program

7.2.1. Det interpretive program består av 223 kort. Det leses inn og lagres i sporene 11/5 til 15/15 på trommelen og omsetter de binære kodeord til DEUCE-operasjoner.

Har samleprogrammet lest inn instruksjonskortene på forhånd, vil de tilsvarende kodeord fortsatt være lagret på trommelen. Det interpretive programmet med alle dets 223 kort legges deretter i innlesningsmagasinet foran datakortene. Innlesning og beregninger settes i gang ved bryteren READ på kontrollbordet.

Foreligger de binære kodeord på hullkort fra tidligere bruk av samleprogrammet, legges hele det interpretive program i kortmagasinet etterfulgt av kortene med de binære kodeord og til slutt datakortene. DEUCE settes i gang ved bryteren INITIAL INPUT på 528.

De variable blir lagret i sporene 2/0 til 11/4 på trommelen. Hver variabel med adresse opptar et ordpar som disponeres slik:

Like m.c.	P 1- 4: Ordparets nr. i sporet
	P 5- 8: Sporets nr. i blokken
	P 9-12: Blokkens nr.
	P13-32: Tallets eksponent
Ulike m.c.	P 1-32: Tallets mantisse.

Parametrene har som tidligere nevnt (9.1.2) adresser mellom kodeordene i sporene 0/0 til 1/15. Første halvpart av et parameterordpar vil alltid være null, og det vil alltid være en P1 i den ulike m.c.

Under beregningen holdes alltid to spor med instruksjoner i DL 9-10 og to spor med variable i DL 11-12. Rutiner for alle $r < 16$ holdes vanligvis i DL 1-8. Dersom $r \geq 16$, må tilsvarende rutine hentes fra trommelen og etter bruk erstattes med de vanlige rutiner som er visket ut.

7.2.2. Under beregningen vil DEUCE stoppe med 1,2-24 x i TS COUNT og første halvpart av kodeordet på OS dersom:

1. instruksjonen er en ubetinget stoppinstruksjon, dvs. når S er punchet i kol. 31
2. instruksjonen er en betinget stoppinstruksjon, dvs. når det er punchet 0 i kol. 31, og det er satt inn en P31 på ID
3. det er satt inn en P32 på ID. Maskinen stopper da foran hver instruksjon.

Settes TIL "On" og det gis et Single-Shot, vil innholdet av ID erstatte første halvdel av kodeordet og kan avleses på OS når DEUCE deretter stopper på 1,2-24 x.

Slås TIL "Off", vil DEUCE etter første Single-Shot stoppe på 7,2-24 x med annen halvpart av kodeordet på OS. Slås deretter TIL "On", endres annen halvpart på samme måte som beskrevet for første halvpart.

Et Single-Shot med TIL "Off" får maskinen til å utføre instruksjonen og fortsette beregningene.

7.2.3. DEUCE puncher ut resultatene i standardisert eller ustandardisert form (3.1.2). Dersom puncheinstruksjonen har en 1 i D-feltet (Type 1), får en standardisert output, som svarer til de kravene en stiller for datainput (8.2). Standardisert output er både den mest oversiktlige form og den som er nødvendig dersom materialet senere skal nyttes i DEUCE.

Den ustandardisert form, dvs. resultater med flytende komma, men uten restriksjonen $1 \leq |A| < 10$ på mantissen får en ved å nytte bokstaven 0 i D-feltet (Type 0). De ni utsagnskraftige siffer punches da i feltet kol. 1-23 med komma mellom kol. 7 og 8 og fortegn som overpunching i kol. 7. Dersom tallet er så lite eller så stort at ikke alle de utsagnskraftige siffer kommer med i feltet, skiftes tallet henholdsvis til venstre eller til høyre inntil alle siffer er kommet inn i feltet. Tallet på skift punches da i kol. 24-26 med fortegn i kol. 24.

Den ustandardiserte form for output vil vanligvis være noe raskere enn den standardiserte.^{x)}

x) P-punching oppnås ved å sette P30 på ID.

7.2.4. På samme måte som samleprogrammet, vil også det interpretive program stoppe DEUCE og indikere visse feil etter følgende tabell:

TS COUNT	FUNCTION	Feil
7, 1-1 x	GRAPH	$b > 1$ (b er antall punkter)
7, 2-2 x	"	X_c er mindre enn minste arg.
7, 3-3 x	"	X_c er større enn største arg.
0,16-0 x	COSH-1	$X_c < 1$
0,18-14 x	END OF SUBR.	Subrutinen er endret på annen måte enn ved "JUMP to Sd" eller "DIFF. EQUA."
0,21-21 x	SERIES	$b = 0$ (b er antall ledd)
0,30-21 x	Na is Xb MOVED	$ X_b \geq 2.147.483.648 = 2^{31}$
0,31-21 x	ROOT	X_c negative
"	MODULUS	Aritmetisk feil
"	SINH-1	"
"	COSH-1	"
1, 1-1 x	Nb DIVIDED by Nc	$ N_b $ eller $ N_c \geq 16384 = 2^{14}$
1, 2-2 x	"	$N_c = \text{null}$
1,30-21 x	EXP	$ X_c \geq 4096 = 2^{12}$
"	SINH	"
"	COSH	"
"	TANH	"
1,30-15 x	SIN-1	$ X_c > 1$
"	COS-1	"
2,30-21 x	LOG	X_c er negative
"	SINH-1	Aritmetisk feil
"	COSH-1	"
3,14-28	Xb DIVIDED by Xc	$X_c = \text{null}$
"	TAN	Tangens X_c uendelig
"	TANH	Aritmetisk feil
"	GRAPH	"
3,12-24 x	DATA	Et kort er galt punchet
0,16-0 x	INTEGRAL	$b < 1$ (b er antall intervall)
7,30-21 x	COS	$ X_c \geq 536.870.912 = 2^{29}$
"	SIN	"
"	TAN	"

Etter 3,14-28 vil DEUCE fortsette i programmet når en først stopper den, setter DISCR. "On", gir et Single-Shot, slår DISCR. og STOP tilbake til normalstilling og gir et nytt Single-Shot.

Etter alle andre feil vil DEUCE fortsette programmet etter et Single-Shot

8. KAPASITETSBEGRENSNINGER

8.1. Tallområder

$X_i = A \cdot 10^B$, hvor $1 \leq A < 10$ og $0 \leq B \leq 999$.

$N_i = A \cdot 10^B$ må dessuten være $N_i \leq 2 \cdot 10^9$

0 er definert som $\pm 0 \cdot 10^{\pm 0}$

8.2. Nøyaktighet

A har 9 siffer

8.3. Tallet på variable og parametre

X_i er variable, hvor $1 \leq i \leq 2383$

N_i er parametre, hvor $1 \leq i \leq 63$

T_i er hjelpevariabel, hvor $1 \leq i \leq 8$

8.4. Tallet på instruksjoner og parametre

I_i er instruksjoner og

N_j er parametre, hvor $i+j \leq 511$

8.5 Konstanter

X_0 og N_0 er alltid null. Kan brukes i alle kolonner.

0,1 og i kan dessuten leses direkte inn i B- og C-kolonner

8.6. Gjennomganger av en indeksdel

Med telleinstruksjonen kan en indeksdel gjennomgås opp til 4 095 ganger.

8.7. Område for feltnummer, referansenummer og subrutinenummer

Feltnummer = a, hvor $0 \leq a \leq 999$

Referansenummer = R_i , hvor $1 \leq i \leq 99$, (R_0 = neste instruksjon)

Subrutinenummer = S_i , hvor $1 \leq i \leq 27$

9. OVERSIKT OVER INSTRUKSJONENE

Nr.	r	R	Tekst	A	Tekst	B	Tekst	FUNCTION	Tekst	C	Tekst	D	
1			<u>Input- og Outputinstruksjoner</u>										
1	23				Read	b		DATA	into	Xc	onwards		
2	23				Read	l		DATA	into	Nc			
3	23				Read	Nb		DATA	into	Xc	onwards		
4	11			Xa	is the			CONSTANT	(X)				
5	11			Na	is the			CONSTANT		c			
6	24		Batch	a	Punch	b		RESULTS	from	Xc	onwards. Type	d	
7	24		Batch	a	Punch	Nb		RESULTS	from	Xc	onwards. Type	d	
8	24		Batch	a	Punch	l		RESULTS	from	Nc			
			<u>Elementære regneinstruksjoner</u>										
9	01			Xa	is	Xb		PLUS		Xc			
10	01			Na	is	Nb		PLUS		Nc			
11	02			Xa	is	Xb		MINUS		Xc			
12	02			Na	is	Nb		MINUS		Nc			
13	03			Xa	is	Xb		MULTIPLIED	by	Xc			
14	03			Na	is	Nb		MULTIPLIED	by	Nc			
15	04			Xa	is	Xb		DIVIDED	by	Xc			
16	04			Na	is	Nb		DIVIDED	by	Nc			
			<u>Administrative instruksjoner</u>										
17	00			Xa	is	Xb		MOVED					
18	00			Na	is	Nb		MOVED					
19	00			Xa	is	Nb		MOVED					
20	00			Na	is	Xb		MOVED					
21	05							JUMP	to			Rd	
22	05							JUMP	to		the subroutine	Sd	
23	05							JUMP	to		the reference	Nd	
24	14							STOP					
25	21							BUZZ	stop and		Jump to	Rd	
26	18							FINISH					
			<u>Telleinstruksjoner</u>										
27	10				Count	Nb		UP TO		c	Jumping to	Rd	
28	10				Count	Nb		UP TO		Nc	Jumping to	Rd	

Nr.	r	R	Tekst	A	Tekst	B	Tekst	FUNCTION	Tekst	C	Tekst	D
<u>Modifiseringsinstruksjon</u>												
29	12			Na		Nb		MODIFY		Nc		
<u>Sammenlikningsinstruksjoner</u>												
30	06				If	Xb		EQUALS		Xc	Jump to	Rd
31	06				If	Nb		EQUALS		Nc	Jump to	Rd
32	07				If	Xb	is	UNEQUAL	to	Xc	Jump to	Rd
33	07				If	Nb	is	UNEQUAL	to	Nc	Jump to	Rd
34	08				If	Xb	is	AS BIG	as	Xc	Jump to	Rd
35	08				If	Nb	is	AS BIG	as	Nc	Jump to	Rd
36	09				If	Xb	is	BIGGER	than	Xc	Jump to	Rd
37	09				If	Nb	is	BIGGER	than	Nc	Jump to	Rd
<u>Overføring av største verdi</u>												
38	13			Xa	is	Xb		MAXIMUM		Xc		
39	13			Na	is	Nb		MAXIMUM		Nc		
<u>Rotutdraging</u>												
40	25			Xa	is the			ROOT	of	Xc		
<u>Modulen</u>												
41	26			Xa	is	Xb		MODULUS		Xc		
<u>Logaritmer og eksponentialfunksjon</u>												
42	28			Xa	is			LOG	of	Xc		
43	29			Xa	is			EXP	of	Xc		
<u>Trigonometriske o.a. funksjoner</u>												
44	30			Xa	is			SIN		Xc		
45	31			Xa	is			COS		Xc		
46	32			Xa	is			TAN		Xc		
47	34			Xa	is			SIN-1		Xc		
48	35			Xa	is			COS-1		Xc		
49	36			Xa	is			TAN-1		Xc		
50	37			Xa	is			SINH		Xc		
51	38			Xa	is			COSH		Xc		

Nr.	r	R	Tekst A	Tekst B	Tekst FUNCTION	Tekst	C	Tekst	D
<u>Trigonometriske o.a. funksjoner (forts.)</u>									
52	39		Xa is		TANH		Xc		
53	40		Xa is		SINH-1		Xc		
54	41		Xa is		COSH-1		Xc		
55	42		Xa is		TANH-1		Xc		
<u>Rekkesum</u>									
56	27		Xa is sum	b term	SERIES	Argument	Xc		
<u>Grafisk avlesning og tabelloppslag</u>									
57	44		Read Xa from	b pt	GRAPH	at	Xc		
<u>Integrasjon</u>									
58	43		Xa is the	b step	INTEGRAL	with interval	Xc		
<u>Simultane differensiallikninger</u>									
59	33		Solve	b	DIFF.EQUA	at	Xc	intervals, using	Sd
<u>Subrutiner</u>									
60	Sr				SUBROUTINE				
61					END OF	Subroutine			Sd

