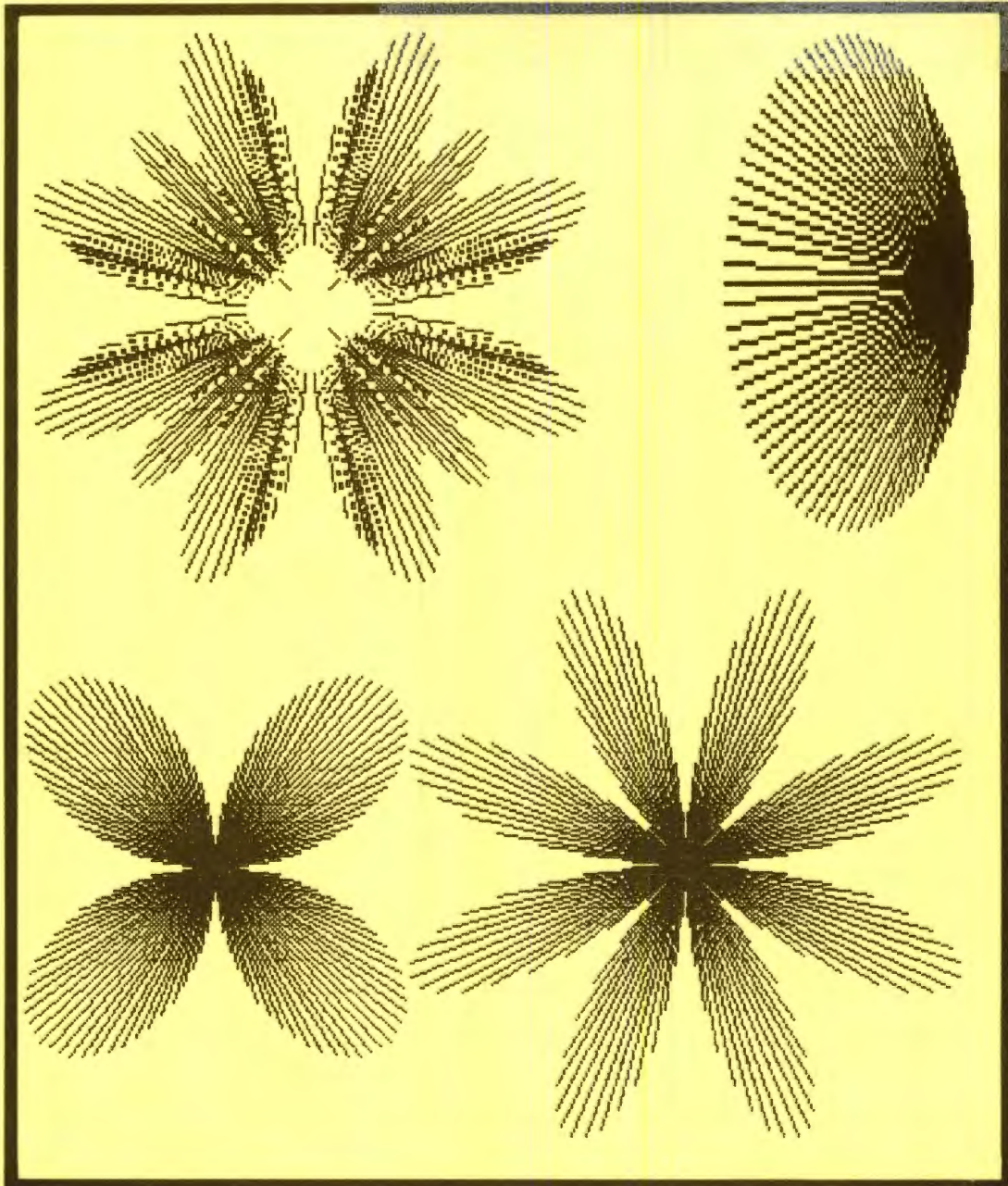


Ole M.

HØJOPLØSNINGSGRAFIKMODUL
MPS-24

BRUGERVEJLEDNING - JUNI 1984



ICL

I N D H O L D S F O R T E G N E L S E

1.	INDHOLD	1
2.	FUNKTION	1
3.	OUT og INP	2
	3.1 OUT	2
	3.2 INP	3
4.	PORTE / REGISTRE	3
5.	SPECIELLE PORTE	4
	5.1 VALG MELLEM VIDEO og GRAFIK	4
	5.2 READ-MODIFY-WRITE (RMW)	4
	5.3 LÆS REGISTER	5
	5.4 VALG AF GRAFIKBILLEDE	5
6.	CTRL1 REGISTERET	6
	6.1 PEN/SLETTE - OPPE/NEDE	6
	6.2 FLAD/CYKLISK SKÆRM	6
	6.3 DATA I CTRL1	6
7.	KOMMANDOER	7
8.	GP'ens STATUS	8
9.	TEGNING AF RETTE LINIER	9
10.	CTRL2 REGISTERET	11
11.	KARAKTERER	12
12.	SMAVEKTORER	14
13.	AFLÆSNING AF SKÆRM	14

Appendix 1: Detaljeret beskrivelse af grafikprocessor.

Appendix 2: Programeksempler.

ICL A/S tilstræber, at informationerne i denne vejledning til enhver tid er korrekte og ajourførte. Imidlertid påtager ICL A/S sig intet ansvar på grund af eventuelle unøjagtigheder eller mangler i informationerne i denne vejledning. På den anden side vil vi være taknemmelige for enhver kritik af form og indhold, som kan være med til at forbedre vejledningen.

Lyngby, juni 1984

1. INDLEDNING

=====

Denne beskrivelse af COMET'ens grafikmodul MPS-24 er baseret på programmeringssproget COMAL-80. Dette er gjort for at gøre eksempler lettere at forstå og efterprøve. Det er enkelt at overføre beskrivelsen til andre sprog såsom PASCAL og assembler.

Sammen med et MPS-24 modul leverer ICL A/S et sæt COMAL-80 procedurer COMGRAPH, som gør udvikling af COMAL-80 programmer meget enkel, og som ikke kræver nogen dyb indsigt i grafikmodulets detaljer.

Denne beskrivelse er opbygget således at der i første del gives en beskrivelse af programmeringen til MPS-24, medens der for yderligere detaljer henvises til appendix 1.

2. FUNKTION

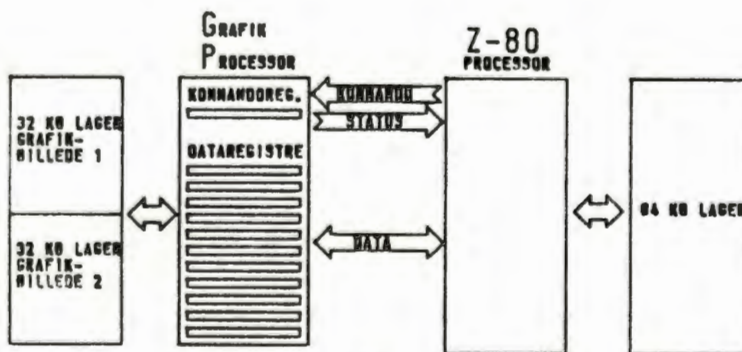
=====

Et MPS-24 modul består hovedsageligt af en grafikprocessor - i det følgende kaldet GP - samt et 64 kB stort arbejdslager.

COMET'ens sædvanlige processor, som jo ligeledes arbejder med et 64 kB stort lager, vil vi i det følgende benævne Z-80.

GP'en består af et antal REGISTRE, hvoraf et er et KOMMANDO-REGISTER, medens øvrige registre (11 i alt) er DATAREGISTRE.

Udførelsen af grafik på en COMET foregår på den måde - jvfr. figur 1. - at der fra Z-80 skrives data til GP'ens dataregistre samt kommandoer til kommandoregisteret. Når GP'en modtager en kommando i kommandoregisteret, vil denne kommando (som f.eks. kan være at tegne en ret linie af en given længde fra et givet punkt i en given retning) blive udført. Herunder anvendes data, der forud er skrevet i GP'ens dataregistre.



FIGUR 1.

Kommandoer til GP'en vedrører enten dataregistrene eller tegning af grafik. Dette sidste sker ved at, GP'en i det tilhørende hovedlager sætter bits ON og OFF. Hovedlagerets indhold kan derefter gengives på dataskærmen, således at bits, der er OFF, giver et lysende punkt, medens bits, der er ON, giver et mørkt punkt.

Et grafikbillede på skærmen består af 512 x 512 punkter svarende til 32 kB. GP'ens hovedlager har derfor kapacitet til at rumme to grafikbilleder samtidig. Grafikbilleder kan vises på samme skærm, som også anvendes af Z-80 til alfanumeriske data, men det er også muligt at have tilsluttet to skærme, hvoraf en anvendes til Z-80 ind-/uddata, medens den anden udelukkende viser grafik.

Ud over at kunne skrive data/kommandoer TIL GP'ens registre, kan Z-80 også læse data FRA registrene i GP'en. Kommunikationen mellem GP og Z-80 sker via PORTE, idet der til hvert register i GP er knyttet en port. Z-80'erenes kontrol af GP'en består derfor i skrivning til hhv. læsning fra porte.

3. OUT og INP

=====

I COMAL-80 foregår kommunikation via porte v.hj. af sætningen OUT (skrivning til port) og funktionen INP (læsning fra port).

3.1. OUT

COMAL-80 sætningen OUT har formatet:

OUT portnummer, decimalværdi

hvor såvel portnummer som decimalværdi kan være en variabel, et aritmetisk udtryk eller en konstant. Funktionen af OUT er den, at 'decimalværdi' skrives til porten 'portnummer'.

EKSEMPEL:

```
OUT 176, 0 // værdien 0 skrives til port 176
OUT 160, 7 // værdien 7 skrives til port 160
OUT XREG+1, XKOORD // indholdet af variabelen XKOORD
                  // skrives til den port, hvis
                  // nummer svarer til resultatet af
                  // udtrykket XREG+1.
```

Der er mulighed for at skrive til i alt 256 porte, som er nummereret fra 0 til 255. Det der skrives til en port er et heltal indenfor det talinterval, der svarer til en byte, altså 0 til 255. I forbindelse med GP anvendes portnumrene fra og med 160 til og med 179. Enkelte af de mellemliggende portnumre er dog ikke anvendt i dag, men reserveret til senere brug.

3.2. INP

Indlæsning fra en port sker ved funktionen:

INP(portnummer)

hvor portnummer kan være en variabel, et aritmetisk udtryk eller en konstant, der opfylder samme krav, som nævnt under OUT.

EKSEMPEL:

```

DELTA:=INP(165) // variabelen DELTA tildes
                // værdien indlæst fra port
                // nummer 165.
PRINT INP(160) // værdien indlæst fra port
                // nummer 160 udprintes.
    
```

4. PORTE / REGISTRE

En oversigt over GP'ens registre og de tilhørende portnumre er vist i tabel 1. En detaljeret beskrivelse af hver enkelt port findes i appendix 1. I det følgende gives en beskrivelse af registrene i relation til GP'ens enkelte funktioner.

PORT NR.	REGISTER NAVN	VÆRDI INTERVAL	ANVENDELSE
160	CMD	<0..255>	KOMMANDOREGISTER. Modtager kommandoer fra Z-80.
160	STATUS	<0..255>	Giver ved læsning status for GP, fx. om GP er klar til ny kommando.
161	CTRL1	<0..127>	Indholdet af dette reg.viser bl.a. om der skal tegnes (PEN) el. slettes (SLETTE).
162	CTRL2	<0..15>	Data i dette register angiver: - linietype v. tegning af linier - karatertype v. skrivning af tekst.
163	CSIZE	<0..255>	Angiver karakterstørrelse i forbindelse med skrivning af tekst.
165	DELTA	<0..255>	Giver en linies længde i X-retningen.
167	DELTAY	<0..255>	Giver en linies længde i Y-retningen.
168	Xa	<0..1>	MSB = Most Significant Bits.
169	Xb	<0..255>	LSB = Least Significant Bits. (256 * Xa + Xb) angiver X-koordinaten for start af en linie eller karakter.
170	Ya	<0..1>	MSB = Most Significant Bits.
171	Yb	<0..255>	LSB = Least significant Bits. (256 * Ya + Yb) angiver Y-koordinaten for start af en linie eller karakter.
172	XL	<0..255>	Anvendes i forbindelse med lyspen
173	YL	<0..255>	Anvendes i forbindelse med lyspen

TABEL 1. (Se noter næste side)

- Note:
1. Grafikskærm-billedet betragtes som indlagt i et retvinklet X-Y-kordinatsystem:
 $0 \leq X \leq 511$ og $0 \leq Y \leq 511$.
(X, Y) = (0, 0) er billedets nederste venstre hjørne.
 2. Vedr. XLP og YLP henvises til appendix 1.
 3. Registrene for X- og Y-kordinater omfatter to byte for at kunne dække hele intervallet <0..511>.
 4. I det generelle tilfælde udskrives et sæt kordinater (X', Y') til X- og Y-registrene således:
OUT 168, X' DIV 256 // Xa
OUT 169, X' MOD 256 // Xb
OUT 170, Y' DIV 256 // Ya
OUT 171, Y' MOD 256 // Yb
 5. Tilsvarende indlæses det aktuelle indhold af X- og Y-registrene ved:
X' := INP(168)*256 + INP(169)
Y' := INP(170)*256 + INP(171)
 6. Kommandoregisteret (port 160) fungerer også som et statusregister, der fortæller noget om GP'ens tilstand. Herom senere.

5. SPECIELLE PORTE

=====

Ud over de i Tabel 1. viste porte anvendes i alt 4 porte, som ikke er knyttet til registre i GP'en.

5.1. VALG MELLEM VIDEO og GRAFIK

Port nummer 176 anvendes til at vælge, om dataskærmen skal vise grafikbillede eller videobillede:

```
OUT 176, 0 // der skiftes til VIDEO-billede
OUT 176, 255 // der skiftes til GRAFIK-billede
```

5.2. READ-MODIFY-WRITE (RMW)

RMW er en funktion, der kan være ON eller OFF.

RMW ON betyder, at GP'en ved tegning/skrivning på skærmen ombytter tilstanden for de berørte punkter, dvs lysende punkter gøres mørke og vice versa. Krydses f.eks. to linier med RMW ON, bliver fællesarealet for de to linier mørkt. Ved RMW OFF sker ovennævnte ombytning af punkters tilstand ikke, dvs alle berørte punkter bliver lysende.

Valg mellem RMW ON eller OFF sker via port nummer 177:

```
OUT 177, 0 // RMW OFF
OUT 177, 255 // RMW ON
```


5.3. LÆS REGISTER

Port nummer 178 anvendes i forbindelse med aflæsning af skærbilledets indhold, f.eks. hvis man ønsker at gemme dette på diskette. Omtales nærmere i afsnit 10.

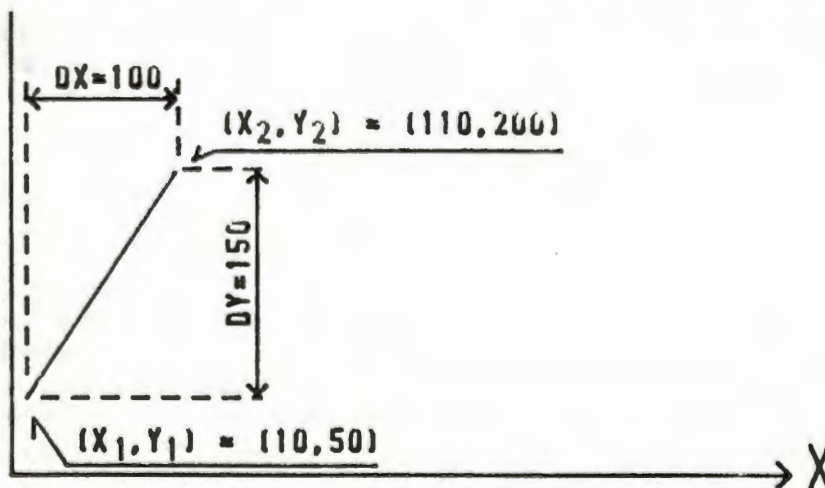
5.4. VALG AF GRAFIKBILLEDE

Som tidligere omtalt kan GP'ens hovedlager rumme to forskellige grafikbilleder. Valget mellem at arbejde med det ene eller det andet af disse billeder sker ved skrivning til port nummer 179:

```
OUT 179, 0 // Grafikbillede 1 (0 - 32 kB)
OUT 179, 255 // Grafikbillede 2 (32 - 64 kB)
```

EKSEMPEL: Idet kommando 17 tegner en ret linie i X- og Y-aksernes positive retning, tegner følgende programdel af en ret linie med start i (10,50) og med udstrækningen 100 i X-aksens retning og 150 i Y-aksens retning:

```
0010 OUT 176, 255 // valg af grafik.
0020 OUT 160, 7 // sletter skærm + nulstiller X-Y-reg
0030 OUT 161, 3 // initier PEN nede - se afsnit 6.
0040 OUT 169, 10 // kun X LSB sættes.
0050 OUT 171, 50 // kun Y LSB sættes.
0060 OUT 165, 100 // sæt DELTAX.
0070 OUT 166, 150 // sæt DELTAY.
0080 OUT 160, 17 // tegn linien.
0090 FOR I:=1 TO 1000 DO // tidsforsinkelse
0100 NEXT I // for at se resultatet.
0110 OUT 176, 0 // valg af VIDEO.
```



FIGUR 2.

Prøv at indtaste og køre dette program.

6. CTRL1 REGISTERET

Via de data, der skrives til CTRL1 registeret bestemmes:

- om man vil skrive (PEN) eller viske ud (SLETTE)
- om PEN/SLETTE skal bevæges hen over skærmen NEDE (aktiv) eller OPPE (inaktiv).
- om skærmen skal betragtes som FLAD eller CYKLISK.

6.1. PEN/SLETTE - OPPE/NEDE

Ønsker man at tegne på skærmen, skal man arbejde med PEN NEDE.

Skal der slettes på skærmen vælges SLETTE NEDE.

PEN eller SLETTE OPPE medfører ingen ændringer på eksisterende skærbillede, når et af de to 'tegneredskaber' bevæges hen over skærmen. Hvad kan så formålet med dette være? Et eksempel på anvendelse kan være at tegne rette linier eller kurver med en bestemt punkteret eller stiplede linietype.

6.2. FLAD/CYKLISK SKÆRM

Hvis 'tegneredskabet' kommer udenfor skærmområdet, vil dette medføre en fejl (som kan aflæses i statusregisteret), som stopper GP'en, hvis der arbejdes med FLAD skærm.

Arbejdes derimod med CYKLISK skærm, vil skærmen blive opfattet som en cylinder, dvs. 'tegneredskabet' vil fortsætte fra den modsatte side af, hvor overskridelse af skærmområdet fandt sted.

6.3. DATA I CTRL1

DATA	PEN/SLETTE	OPPE/NEDE	SKÆRMSTYPE
0	SLETTE	OPPE	FLAD
1	SLETTE	NEDE	FLAD
2	PEN	OPPE	FLAD
3	PEN	NEDE	FLAD
8	SLETTE	OPPE	CYKLISK
9	SLETTE	NEDE	CYKLISK
10	PEN	OPPE	CYKLISK
11	PEN	NEDE	CYKLISK

TABEL 2.

Vedrørende betydningen af øvrige data i CTRL1: se appendix 1.

7. KOMMANDOER

=====

Der findes som nævnt i alt 256 (<0..255>) forskellige kommandoer, der kan sendes til GP'ens kommandoregister via port 160. Disse kommandoer deler sig naturligt ind i nogle hovedgrupper:

- a. kommandoerne 0 - 15: diverse styrekommandoer.
- b. kommandoerne 16 - 31: tegning af rette linier.
- c. kommandoerne 32 - 127: skrivning af karakterer.
- d. kommandoerne 128 - 256: tegning af småvektorer.

b., c. og d. behandles i efterfølgende afsnit, medens vi nedenfor anfører betydningen af en del af gruppe a. kommandoerne.

KOMMANDO BESKRIVELSE

0	Valg af PEN
1	Valg af SLETTE
2	Valg af PEN/SLETTE NEDE
3	Valg af PEN/SLETTE OPPE
4	Sletter (blankstiller) grafikskærm (den p.t. valgte)
5	Sætter X- og Y-registrene til 0 (nul)
6	Sletter skærm og nulstiller X- og Y-reg. (komm. 4 + 5)
7	Sletter skærm, sætter CSIZE-register til minimum karakterstørrelse og nulstiller alle øvrige registre.
13	Nulstiller X-registeret
14	Nulstiller Y-registeret

TABEL 3.

Bemærk, at kommando 0 - 3 skriver data i CTRL1-registeret. Dette sker uden at øvrige data i dette register berøres.

EKSEMPEL:

```
0010 OUT 160, 7 // bl.a. CTRL1=0 ==> SLETTE OPPE ,FLAD skærm
0020 OUT 161, 8 // SLETTE - OPPE - CYKLISK skærm
0030 OUT 160, 0 // PEN - OPPE - CYKLISK skærm
0040 OUT 160, 2 // PEN - NEDE - CYKLISK SKÆRM
```

Resultatet efter linie nr. 0040 kunne også opnås ved:

```
0010 OUT 160, 7
0020 OUT 161, 11
```

8. GP'ens STATUS

=====

Z-80 og GP kan arbejde parallelt, dvs. medens GP'en er i færd med at udføre en kommando - f.eks. tegne en del af en kurve - kan Z-80 beregne data for den næste del af kurven. Imidlertid må man i forbindelse med afsendelse af data fra Z-80 til GP iagttage:

- der må ikke sendes en ny kommando til GP'en, medens en kommando er under udførelse.
- sendes nye data til et register, hvis indhold anvendes til udførelse af en igangværende kommando, kan kommandoen blive udført med de nye data.

I førstnævnte tilfælde stopper GP'en, medens sidstnævnte tilfælde som oftest vil medføre et ikke ønsket resultat.

Oplysning om GP'ens status, dvs om en kommando er under udførelse eller om GP'en er klar til en ny kommando, fås ved at indlæse og analysere indholdet af GP'ens statusregister via port 160.

BEMÆRK: skrives der til port 160, fungerer det tilknyttede register som kommandoregister. Læses der derimod fra port 160, er det samme register et statusregister.

Indlæsning af GP'ens status sker ved:

```
STATUS:= INP(160)
```

og analysen kan f.eks. udføres således:

```
KOMMANDOSTATUS:= (STATUS MOD 8) DIV 4
```

hvilket svarer til at teste, om den tredje bit i den indlæste byte er 1 eller 0, svarende til "GP klar" hhv. "GP optaget".

Ovenstående laves nturligvis som en procedure, hvofra der først returneres, når GP er klar til en ny kommando:

```
PROC STATUSPROC  
  REPEAT  
    UNTIL (INP(160) MOD 8) DIV 4 = 1  
  ENDPROC STATUSPROC
```

Denne procedure kaldes efter behov før ændring af dataregistre og før udstedelse af ny kommando.

EKSEMPEL: se appendix 2 eksempel 1.

Test af GP'ens status kan også ske via en assembleroutine indbygget i COMAL-80.

EKSEMPEL: se proceduregruppen COMGRAPH procedurerne INITSTATUS og STATUSINIT.

9. TEGNING AF RETTE LINIER

Ved tegning af rette linier er følgende registre interessante:

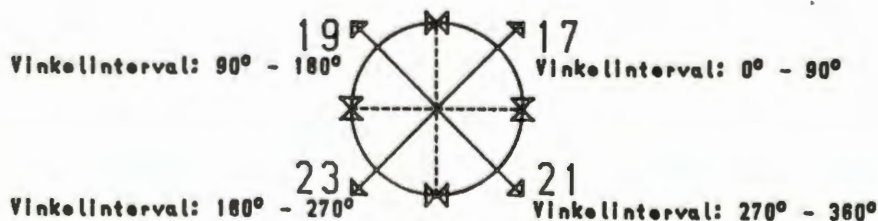
- X- og Y-registrene for angivelse af liniens startpunkt.
- DELTAX og DELTAY registrene for angivelse af liniens længde.
- CTRL1 registeret for valg af PEN/SLETTE enten OPPE el. NEDE.
- CTRL2 registeret for angivelse af linietype - omtales senere.
- CMD registeret for udstedelse af tegnekommando.

DELTAX og DELTAY registrene angiver den numeriske værdi af en linies udstrækning i hhv X- og Y-retningen. Betragter man imidlertid et vilkårligt punkt i koordinatsystemet for grafikbilledet vil der i almindelighed kunne tegnes rette linier ud fra dette punkt i 4 forskellige retninger givet ved koordinataksernes positive og negative retninger. Det vil sige, at DELTA-værdierne i virkeligheden skal regnes med fortegn. Fortegnene for DELTA-værdierne knyttes til den kommando, der udstedes for tegning af en ret linie.

Fire af de kommandoer, der findes til tegning af rette linier er:

KOMMANDO:	DELTAX:	DELTAY:	LINIETYPE:
17	positiv	positiv	skråt opad mod højre.
19	negativ	positiv	skråt opad mod venstre.
21	positiv	negativ	skråt nedad mod højre.
23	negativ	negativ	skråt nedad mod venstre.

TABEL 4.



FIGUR 3.

Under tegning af en ret linie opdaterer GP'en X- og Y-registrene således at disse efter udførelse af tegnekommandoen indeholder:

$$X' + \text{DELTAX} \quad \text{hhv} \quad Y' + \text{DELTAY}$$

hvor (X', Y') er liniens startpunkt.

EKSEMPEL: se appendix 2 eksempel 2, hvor der udføres det samme som i eksempel 1, men nu med brug af alle ovennævnte kommandoer samt opdateringen af X- og Y-registrene.

Med de anførte fire kommandoer kan man tegne rette linier af enhver art.

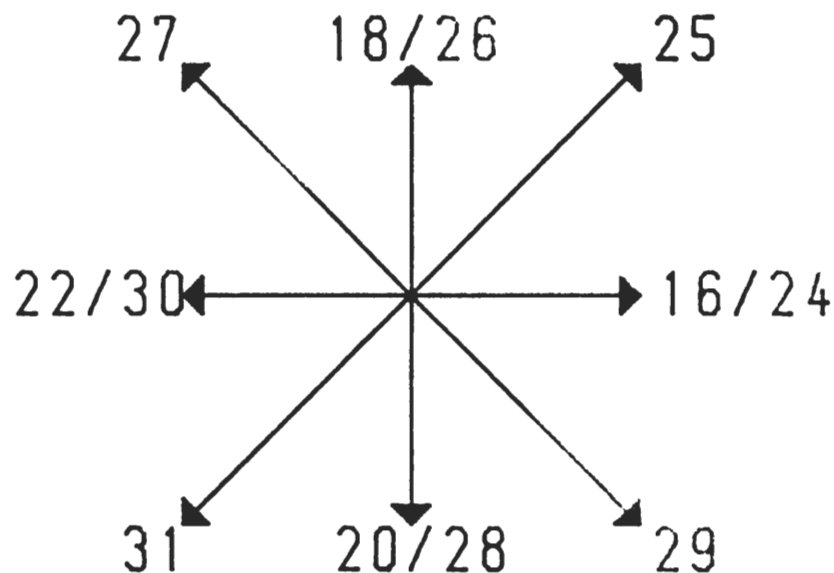
Der findes en række specialkommandoer til tegning af linier med specielle retninger:

- linier parallelle med X- el. Y-aksen dvs. hhv DELTAY eller DELTAX lig 0 (nul).
- linier parallelle med diagonalerne dvs. DELTAX = DELTAY.

KOM- MANDO:	DELTAX:	DELTAY:	LINIE:
16	positiv	0	paral. m. X-akse mod højre.
18	0	positiv	paral. m. Y-akse opad.
20	0	negativ	paral. m. Y-akse nedad.
22	negativ	0	paral. m. X-akse mod venstre.
24	positiv	0	paral. m. X-akse mod højre.
25	positiv	positiv	paral. m. diagonal opad mod højre.
26	0	positiv	paral. m. Y-akse opad.
27	negativ	positiv	paral. m. diagonal opad mod venstre.
28	0	negativ	paral. m. Y-akse nedad.
29	positiv	negativ	paral. m. diagonal nedad mod højre.
30	negativ	0	paral. m. X-akse mod venstre.
31	negativ	negativ	paral. m. diagonal nedad mod venstre.

TABEL 5.

- NOTE 1. Et '0' ved kommando 16, 18, 20, 22, 24, 26, 28 og 30 betyder, at pågældende DELTA-register ignoreres, dvs. betragtes som 0 (nul) uanset indhold.
2. For kommandoerne 25, 27, 29 og 31 gælder, at den største af værdierne DELTAX/DELTAY anvendes, medens mindste værdi ignoreres.



FIGUR 4.

EKSEMPEL: Eksempel 3 og 4 i appendix 2 viser tegning af figurer med et minimum af skrivning til DELTA-registrene.

Tegning af kurver foregår ved tegning af ganske korte linier, som kan have forskellige retninger. I sådanne tilfælde er det bekemt (programmeringsmæssigt) at beregne sig frem til den kommando, der skal anvendes. Dette kan ske ved følgende fomel:

$$\text{KOMMANDO} := 17 + (\text{SGN}(\text{DX}) < 0) * 2 + (\text{SGN}(\text{DY}) < 0) * 4$$

hvor DX og DY er tilvæksterne i X- og Y-retningen regnet med fortegn. Den viste formel giver en af de kommandoer, der er anført i tabel 4.

EKEMPEL: i eksempel 5 i appendix 2 er vist et program til tegning af en sinuskurve.

10. CTRL2 REGISTERET

Ved tegning af rette linier kan man vælge mellem 4 forskellige linietyper (stregtyper): kontinuert, punkteret, stiplet og stiplet/punkteret - se tabel 6.

Tilsvarende kan man ved skrivning af tekst arbejde med 4 forskellige kraktertyper: lodret og skrå skrift på hhv vandret og lodret linie - se tabel 6.

Valg af linie- og karaktertype sker ved data til CTRL2-registret som følger:

CTRL2	LINIETYPE	KARAKTERTYPE
0	KONTINUERT	LODRETTE KARAKTERER VANDRET LINIE A
1	PUNKTERET	
2	STIPILET	
3	STIPILET/PUNKTERET	
4	KONTINUERT	SKRÅ KARAKTERER VANDRET LINIE A
5	PUNKTERET	
6	STIPILET	
7	STIPILET/PUNKTERET	
8	KONTINUERT	LODRET KARAKTERER LODRET LINIE A
9	PUNKTERET	
10	STIPILET	SKRÅ KARAKTERER LODRET LINIE A
11	STIPILET/PUNKTERET	
12	KONTINUERT	
13	PUNKTERET	
14	STIPILET	
15	STIPILET/PUNKTERET	

TABEL 6.

EKSEMPEL: se pocedurerne SETLINETYPE og SETCHRTYPE i proceduregruppen COMGRAPH. Her kan man bl.a. se, hvorledes man kan ændre linietype uden at berøre karaktertypen og vice versa.

Se også eksempel 4 i appendix 2.

11. KARAKTERER.

GP'en har en indbygget karaktergenerator, ved hjælp af hvilken man kan skrive tekster med variabel karakterstørrelse og - som vi så i foregående afsnit - karaktertyper. Det til rådighed værende karaktersæt omfatter et fuldstændigt 96 karakter stort ASCII alfabet. Alfabetet er internationalt og omfatter ikke specielle danske nationale karakterer. Proceduregruppen COMGRAPH omfatter faciliteter for behandling af danske karakterer.

Ved skrivning af tekster skrives en karakter ad gangen, idet der findes en kommando (32 til 127) pr. karakter. I øvrigt er følgende registre involveret i karakterskrivning:

- X-Y-registrene for angivelse af karakterens startpunkt dvs. karakterens nederste venstre punkt.
- CTRL1 registeret for valg af PEN/SLETTE og OPPE/NEDE.
- CTRL2 registeret for valg af karaktertype.
- CSIZE registeret for valg af karakterstørrelse - se nedenfor.
- CMD registeret for skrivning af karakter.

I appendix 1 findes en detaljeret oversigt over det til rådighed værende karaktersæt samt de enkelte karakterers udformning.

Hver karakter findes i en grundudgave på 5 x 8 punkter. Denne grundudgave kan forstørres i såvel X- som Y-retningen ved at multiplicere med faktorer mellem 1 og 16 for det enkelte punkts udstrækning i X- og Y-retningen. Dette giver ialt 256 forskellige karakterstørrelser, som altså hver for sig kan udskrives i 4 forskellige typer (tabel 6) bestemt ved CTRL2-registeret. Bestemmelse af karakterstørrelsen sker ved data til CSIZE-registeret. Ønsker man at multiplicere grundudgaven med MX i X-retningen og MY i Y-retningen, skal der til CSIZE skrives:

OUT 163, MX*16 + MY

(MX, MY) = (1, 1) svarer til mindste karakterstørrelse (grundudgaven), medens (MX, MY) = (0, 0) svarer til maksimal størrelse, dvs. 0 tolkes som multiplikationsfaktoren 16.

EKSEMPEL: Ønskes en karakterforstørrelse på 4 gange i X-retningen og 6 gange i Y-retningen:

OUT 163, 70

Karakterer, der udskrives herefter optager et 20 x 48 punkter stort felt (20 i X-retningen, 48 i Y-retningen). Ved forstørrelsen er det hvert enkelt punkt i grundudgaven, der multipliceres m. hhv. 4 og 6.

ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890
abcdefghijklmnopqrstuvwxyz?@+*-=<>%

ABCDEabcde12345?+%

ABCabc123

} GRUNDUDGAVE
5 x 8 PKT.

MX = MY = 2
10 x 16 PKT.

MX = 4, MY = 6
20 x 48 PKT.

Når nødvendige data er skrevet til de relevante GP-registre, udstedes en kommando, hvis værdi er den til den ønskede karakter svarende decimalværdi i ASCII-alfabetet - jvfr. f.eks. COMAL-80 manualen. I appendix 1 findes også en tabel, som viser afvigelserne fra den danske udgave.

EKSEMPEL: Ønsker man at udskrive et A, udstedes kommando 65:

OUT 160, 65

Se i øvrigt appendix 2 eksempel 6.

Ud over de 96 ASCII-karakterer findes to andre 'karakterer' el. kommandoer, der følger CSIZE og karaktertypen valgt i CTRL2:

- kommando 10: skriver et 5 x 8 punkter stort fuldt udfyldt felt forstørret i overensstemmelse med CSIZE-registerets indhold og følgende karaktertypen specificeret i CTRL2-registeret. Denne kommando er velegnet til sletning af tekster på skærmen.
- kommando 11: som kommando 10, men grundfeltet er 4 x 4 punkter. Med denne kommando kan man på en gang slette eller skrive et 64 x 64 (max. karakterstørrelse) punkter stort område.

EKSEMPEL: se eksempel 7 i appendix 2, hvor kommando 10 er anvendt til at opnå en blinkende tekst på skærmen. Samme opgave er løst i eksempel 8 med anvendelse af RMW.

Ligesom ved tegning af linier sker der ved skrivning af karakterer en opdatering af X- og Y-registrene, idet disse efter skrivning af en karakter forstørret (MX, MY) startende i punktet (X', Y') vil have indholdet:

$$X'' = X' + MX * 6 \qquad Y'' = Y'$$

med andre ord, ved skrivning af flere karakterer efter hinanden uden ændring i X og Y, fås et karaktermellemlum på MX punkter.

Kommando 10 følger ovennævnte regler for opdatering af X-registeret, medens der i forbindelse med kommando 11 sker en opdatering af X-registeret på 5 * MX, altså hænger felterne her sammen.

12. SMAVEKTORER.

Vi har nu behandlet (næsten) halvdelen af de 256 kommandoer.

Den sidste halvdel udgøres af en samling kommandoer til tegning af hvad vi kan kalde 'småvektorer', dvs linier hvis udstrækning i X-og Y-retningen begrænses sig til intervallet 0 til 3.

Kommandoerne til tegning af småvektorer indeholder udover information om vektorens retning også information om liniens længde, dvs at man i forbindelse med disse vektorer ikke anvender DELTAX- og DELTAY-registrene. Herved spares programplads og -afviklingstid. Kommandoerne er velegnet for tegning af buer, kurver o.lg.

Kommandoerne til tegning af småvektorer falder ligesom kommandoerne til tegning af længere linier i to kategorier:

- a. **Generelle**, for tegning af linier i enhver tænkelig retning. Idet DX og DY betegner tilvæksten i hhv. X- og Y-retningen, kan disse kommandoer udregnes efter formlen:

$$129 + \text{ABS}(\text{DX}) * 32 + \text{ABS}(\text{DY}) * 8 + (\text{SGN}(\text{DX}) < 0) * 2 + (\text{SGN}(\text{DY}) < 0) * 4$$

- b. **Vektorer parallelt med X- eller Y-aksen** ved ignorering af DY hhv. DX. Disse kommandoer beregnes af:

$$128 + \text{ABS}(\text{DX}) * 32 + \text{ABS}(\text{DY}) * 8 + \text{RETNINGSKODE}$$

hvor RETNINGSKODE =

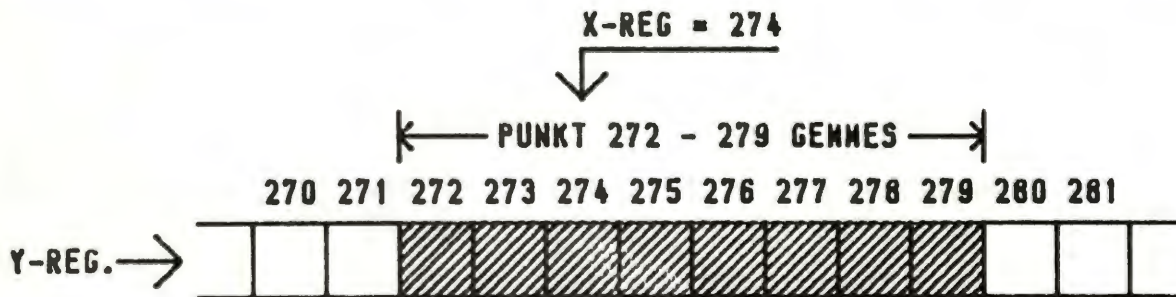
- 0: parallel m. X-aksen - retning mod højre
- 2: parallel m. Y-aksen - retning opad.
- 4: parallel m. Y-aksen - retning nedad.
- 6: parallel m. X-aksen - retning mod venstre.

Hver af de to kategorier omfatter 64 kommandoer.

13. AFLÆSNING AF SKÆRM.

Et grafikbillede er ikke blot en række figurer udtegnet af et program på samme måde fra gang til gang. Et grafikbillede kan også dannes ved en kommunikation mellem bruger og program (jvfr. f.eks. programmet REGRS, der følger med MPS-24 grafikmodul). I sådanne tilfælde kan det være af interesse at kunne aflæse skærmen og enten gemme mønsteret på diskette eller printe det ud på en passende matrixprinter. Til dette brug findes der mellem Z-80 og GP'en et register, som er knyttet til port 178, og som anvendes i forbindelse med kommando 15.

Udstedes en kommando 15, vil tilstanden af 8 punkter på skærmen blive udlæst til nævnte register. De 8 punkter, der er tale om, er de 8 punkter, der befinder sig på linien adresseret af Y-registeret, og som starter i punktet bestemt af X-registerets indhold afrundet til nærmeste mindre hele multiplum af 8, altså:



FIGUR 5.

Punkterne udlæses som bit til registeret, således at punktet adresseret af X- og Y-registeret bliver bit 0 i udlæsningsregisteret. Næste punkt mod højre bliver bit 2 etc.

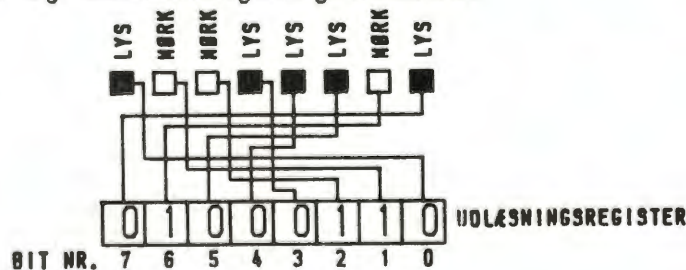
Tilstanden af de enkelte punkter registreres som et binært 0 (nul) hvis punktet er lysende og som 1 ellers.

EKSEMPEL: Hvis kommando 15 udstedes, når (X, Y) = (50, 100), vil udlæsningsregisterets bit komme til at svare til:

- bit 0: punkt (48, 100)
- bit 1: punkt (49, 100)
- bit 2: punkt (50, 100) etc.

(bit 0 er mindst betydende bit i udlæsningsregisteret).

EKSEMPEL: Følgende figur viser sammenhæng mellem 8 punkter på skærmen og udlæsningsregisteret.



FIGUR 6.

Efter udlæsning af de otte punkters tilstand til udlæsningsregisteret, kan dette registers indhold (de otte punkter) indlæses til programmet ved INP(178).

Det skal bemærkes, at indholdet af X- og Y-registrene er uændret efter udførelsen af kommando 15.

BRUGERVEJLEDNING FOR MPS-24 GRAFIKMODUL

JUNI.84/ULJ

Appendix 1

DETALJERET BESKRIVELSE AF
GRAFIKPROCESSOREN

CMD COMMAND REGISTER (Address : 0₁₆)

The CMD register is an 8-bit write-only register. Each write operation in this register causes a command to be executed, upon completion of the time necessary for synchronizing the microprocessor access and the GDP's CK clock.

Several types of command are available :

- vector plotting
- character plotting
- screen erase
- light pen circuitry setting
- access to the display memory through an external circuitry.
- indirect modification of the other registers (commands that make it possible for the X, Y, DELTAX, DELTAY, CTRL1, CTRL2 and CSIZE registers to be amended or scratched).

STATUS REGISTER (Address 0₁₆)

The STATUS register is an 8-bit read-only register. It is used to monitor the status of the executing statements entered into the circuit, and more specifically to avoid the need for modifying a register that is already used for the command currently executing.

- Bit 0 : When low, this bit indicates that a light pen sequence is currently executing.
When high, it indicates that no light pen sequence is currently executing.
- Bit 1 : This bit is high during vertical blanking. It is the VB signal recopy.
- Bit 2 : When low, this bit indicates that a command is currently executing.
When high, this bit indicates that the circuit is ready for a new command.
- Bit 3 : When low, this bit indicates that the X and Y registers point within the display window.
When high, this bit indicates that the X and Y registers are pointing outside the memory display.
This bit is the logic OR of the unused MSBs of the X and Y registers.
- Bit 4 : When high, this bit indicates that an interrupt has been initiated by the completion of a light pen running sequence. Such an interrupt is enabled by bit 4 in CTRL1 register.
- Bit 5 : When high, this bit indicates that an interrupt has been initiated by vertical blanking. Such an interrupt is enabled by bit 5 in CTRL1 register.

Bit 6 : When high, this bit indicates that an interrupt has been initiated by the completion of execution of a command. Such an interrupt is enabled by bit 6 in CTRL1 register.

Bit 7 : When high, this bit indicates that an interrupt has been initiated. It is the logic OR of bits 4, 5 and 6 in STATUS register. The \overline{IRQ} output state is always the opposite of the status of this bit.

Note : Bits 4, 5, 6 and 7 are reset low by a read of the STATUS register.

XLP AND YLP REGISTERS (Addresses C₁₆ and D₁₆)

The XLP and YLP registers are read-only registers, with 7 and 8 bits respectively. Upon completion of a light pen running sequence, they contain the display address sampled by the first edge appearing rising on the LPCK input. The use of such registers is discussed in section : **Use of light pen circuitry.**

NOTES :

1. All internal registers may be read or written at any time by the microprocessor. However, the precautions outlined below should be observed :
 - Do not write into the CMD register if execution of the previous command is not completed (bit 2 of STATUS register).
 - Do not alter any register if it is used as an input parameter for the internal hardwired systems (e. g. : modifying the DELTAX register while a vector plotting sequence is in progress).
 - Do not read a register that is being asynchronously modified by the internal hardwired systems (e. g. : reading the X register while a vector plotting sequence is in progress may be erroneous if CK and \overline{E} are asynchronous).
2. On powering up, the writing devices may have any status. Before entering a command for the first time, it is necessary to wait until all functions currently underway are completed, which information can be derived from the STATUS register.

REGISTER DESCRIPTION

X AND Y REGISTERS (Addresses : 8_{16} , 9_{16} , A_{16} , B_{16})

The X and Y registers are 12-bit read-write registers. They indicate the position of the next dot to be written into the display memory. They have no connection at all with the video signal generating scan, but they point the write address, in the same way as the pen address on a plotter.

These 2 registers are incremented or decremented, prior to each write operation into the display memory, by the internal vector and character generators, or they may be directly positioned by the microprocessor.

This 2×12 bit write address covers a 4096×4096 point addressing space. Only the LSBs are used here, since the maximum definition of the picture actually stored is 512×512 pixels (picture elements).

The MSBs are either ignored or used to inhibit writing where the actual screen is regarded as being a window within a 4096×4096 space.

The above features along with the relative mode description of all picture component elements make it possible to automatically solve the great majority of edge cut-off problems.

DELTA X AND DELTA Y REGISTERS (Addresses : 5_{16} , 7_{16}).

The DELTA X and DELTA Y registers are 8-bit read-write registers. They indicate to the vector generator the projections of the next vector to be plotted, on the X and Y axes respectively. Such values are unsigned integers. The plotting of a vector is initiated by a write operation in the command register (CMD).

CSIZE REGISTER (Address : 3_{16})

The CSIZE register is an 8-bit read-write register. It indicates the scaling factors of X and Y registers for the symbols and characters. 98 characters are generated from a 5×8 pixel matrix defined by an internal ROM. In the standard version, it contains the alphanumeric characters in the ASCII code which may be printed, together with a number of special symbols.



Each symbol can be increased by a factor P(X) or Q(Y). These factors are independent integers which may each vary from 1 to 16 and which are defined by the CSIZE register. The symbol generation sequence is started after writing the ASCII code of the symbol to be represented in the CMD register.

CTRL1 REGISTER (Address : 1_{16}).

The CTRL1 register is a 7-bit read-write register, through which the general circuit operation may be fed with the required parameters.

Bit 0 : When low, this bit inhibits writing in display memory (equivalent to pen or eraser up).
When high, this bit enables writing in display memory (pen or eraser down).
This bit controls the DW output.

Bit 1 : When low, this bit selects the eraser.
When high, this bit selects the pen.
This bit controls the DIN output.

Bit 2 : When low, this bit selects normal writing mode (writing apart from the display and refresh periods, which are a requirement for the dynamic storages) in display memory.
When high, this bit selects the high speed writing mode : the display periods are deleted. Only the dynamic storage refresh periods are retained.

Bit 3 : When low, this bit indicates that the 4096×4096 space is being used (the 12 X and Y bits are significant)
When high, this bit selects the cyclic screen operating mode.

Bit 4 : When low, this bit inhibits the interrupt triggered by the light pen sequence completion.
When high, this bit enables the interrupt.

Bit 5 : When low, this bit inhibits the interrupt release by vertical blanking.
When high, this bit enables the interrupt.

Bit 6 : When low, this bit inhibits the interrupt indicating that the system is ready for a new command.
When high, this bit enables the interrupt.

Bit 7 : Not used. Always low in read mode.

CTRL2 REGISTER (Address : 2_{16})

The CTRL2 register is a 4-bit read/write register, through which the plotting of vectors and characters may be denoted by parameters.

Bit 0, 1 : These 2 bits define 4 types of lines (continuous, dotted, dashed, dash-dotted).

Bit 2 : When low, this bit defines straight writing.
When high, it defines tilted characters.

Bit 3 : When low, this bit defines writing along an horizontal line.
When high, this bit defines writing along a vertical line.

Bit 4, 5, 6, 7 : Not used. Always low in read mode.

HARDWIRED WRITE PROCESSOR OPERATION IN DISPLAY MEMORY

The hardwired write processors are sequenced by the master clock CK. They receive their parameters from the microprocessor bus. They control the X, Y write address, and the DIN, DW, MFREE and IRQ outputs.

These harwired processors operate in continuous mode. In the event of conflicting access to the display memory, the display and refresh processors have priority.

Since command decoding is synchronous with the CK master clock, any write operation into the (CMD) command register triggers a synchronizing mechanism which engages the circuit for a maximum of 2 CK cycles when the \bar{E} input returns high. The circuit remains engaged throughout command execution.

No further command should be entered as long as bit 2 in STATUS register is low.

VECTOR PLOTTING

The internal vector generator makes it possible to modify, within the display memory, all the dots which form the approximation of a straight line segment. All vectors plotted are described by the origin dot and the projections on the axes.

The starting point co-ordinates are defined by the X, Y register value, prior to the plotting operation.

Projections onto the axes are defined as absolute values by the DELTAX and DELTAY registers, with the sign in the command byte that initiates the vector plotting process.

The vector approximation achieved here is that established by J. F. BRESENHAM ("Algorithm for computer control of a digital plotter"). This algorithm is executed by a hardwired processor which allows for a further vector component dot to be written in each CK clock cycle.

During plotting, the display memory is addressed by the X, Y registers, which are incremented or decremented.

On completion of vector plotting, they point to the end of this vector.

All vectors may be plotted using any of the following line patterns : continuous, dotted, dashed, dash-dotted, according to the 2 LSBs in register CTRL2.

Irrespective of such patterns, the plotting speed remains unchanged. The "pen down-pen up" statement required for plotting non-continuous lines is controlled by the DW output.

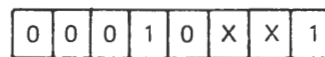
For a specified non-continuous line plotted vector, defined by DELTAX, DELTAY, CTRL2, CMD, the DW sequencing during the plotting process is always the same, irrespective of vector origin and of the nature of previous plots. This feature guarantees that a specified vector can be deleted by plotting it again after moving X and Y to the starting point, and complementing bit 1 in register CTRL1.

Since the vector plotting initiation command defines the sign of the projections onto the axes, all vectors may be plotted using 4 different commands.

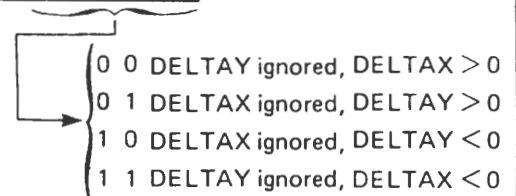
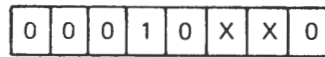
For increased programming flexibility, the system incorporates 16 different commands, supplemented by a set of 128 commands which make it possible to plot small size vectors by ignoring the DELTAX and DELTAY registers.

Such commands are as follows :

- Basic commands

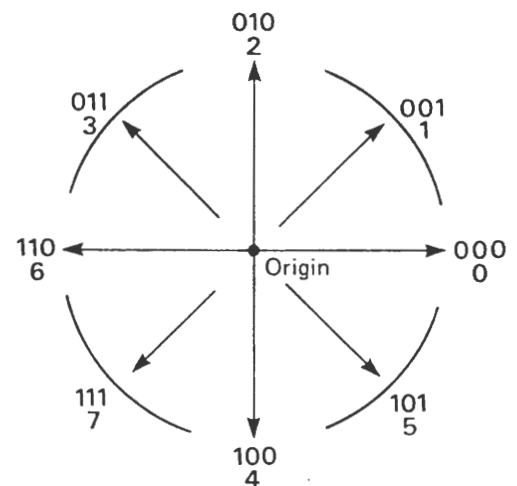


- Commands which allow ignoring the DELTAX or DELTAY registers by considering them as of zero value.

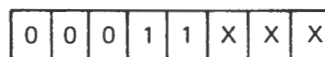


Note : Bits 1 and 2 always have the same sign meaning.

These 8 codes may be summarized by the following diagram :

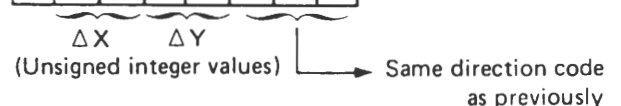
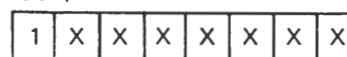


- Commands which allow ignoring the smaller of the two DELTAX and DELTAY registers, by considering it as being equal to the larger one, which is the same as plotting vectors parallel to the axes or diagonals, using a single DELTA register.



Same direction codes as above.

- Commands in which the two registers DELTAX and DELTAY may be ignored by specifying the projections through the CMD register (0 to 3 steps for each projection).



EXAMPLE : PLOTTING A DOTTED VECTOR

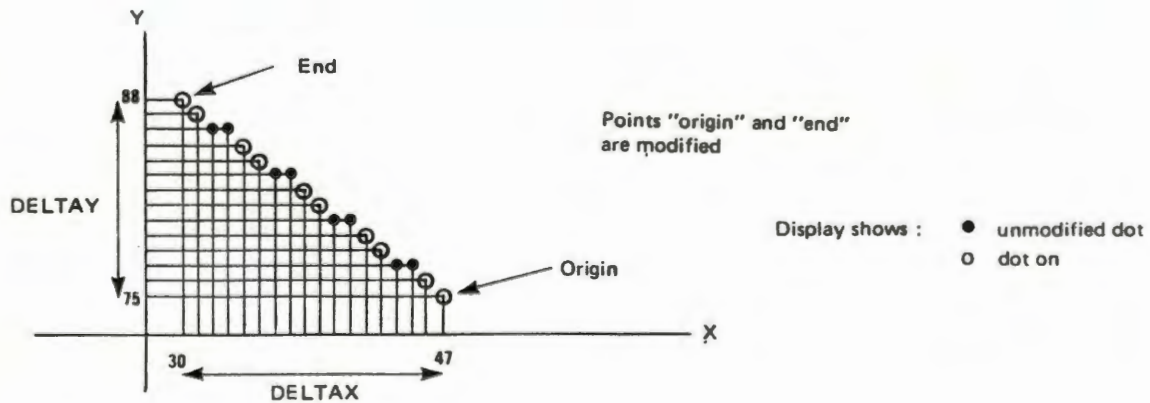
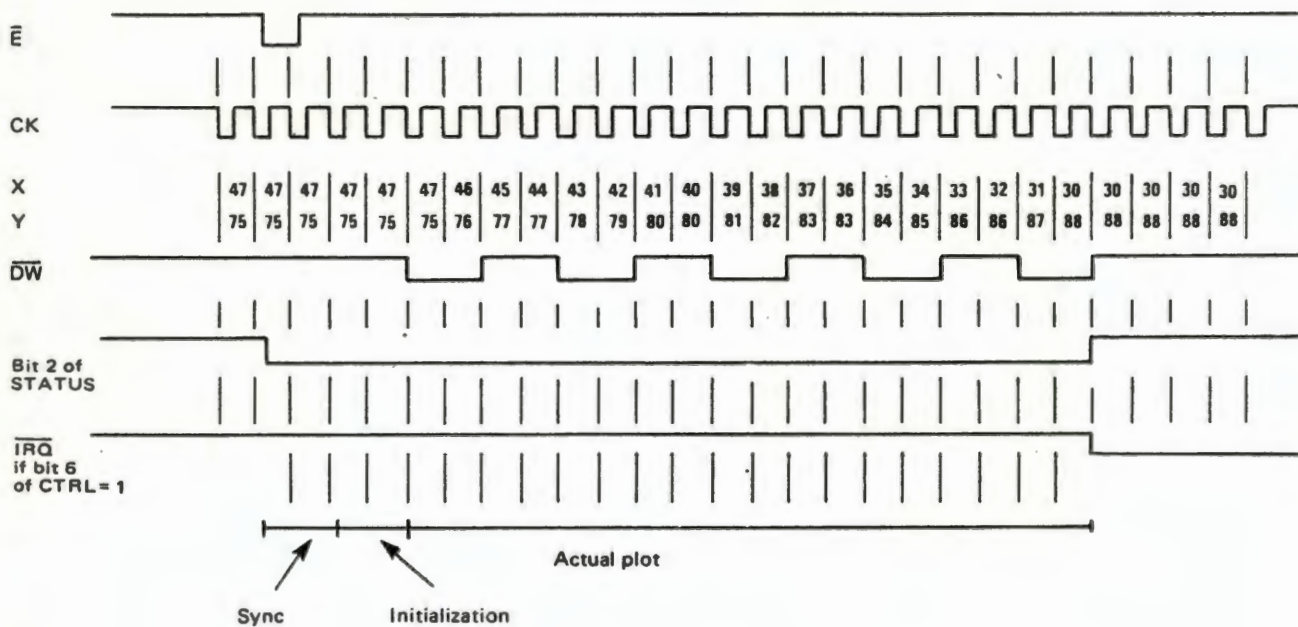
Origin : $\begin{cases} X = 47_{10} \\ Y = 75_{10} \end{cases}$

CMD = 13_{16} Corresponding to
 - Basic command,
 - DELTAX < 0
 - DELTAY > 0

Projections: $\begin{cases} DELTAX = 17_{10} \\ DELTAY = 13_{10} \end{cases}$

CTRL1 = 03_{16} Pen down
 CTRL2 = 1_{16} Dotted vector :
 2 dots on,
 2 dots off.

Plotting cycle sequence : (It is assumed that the vector generator is not interrupted by the display or refresh cycle).



Note :

Plotting a vector with DELTAX = DELTAY = 0 writes the point X, Y in memory. It occupies the vector generator for synchronization, initialization and one write cycle.

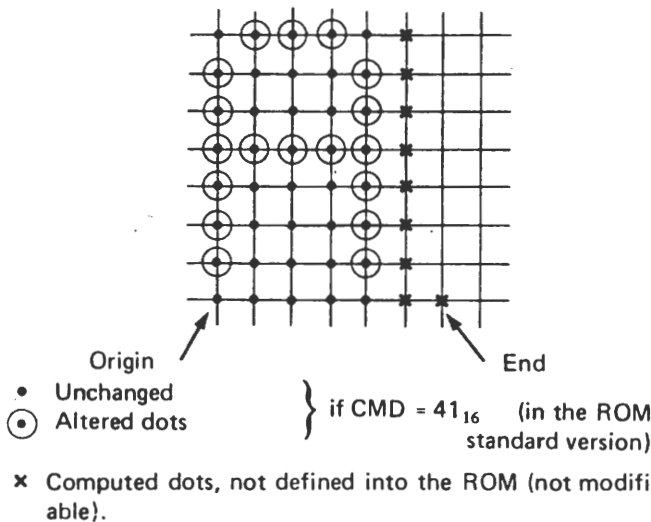
CHARACTER AND SYMBOL GENERATOR

The character generator operates in the same way as the vector generator, i.e. through incrementing or decrementing the X, Y registers, in conjunction with a \overline{DW} output control.

It receives parameters from the CSIZE, CTRL2 and CMD registers. The characters plotted are selected, according to the CMD value, out of 98 matrices (97 8-dot high x 5-dot wide rectangular matrices, and one 4 dot x 4 dot matrix) defined in an internal ROM. Two scaling factors may be applied to the characters plotted using X and Y defined by the CSIZE register. The characters may be tilted, according to the content of register CTRL2.

Basic matrix

Upon completion of a character writing process, the X and Y registers are positioned for writing a further character next to the previous one, with a 1 dot spacing, i.e. Y is restored to its original value and X is incremented by 6.



Scaling factors

Each individual dot in the 5 x 8 basic matrix may be replaced by a P x Q size block.

P : X co-ordinate scaling factor

Q : Y co-ordinate scaling factor

The character size becomes 5P x 8Q. Upon completion of the writing process, X is incremented by 6P. The CK clock cycle count required is 6P x 8Q.

USE OF LIGHT PEN CIRCUITRY

A rising edge on the LPCK input is used to sample the current display address in the XLP and YLP registers, provided that this edge is present in the frame immediately following loading of the 08₁₆ or 09₁₆ code into the CMD register.

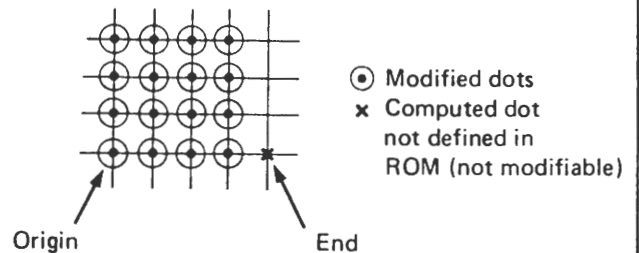
Here, the frame origin is counted starting with the VB falling edge. With code 08₁₆, the WHITE output recopies the BLK signal from the frame origin up to the rising edge on the LPCK input, or when VB starts rising again, if the LPCK input remains low for the entire frame. With code

P and Q may each take values from 1 through 16. They are defined by the CSIZE register. Each value is encoded on 4 bits, value 16 being encoded as 0₁₆.

In register CSIZE, P is encoded on the 4 MSBs and Q on the 4 LSBs.

Among the 97 rectangular matrices available in the standard ROM, 96 correspond to CMD values ranging from 20₁₆ to 7F₁₆, and the 97th matrix to 0A₁₆. In the standard version, these values correspond to the 96 printable characters in the ASCII set. The 97th character is a 5P x 8Q block which may be used for deleting the other characters.

The 98th code (0B₁₆) is used to plot a 4P x 4Q graphic block. It locates X, Y, without spacing for the next symbol. Such a block makes it possible to pad uniform areas on the screen.



Tilted characters

All characters may be modified to produce tilted characters or to mark the vertical co-ordinate with straight or tilted type symbols. Such changes may be achieved using bits 2 and 3 in register CTRL2.

Note : Scaling factors P and Q are always applied within the co-ordinates of the character before conversion.

Character deletion

A character may be deleted using either the same command code or command code 0A₁₆. In either case, bit 1 in register CTRL1 should be inverted, the origin should be the same as prior to a character plotting operation, as should the scaling factors.

Note : Vector generator and character generator operate in similar ways :

	VECTOR	CHARACTER
Dimensions	DELTAX, DELTAY	CSIZE, tilting
\overline{DW} modulation	Type of line	Character code

09₁₆, the WHITE output is not activated.

The YLP address is 8-bit coded since there are 256 display lines in each frame. The XLP address is 6-bit coded since there are 64 display cycles in each line.

These 6 bits are left justified in the XLP register. XLP and YLP register contents match the write address if FMAT is low (or for the EF9366), but should be multiplied by 2 if FMAT is high, so as to be able to match the write address.

The address sampled into XLP corresponds to the current memory cycle. Dots detected by the light pen were addressed in the memory during the previous cycle. Hence, 1 should be subtracted from bit 2 in XLP register where the light pen electronic circuitry does not produce any additional delay.

If the rising edge on input LPCK occurs while VB is low, then the LSB in XLP is set high. This bit acts as a status signal which is reset to the low state by reading register XLP or YLP.

SCREEN BLANKING COMMANDS

Three commands (04_{16} , 06_{16} , 07_{16}) will set the whole display memory to a status corresponding to a "black display screen" condition. Another command ($0C_{16}$) may be used to set the whole memory to a status other than black (this condition being determined by bit 1 in register CTRL1).

The 4 commands outlined above use the planned scanning of the memory addresses achieved by the display stage. The X and Y registers are not affected by commands 04_{16} and $0C_{16}$. Hence, the time required is that corresponding to one frame (EF9366 or FMAT low) or two frames (FMAT high). The time corresponding to the completion of the

The rising edge first received (LPCK or VB) sets bit 0 in STATUS register high. An interrupt is initiated if bit 4 in CTRL1 is high.

When commands 08_{16} or 09_{16} have been decoded, bit 2 of the status register goes high (circuit ready for any further command) and bit 0 goes low (light pen operating sequence underway).

frame currently executing when the CMD register is loaded, should be added to the above time.

For the screen blanking process, the frame origin is counted starting with the VB falling edge.

The only signals affected here are the \overline{DW} output, which remains low when VB is low, and the DIN output which is forced high where the 04_{16} , 06_{16} and 07_{16} commands are entered.

Such commands are activated without requiring action by WO input or bit 2 in register CTRL1. While these commands are executing, bit 2 in STATUS register remains low.

EXTERNAL REQUEST FOR DISPLAY MEMORY ACCESS (\overline{MFREE} OUTPUT)

On writing code $0F_{16}$ into the CMD register, the \overline{MFREE} output is set low by the circuitry, during the next free memory cycle.

Apart from the display and refresh periods, this cycle is the first complete cycle that occurs after input \overline{E} is reset high.

During this cycle, those addresses output on DAD and MSL correspond to the X and Y register contents: \overline{DW} is high, \overline{ALL} is high.

Should the memory be engaged in a display or refresh operation, (which is the case when \overline{ALL} is low), then this cycle is postponed to be executed after \overline{ALL} is reset high. The maximum waiting time is thus 64 cycles.

The \overline{MFREE} signal may be used e. g. for performing a read or write operation into a register located between the display memory and the microprocessor bus.

INTERRUPTS OPERATION

An interrupt may be initiated by three situations denoted by internal signals:

- Circuit ready for a further command
- Vertical blanking signal
- Light pen sequence completed.

These three signals appear in real time in the STATUS register (bits 0, 1, 2). Each signal is cross-referenced to a mask bit in the register CTRL1 (bits 4, 5, 6).

If the mask bit is high, the first rising edge that occurs on the interrupt initiating signal sets the related interrupt flip-flop circuit high.

The outputs from these three flip-flop circuits appear in the STATUS register (bits 4, 5, 6). If one flip-flop circuit

is high, bit 7 in the STATUS register is high, and pin \overline{IRQ} is forced low.

A read operation in the STATUS register resets its 4 MSBs low, after input \overline{E} is reset high.

The three interrupt control flip-flops are duplicated to prevent the loss of an interrupt coming during a read cycle of the STATUS register.

The status of bits 4, 5 and 6 corresponds to the interrupt control flip-flop circuit output, before input \overline{E} goes low.

An interrupt coming during a read cycle of the STATUS register does not appear in bits 4, 5 and 6 during this read sequence, but during the following one. However, it may appear in bits 0, 1, 2 or on pin \overline{IRQ} .

TABLE 1 – REGISTER ADDRESS

ADDRESS REGISTER					REGISTER FUNCTIONS		Number of bits
Binary				Hexa	Read R/W = 1	Write R/W = 0	
A3	A2	A1	A0				
0	0	0	0	0	STATUS	CMD	8
0	0	0	1	1	CTRL 1 (Write control and interrupt control)		7
0	0	1	0	2	CTRL 2 (Vector and symbol type control)		4
0	0	1	1	3	CSIZE (Character size)		8
0	1	0	0	4	Reserved		—
0	1	0	1	5	DELTA X		8
0	1	1	0	6	Reserved		—
0	1	1	1	7	DELTA Y		8
1	0	0	0	8	X MSBs		4
1	0	0	1	9	X LSBs		8
1	0	1	0	A	Y MSBs		4
1	0	1	1	B	Y LSBs		8
1	1	0	0	C	XLP (Light-pen)	Reserved	7
1	1	0	1	D	YLP (Light-pen)	Reserved	8
1	1	1	0	E	Reserved		—
1	1	1	1	F	Reserved		—

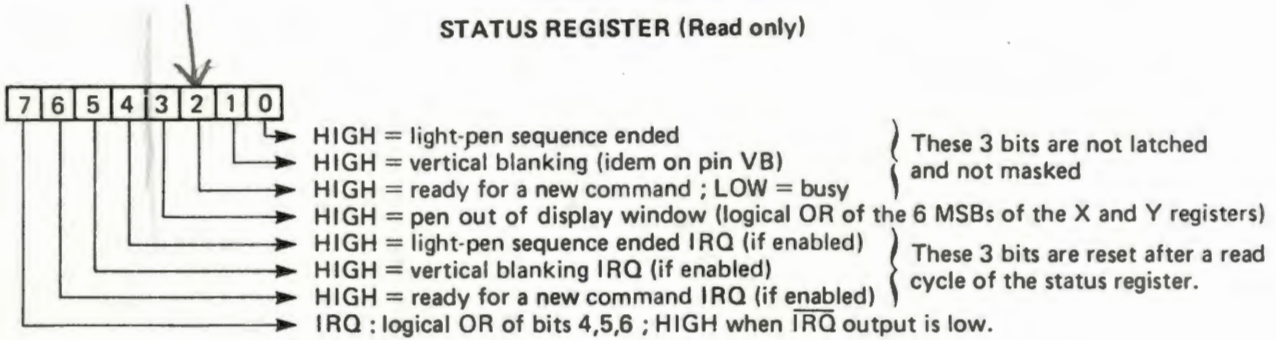
Reserved : These addresses are reserved for future versions of the circuit. In read mode, output buffers D0-D7 force a high state on the data bus.

TABLE 2 – COMMAND REGISTER

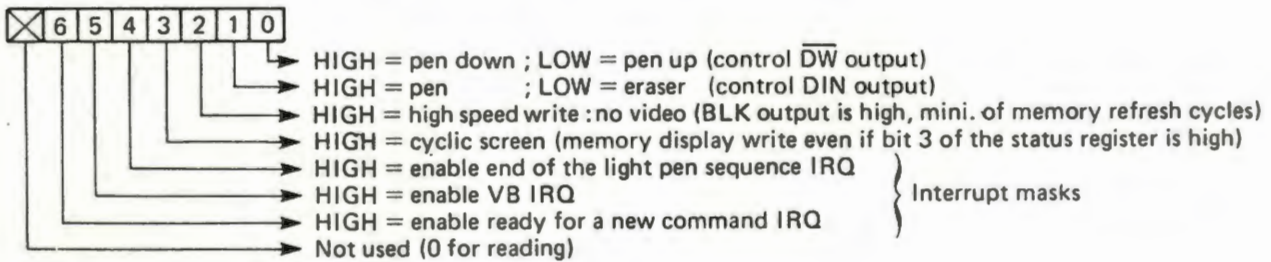
b7 b6 b5 b4	0							1																										
	b3	b2	b1	b0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F														
0 0 0 0	0	Set bit 1 of CTRL 1 : Pen selection		Vector generation (for b2, b1, b0 see small vector definition)	SPACE	0	@	P	·	p	SMALL VECTOR DEFINITION :																							
0 0 0 1	1	Clear bit 1 of CTRL 1 : Eraser selection			!	1	A	Q	a	q	<table border="1"> <tr><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr> <tr><td>1</td><td> ΔX </td><td> ΔY </td><td colspan="4">Direction</td></tr> </table>									b7	b6	b5	b4	b3	b2	b1	b0	1	ΔX	ΔY	Direction			
b7	b6	b5	b4		b3	b2	b1	b0																										
1	ΔX	ΔY	Direction																															
0 0 1 0	2	Set bit 0 of CTRL 1 : Pen/Eraser down selection			"	2	B	R	b	·	r	Dimension																						
0 0 1 1	3	Clear bit 0 of CTRL 1 : Pen/Eraser up selection			#	3	C	S	c	s	<table border="1"> <tr><td>ΔX or ΔY</td><td>Vector length</td></tr> <tr><td>0 0</td><td>0 step</td></tr> <tr><td>0 1</td><td>1 step</td></tr> <tr><td>1 0</td><td>2 steps</td></tr> <tr><td>1 1</td><td>3 steps</td></tr> </table>									ΔX or ΔY	Vector length	0 0	0 step	0 1	1 step	1 0	2 steps	1 1	3 steps					
ΔX or ΔY	Vector length																																	
0 0	0 step																																	
0 1	1 step																																	
1 0	2 steps																																	
1 1	3 steps																																	
0 1 0 0	4	Clear screen		\$	4	D	T	d	t	Direction																								
0 1 0 1	5	X and Y registers reset to 0		%	5	E	U	e	u																									
0 1 1 0	6	X and Y reset to 0 and clear screen		&	6	F	V	f	v																									
0 1 1 1	7	Clear screen, set CSIZE to code "minsize" All other registers reset to 0 (except XLP, YLP)		·	7	G	W	g	w																									
1 0 0 0	8	Light-pen initialization (WHITE forced low)		(8	H	X	h	x																									
1 0 0 1	9	Light-pen initialization)	9	I	Y	i	y																									
1 0 1 0	A	5 x 8 block drawing (size according to CSIZE)		*	:	J	Z	j	z																									
1 0 1 1	B	4 x 4 block drawing (size according to CSIZE)		+	;	K	[k	{																									
1 1 0 0	C	Screen scanning : Pen or Eraser as defined by CTRL1		,	<	L	\	l	!																									
1 1 0 1	D	X register reset to 0		-	=	M]	m	}																									
1 1 1 0	E	Y register reset to 0		.	>	N	↑	n	┘																									
1 1 1 1	F	Direct image memory access request for the next free cycle.		/	?	O	←	o	⊗																									

OTHER REGISTERS

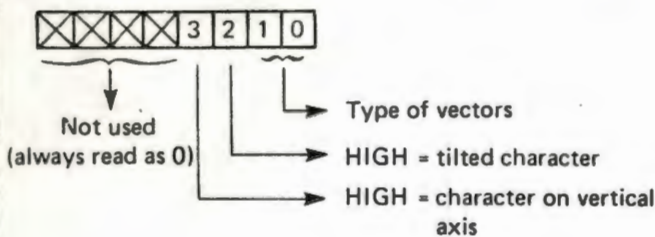
STATUS REGISTER (Read only)



CONTROL REGISTER 1 (Read/Write)



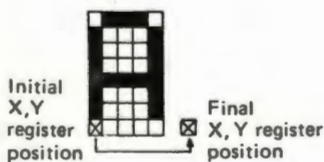
CONTROL REGISTER 2 (Read/Write)



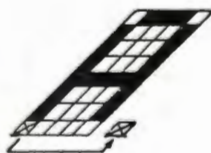
b1	b0	Type of vectors
0	0	———— continuous
0	1	----- dotted
1	0	----- dashed
1	1	----- dotted-dashed

2 dots on, 2 dots off
4 dots on, 4 dots off
10 dots on, 2 dots off,
2 dots on, 2 dots off.

Types of character orientations



$b_3 = 0, b_2 = 0$



$b_3 = 0, b_2 = 1$

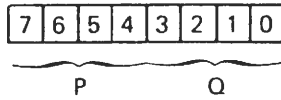


$b_3 = 1, b_2 = 0$



$b_3 = 1, b_2 = 1$

C-SIZE REGISTER (Read/Write)



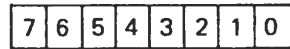
P : Scaling factor on X axis
 Q : Scaling factor on Y axis

P and Q may take any value between 1 and 16. This value is given by the leftmost or rightmost 4 bits for P and Q respectively. Binary value (0) means 16.

X AND Y REGISTERS (Read/Write)



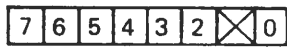
MSBs



LSBs

The 4 leftmost MSBs are always 0.

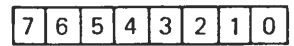
XLP and YLP REGISTERS



Status bit indicating if a rising edge has been applied on LPCK during the first complete frame following light-pen initialization. This bit is reset by a read on XLP or YLP.

always 0

6 bit XLP value



8 bit YLP value

ASCII CHARACTER GENERATOR (5 x 8 matrix)

b7	0	0	0	0	0	0
b6	0	0	1	1	1	1
b5	1	1	0	0	1	1
b4	0	1	0	1	0	1

b3	b2	b1	b0						
0	0	0	0						
0	0	0	1						
0	0	1	0						
0	0	1	1						
0	1	0	0						
0	1	0	1						
0	1	1	0						
0	1	1	1						
1	0	0	0						
1	0	0	1						
1	0	1	0						
1	0	1	1						
1	1	0	0						
1	1	0	1						
1	1	1	0						
1	1	1	1						

BRUGERVEJLEDNING FOR MPS-24 GRAFIKMODUL

JUNI.84/ULJ

Appendix 2

PROGRAMEKSEMPLER

```
0010 // EKSEMPEL 1 - TEGNING AF PARALLELOGRAM
0020 //           ALENE MED ANVENDELSE AF KOMMANDO 17.
0030 //
0040 // Der tegnes et parallelogram med hjørnerne:
0050 // A: (100,100), B: (100,200), C: (200,300), D: (200,200)
0060 // Siderne tegnes i rækkefølgen: A-B, B-C, A-D, D-C
0070 // Efter udstedelse af en kommando kaldes proc. STATUS,
0080 // som afventer GP klar til ny kommando.
0090 // Efter tegning af hele figuren ventes på indtast-
0100 // ning af en karakter fra tastaturet, før der re-
0110 // turneres til video-billede.
0120 //
0130 OUT 176, 255 //           Valg af grafik
0140 OUT 160, 7 //           Slet skærm, nulstil X og Y m.v.
0150 EXEC STATUS
0160 OUT 161, 11 //          Vælg PEN NEDE CYKLISK skærm
0170 OUT 169, 100 //         Sæt X til 100
0180 OUT 171, 100 //         Sæt Y til 100
0190 OUT 167, 100 //         Sæt DELTAY til 100
0200 OUT 160, 17 //         Tegn siden A-B
0210 EXEC STATUS
0220 OUT 171, 200 //         Sæt Y til 200 - X uændret
0230 OUT 165, 100 //         Sæt DELTAX til 100
0240 OUT 160, 17 //         Tegn siden B-C
0250 EXEC STATUS
0260 OUT 168, 0 //           Sæt X til 100
0270 OUT 169, 100
0280 OUT 170, 0 //           Sæt Y til 100
0290 OUT 171, 100
0300 OUT 160, 17 //         Tegn siden A-D
0310 EXEC STATUS
0320 OUT 169, 200 //         Sæt X til 200
0330 OUT 171, 200 //         Sæt Y til 200
0340 OUT 165, 0 //           Sæt DELTAX til 0
0350 OUT 160, 17 //         Tegn siden D-C
0360 // Herefter ventes på tast af vilkårlig karakter
0370 POKE 256, 195
0380 REPEAT
0390 UNTIL PEEK(256)<>195
0400 OUT 176, 0 //           Valg af video
0410 END
0420 //
0430 PROC STATUS
0440   REPEAT
0450   UNTIL (INP(160) MOD 8) DIV 4=1
0460 ENDPROC STATUS
```



```

0010 // EKSEMPEL 2 - TEGNING AF PARALLELOGRAM
0020 //           MED ANVENDELSE AF ALLE KOMMANDOER
0030 //           OG UDNYTTELSE AF OPDATERING AF X OG Y.
0040 //
0050 // Der tegnes et parallelogram med hjørnerne:
0060 // A: (100,100), B: (100,200), C: (200,300), D: (200,200)
0070 // Siderne tegnes i rækkefølgen: A-B, B-C, C-D, D-A
0080 // Efter udstedelse af en kommando kaldes proc. STATUS,
0090 // som afventer GP klar til ny kommando.
0100 // Efter tegning af hele figuren ventes på indtast-
0110 // ning af en karakter fra tastaturet, før der re-
0120 // turneres til video-billede.
0130 //
0140 OUT 176, 255 //           Valg af grafik
0150 OUT 160, 7 //           Slet skærm, nulstil X og Y m.v.
0160 EXEC STATUS
0170 OUT 161, 11 //          Vælg PEN NEDE CYKLISK skærm
0180 OUT 169, 100 //         Sæt X til 100
0190 OUT 171, 100 //         Sæt Y til 100
0200 OUT 167, 100 //         Sæt DELTAY til 100
0210 OUT 160, 17 //         Tegn siden A-B
0220 EXEC STATUS
0230 OUT 165, 100 //         Sæt DELTAX til 100
0240 OUT 160, 17 //         Tegn siden B-C
0250 EXEC STATUS
0260 OUT 165, 0 //           Sæt DELTAX til 0
0270 OUT 160, 21 //         Tegn siden C-D
0280 EXEC STATUS
0290 OUT 165, 100 //         Sæt DELTAX til 100
0300 OUT 160, 23 //         Tegn siden D-A
0310 // Herefter ventes på tast af vilkårlig karakter
0320 POKE 256, 195
0330 REPEAT
0340 UNTIL PEEK(256)<>195
0350 OUT 176, 0 //           Valg af video
0360 END
0370 //
0380 PROC STATUS
0390   REPEAT
0400   UNTIL (INP(160) MOD 8) DIV 4=1
0410 ENDPROC STATUS
    
```

```

0010 // EKSEMPEL 3 - TEGNING AF PARALLELOGRAM
0020 //           MED ANVENDELSE AF ALLE KOMMANDOER
0030 //           OG UDNYTTELSE AF OPDATERING AF X OG Y
0040 //           SAMT IGNORERING AF DELTAX OG DELTAY.
0050 //
0060 // Der tegnes et parallelogram med hjørnerne:
0070 // A: (100,100), B: (100,200), C: (200,300), D: (200,200)
0080 // Siderne tegnes i rækkefølgen: A-B, B-C, C-D, D-A
0090 // Efter udstedelse af en kommando kaldes proc. STATUS,
0100 // som afventer GP klar til ny kommando.
0110 // Efter tegning af hele figuren ventes på indtast-
0120 // ning af en karakter fra tastaturet, før der re-
0130 // turneres til video-billede.
0140 //
0150 // BEMÆRK hastighedsforskellen mellem eksempel 1, 2 og 3.
0160 //
0170 OUT 176, 255 //           Valg af grafik
0180 OUT 160, 7 //           Slet skærm, nulstil X og Y m.v.
0190 EXEC STATUS
0200 OUT 161, 11 //           Vælg PEN NEDE CYKLISK skærm
0210 OUT 169, 100 //           Sæt X til 100
0220 OUT 171, 100 //           Sæt Y til 100
0230 OUT 165, 100 //           Sæt DELTAX til 100
0240 OUT 167, 100 //           Sæt DELTAY til 100
0250 OUT 160, 18 //           Tegn siden A-B
0260 EXEC STATUS
0270 OUT 160, 17 //           Tegn siden B-C
0280 EXEC STATUS
0290 OUT 160, 20 //           Tegn siden C-D
0300 EXEC STATUS
0310 OUT 160, 23 //           Tegn siden D-A
0320 // Herefter ventes på tast af vilkårlig karakter
0330 POKE 256, 195
0340 REPEAT
0350 UNTIL PEEK(256)<>195
0360 OUT 176, 0 //           Valg af video
0370 END
0380 //
0390 PROC STATUS
0400 REPEAT
0410 UNTIL (INP(160) MOD 8) DIV 4=1
0420 ENDPROC STATUS
    
```



```

0010 // EKSEMPEL 4 - TEGNING AF TERNING
0020 //
0030 // I dette program tegnes en terning med skjulte linier
0040 // punkteret.
0050 // I den forreste plan i terningen betegnes hjørnerne:
0060 // A: (50,50), B: (50,200), C: (200,200), D: (200,50)
0070 // Tilsvarende til den bageste plan:
0080 // H: (100,100), I: (100,250), J: (250,250), K: (250,100)
0090 //
0100 OUT 176, 255 //      Valg af grafik
0110 OUT 160, 7 //      nulstil skærm, X, Y m.v.
0120 EXEC STATUS
0130 OUT 161, 11 //      PEN NEDE CYKLISK skærm
0140 OUT 169, 50 //      X:=50
0150 OUT 171, 50 //      Y:=50
0160 OUT 165, 150 //     DELTAX:=150
0170 OUT 167, 150 //     DELTAY:=150
0180 OUT 160, 18 //     Siden A-B
0190 EXEC STATUS
0200 OUT 160, 16 //     Siden B-C
0210 EXEC STATUS
0220 OUT 160, 20 //     Siden C-D
0230 EXEC STATUS
0240 OUT 160, 22 //     Siden D-A
0250 EXEC STATUS
0260 OUT 162, 1 //      Vælg punkteret linie
0270 OUT 169, 250 //     X:=250
0280 OUT 171, 100 //     Y:=100
0290 OUT 160, 22 //     Siden K-H
0300 EXEC STATUS
0310 OUT 160, 18 //     Siden H-I
0320 EXEC STATUS
0330 OUT 162, 0 //      Vælg kontinuert linie
0340 OUT 160, 16 //     Siden I-J
0350 EXEC STATUS
0360 OUT 160, 20 //     Siden J-K
0370 EXEC STATUS
0380 OUT 165, 50 //     DELTAX:=50
0390 OUT 167, 0 //     DELTAY:=0
0400 OUT 160, 31 //     Siden K-D
0410 EXEC STATUS
0420 OUT 169, 200 //     X:=200
0430 OUT 171, 200 //     Y:=200
0440 OUT 160, 25 //     Siden C-J
0450 EXEC STATUS
0460 OUT 169, 50 //     X:=50
0470 OUT 171, 200 //     Y:=200
0480 OUT 160, 25 //     Siden B-I
0490 EXEC STATUS
0500 OUT 162, 1 //      Vælg punkteret linie
0510 OUT 169, 50 //     X:=50
0520 OUT 171, 50 //     Y:=50
0530 OUT 160, 25 //     Siden A-H
0540 // Fortsætter som linie 320 - 420 i eksempel 3.

```

```

0010 // EKSEMPEL 5 - TEGNING AF SINUSKURVE
0020 //
0030 // Kurven tegnes fra 0-360 gr. - 1 pkt. på X-aksen=1 gr.
0040 // I Y-retningen svarer 1 til 250 punkter.
0050 // Koordinatsystemets 0-punkt er (0, 254).
0060 // BEMÆRK at kald af STATUS undgås når det kan udnyttes, at
0070 // Z-80 beregner, medens GP tegner.
0080 //
0090 CMD:=160; CTRL1:=161; CTRL2:=162; DELTAX:=165; DELTAY:=167
0100 XREG1:=168; XREG2:=169; YREG1:=170; YREG2:=171
0110 OUT 176, 255 //      Vælg grafik
0120 OUT CMD, 7 //      Reset
0130 EXEC STATUS
0140 OUT CTRL1, 11
0150 // Tegn X-akse 3 streger tyk
0160 OUT DELTAX, 255 //      DELTAX max. linielængde
0170 FOR I:=0 TO 2 DO
0180     OUT YREG2, 253+I //Sæt Y
0190     OUT CMD, 16 //      1. halvdel
0200     EXEC STATUS
0210     OUT CMD, 16 //      2. halvdel
0220     EXEC STATUS
0230     OUT CMD, 13 //      Reset X
0240 NEXT I
0250 // Tegn Y-akse
0260 OUT CMD, 5 //      (X, Y) = (0, 0)
0270 EXEC STATUS
0280 OUT DELTAY, 255 //      Sæt DELTAY
0290 OUT CMD, 18 //      1. halvdel
0300 EXEC STATUS
0310 OUT CMD, 18 //      2. halvdel
0320 OUT CMD, 5 //      Reset X og Y
0330 OUT YREG2, 254 //      Y=0 i koordinatsystem
0340 TOPI:=8*ATN(1) //      Konstanten 2*pi
0350 OUT DELTAX, 1 //      Y beregnes for hvert X
0360 FOR PKT:=1 TO 360 DO
0370     DY:=INT(SIN(PKT*TOPI/360)*250)-(INP(170)*256+INP(171)-254)
0380     OUT DELTAY, ABS(DY) // Sæt DELTAY
0390     OUT CMD, 17+(SGN(DY)<0)*4 // Tegn kurvesegment
0400 NEXT PKT
0410 // Herefter ventes på en karakter fra tastaturet
0420 POKE 256, 195
0430 REPEAT
0440 UNTIL PEEK(256)<>195
0450 OUT 176, 0 //      Vælg video
0460 //
0470 PROC STATUS
0480     REPEAT
0490     UNTIL (INP(CMD) MOD 8) DIV 4=1
0500 ENDPROC STATUS

```



```
0010 // EKSEMPEL 6 - SKRIVNING AF TEKSTSTRENG
0020 //          VARIABEL KARAKTERSTØRRELSE
0030 //
0040 DIM TXT$ OF 11
0050 CMD:=160; CTRL1:=161; CTRL2:=162; DELTAX:=165; DELTAY:=167
0060 XREG1:=168; XREG2:=169; YREG1:=170; YREG2:=171; CSIZE:=163
0070 TXT$:="COMETGRAFİK"
0080 OUT 176, 255 //          Væg grafik
0090 OUT CMD, 7 //          Reset
0100 EXEC STATUS
0110 OUT CTRL1, 11
0120 OUT YREG2, 200 //          Sæt Y
0130 FOR I:=2 TO 12 DO
0140   OUT CSIZE, I*16+I //          Sæt karakterstørrelse
0150   OUT CMD, ORD(TXT$(I-1)) // skriv karakter
0160 NEXT I
0170 // Herefter ventes på en karakter fra tastaturet
0180 POKE 256, 195
0190 REPEAT
0200 UNTIL PEEK(256)<>195
0210 OUT 176, 0 //          Væg video
0220 //
0230 PROC STATUS
0240   REPEAT
0250   UNTIL (INP(CMD) MOD 8) DIV 4=1
0260 ENDPROC STATUS
```

```
0010 // EKSEMPEL 7 - BLINKENDE TEKST M. BRUG AF SLETTE.
0020 //
0030 // Programmet afsluttes, når vilkårlig karakter indtastes.
0040 //
0050 DIM TXT$ OF 11
0060 CMD:=160; CTRL1:=161; CTRL2:=162; DELTAX:=165; DELTAY:=167
0070 XREG1:=168; XREG2:=169; YREG1:=170; YREG2:=171; CSIZE:=163
0080 TXT$:="COMETGRAFIK"
0090 OUT 176, 255 //      Vælg grafik
0100 OUT CMD, 7 //      Reset
0110 EXEC STATUS
0120 OUT CSIZE, 120 //    Karakterstørrelse 5 x 6
0130 POKE 256, 195
0140 REPEAT
0150   OUT CTRL1, 11 //    Vælg PEN NEDE
0160   OUT CMD, 13 //     Reset X
0170   OUT YREG2, 200 //  Sæt Y
0180   FOR I:=1 TO 11 DO // Skriv tekst
0190     OUT CMD, ORD(TXT$(I))
0200   NEXT I
0210   EXEC TIME //      Tidsforsinkelse
0220   OUT CMD, 13 //     Reset X
0230   OUT CTRL1, 9 //    Vælg SLETTE NEDE
0240   OUT YREG2, 200
0250   FOR I:=1 TO 11 DO
0260     OUT CMD, 10 //    Slet karakter
0270   NEXT I
0280   EXEC TIME //      Tidsforsinkelse
0290 UNTIL PEEK(256)<>195
0300 OUT 176, 0 //      Vælg video
0310 //
0320 PROC STATUS
0330   REPEAT
0340     UNTIL (INP(CMD) MOD 8) DIV 4=1
0350   ENDPROC STATUS
0360 //
0370 PROC TIME
0380   FOR TID:=1 TO 500 DO
0390     NEXT TID
0400 ENDPROC TIME
```



```
0010 // EKSEMPEL 8 - BLINKENDE TEKST M. BRUG READ-MODIFY-WRITE.
0020 //
0030 // Programmet afsluttes, når vilkårlig karakter indtastes.
0040 //
0050 DIM TXT$ OF 11
0060 CMD:=160; CTRL1:=161; CTRL2:=162; DELTAX:=165; DELTAY:=167
0070 XREG1:=168; XREG2:=169; YREG1:=170; YREG2:=171; CSIZE:=163
0080 TXT$:="COMETGRAFIK"
0090 OUT 176, 255 //      Vælg grafik
0100 OUT CMD, 7 //      Reset
0110 EXEC STATUS
0120 OUT CTRL1, 11 //      Vælg PEN NEDE
0130 OUT CSIZE, 120 //      Karakterstørrelse 5 x 6
0140 POKE 256, 195
0150 OUT 177, 255 //      VALG AF R-M-W ON
0160 REPEAT
0170   OUT CMD, 13 //      Reset X
0180   OUT YREG2, 200 //      Sæt Y
0190   FOR I:=1 TO 11 DO //      Skriv tekst
0200     OUT CMD, ORD(TXT$(I))
0210   NEXT I
0220   EXEC TIME //      Tidsforsinkelse
0230 UNTIL PEEK(256)<>195
0240 OUT 176, 0 //      Vælg video
0250 //
0260 PROC STATUS
0270   REPEAT
0280     UNTIL (INP(CMD) MOD 8) DIV 4=1
0290   ENDPROC STATUS
0300 //
0310 PROC TIME
0320   FOR TID:=1 TO 500 DO
0330     NEXT TID
0340   ENDPROC TIME
```

COMGRAPH

EN SAMLING AF PROCEDURER
TIL BRUG SAMMEN MED COMET
GRAFIKMODULET MPS-24

Copyright (c) ICL A/S november 1982.

Denne dokumentation og tilhørende programmel stilles af ICL A/S til brugerens rådighed i forbindelse med levering af grafikprocessor MPS-24 til COMET microprocessoren.

Dokumentation og programmel må anvendes i forbindelse med den leverede grafikprocessor. Det er tilladt at foretage sikkerhedskopiering af programmet i fornødent omfang samt at anvende programmet i programmer til eget brug.

Enhver anden form for anvendelse af dokumentation og programmel - herunder kommerciel udnyttelse af dette samt distribution til tredje part - er ikke tilladt.

ICL A/S påtager sig ingen forpligtigelser med hensyn til fejlrettelser og/eller videreudvikling/opdatering af det leverede programmel samt medfølgende dokumentation.

Ophavsretten til dokumentationen og tilhørende programmel forbliver hos ICL A/S.

GRUPPE 1: INITIERINGER m.v.

INITGRAFIK	- initiering og reset af GP	5
GRAFIK	- valg af grafiskskærm	5
VIDEO	- valg af videoskærm	5
TIME	- tidsforsinkelse	5
WAIT	- vent på inddata fra tastatur	5
INITSTATUS	- initiering af STATUS-rutine	5

GRUPPE 2: FIGURER OG TEKST (fuldstændige specifikationer)

DISPLINEALL	- ret linie	6
DISPREKTALL	- rektangel	6
DISPCIRKELALL	- cirkel	6
DISPCHRALL	- tekst excl. danske kar.	7
DISPCHRALLDK	- tekst incl. danske kar.	7
FILL	- udfyldning	7

GRUPPE 3: FIGURER OG TEKST (gældende specifikationer)

DISPLINE	- ret linie	8
DISPVEKTOR	- vektor	8
DISPREKT	- rektangel	8
DISPCIRKEL	- cirkel	8
DISPCHR	- tekst excl. danske kar.	9
DISPCHRDK	- tekst incl. danske kar.	9

GRUPPE 4: SKRIVNING TIL REGISTRE

SETXY	- skriv X- og Y-register	10
SETX	- skriv X-register	10
SETY	- skriv Y-register	10
SETDELTAXY	- skriv DELTAX- og DELTAY-register	10
SETDELTAX	- skriv DELTAX-register	10
SETDELTAY	- skriv DELTAY-register	10
SETCTRL2	- skriv CTRL2-register	10
SETLINETYPE	- skriv linietype	10
SETCHRTYPE	- skriv karaktertype	10
SETCHRSIZEXY	- skriv karakterstørrelse (X og Y)	11
SETCHRSEX	- skriv karakterstørrelse (X)	11
SETCHRSEY	- skriv karakterstørrelse (Y)	11
SETCTRL1	- skriv CTRL1-register	11
SETPENSLETTE	- skriv valg af PEN/SLETTE	11
SETUPDOWN	- skriv valg af OPPE/NEDE	11
SETSKÆRM	- skriv valg af skærmtypen	11
SETRMW	- skriv valg af RMW	12
CLEARSCREEN	- slet skærm	12
RESETXY	- nulstil X- og Y-register	12
RESETX	- nulstil X-register	12
RESETY	- nulstil Y-register	12
RESETALL	- slet skærm og nulstil registre	12
COMMANDO	- skriv kommando	12

GRUPPE 5: INDLÆSNING FRA REGISTRE

INPX	- læs X-register	12
INPY	- læs Y-register	12
INPDELTA	- læs DELTAX-register	13
INPDELTAY	- læs DELTAY-register	13
INPCTRL1	- læs CTRL1-register	13
INPCTRL2	- læs CTRL2 register	13
INPLINETYPE	- læs linietype	13
INPCHRTYPE	- læs karaktertype	14
INPCHRSIZE	- læs karakterstørrelse	14
INPPENSLET	- læs valg af PEN/SLETTE	14
INPUDDOWN	- læs valg af OPPE/NEDE	14
INPSKÆRM	- læs valg af skærm	14
LÆSSKÆRMALL	- læs punkter på skærm	14
LÆSSKÆRM	- læs punkter på skærm	14

GRUPPE 6: REGISTERVÆRDIER VIA FUNKTIONER

FNPORT	- værdi på spec. port	15
FNX	- værdi af X-register	15
FNY	- værdi af Y-register	15
FNDELTA	- værdi af DELTAX-register	15
FNDELTAY	- værdi af DELTAY-register	15
FNLINETYPE	- gældende linietype	15
FNCHRTYPE	- gældende karaktertype	15
FNCHRSIZE	- karakterstørrelse - X-retning	15
FNCHRSIZEY	- karakterstørrelse - Y-retning	16
FNPPENSLET	- gældende PEN/SLETTE funktion	16
FNUPDOWN	- gældende OPPE/NEDE funktion	16
FNSKÆRM	- gældende skærmtypen	16
FNLÆSSKÆRMALL	- tilstand af punkter på skærm	16
FNLÆSSKÆRM	- tilstand af punkter på skærm	16

COMGRAPH

COMGRAPH er en samling af COMAL-80 procedurer til anvendelse i forbindelse med COMET'ens grafikmodul MPS-24. Procedurerne udfører elementære funktioner som skrivning til og læsning fra registre i grafikprocessoren samt en række sammensatte funktioner så som tegning af linier og figurer.

COMGRAPH består af ca. 60 procedurer (og funktioner) som er opdelt i ialt 7 undergrupper efter kompleksitet og funktion:

GRUPPE 1: Grundlæggende initierings- og styrefunktioner.

GRUPPE 2: Procedurer til tegning af linier, rektangler og cirkler med valgt linietyper og -tykkelse. Endvidere skrivning af tekst med valgt karaktersytype og -størrelse samt med eller uden nationale danske karakterer.

GRUPPE 3: Som gruppe 2, men blot uden mulighed for valg af type, tykkelse og størrelse for linier og karakterer.

GRUPPE 4: Skrivning af værdier til registre eller dele af registre i grafikprocessoren.

GRUPPE 5: Procedurer til indlæsning af værdier fra registre/dele af registre i grafikprocessoren.

GRUPPE 6: Funktioner til samme opgaver som gruppe 5.

GRUPPE 7: Procedurer til behandling af danske nationale karakterer Æ, Ø, Å, æ, ø & å. Kaldes IKKE direkte fra brugerprogrammer.

COMGRAPH procedurerne er lagret i en '.CML' fil for direkte indkorporering brugerprogrammer.

Sammen med COMGRAPH følger et program (GRAFLINK) ved hjælp af hvilket man kan udvælge og indkorporere i sit program netop de procedurer ud af de mange i COMGRAPH, der bruges i det aktuelle program.

I det følgende findes en detaljeret beskrivelse af hver enkelt procedure/funktion i COMGRAPH, men generelt kan nævnes følgende:

- I COMGRAPH findes nødvendige faciliteter for test af grafikprocessorens status. Disse test er indbygget i enkelte af procedurerne. Det er herudover brugerens ansvar at foretage de nødvendige test af status i brugerens egen del af programmet.
- Procedurerne i COMGRAPH indeholder kun i meget begrænset omfang test af validiteten af data, der overføres til procedurerne som formelle parametre. Denne form for test må i fornødent omfang ske i den brugerskrevne del af programmet.
- Alle procedurer i COMGRAPH, som gør brug af variable, er lukkede (CLOSED) procedurer, således at der ikke forekommer sammenfald af variabelnavne.

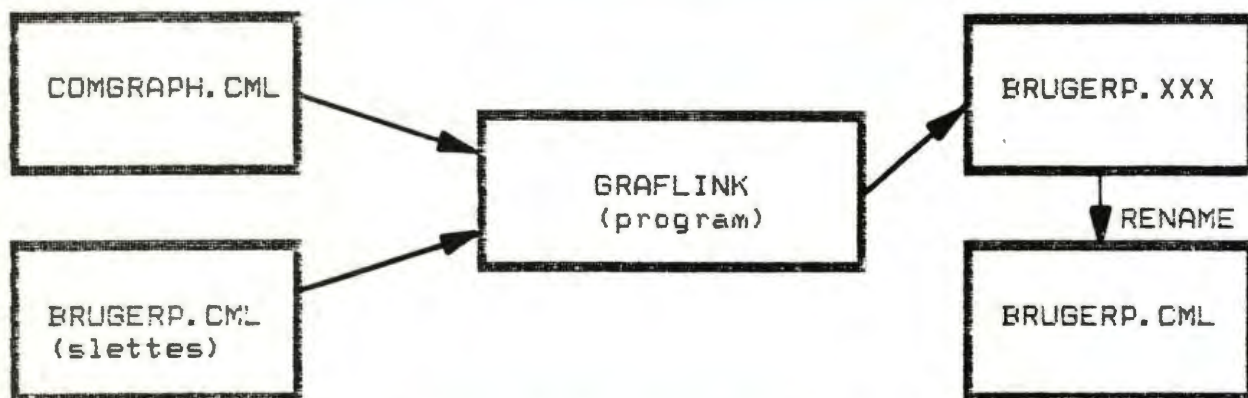
Procedurerne i COMGRAPH optager linienumrene fra 9000 og frem-
efter. Undgå disse linienumre i programmer, der anvender COMGRAPH.

COMGRAPH procedurerne/funktionerne kan indkorporeres i et bruger-
skrevet COMAL-80 program ved:

ENTER COMGRAPH

i hvilket tilfælde samtlige procedurer/funktioner, som fylder om-
kring 12 Kb, bliver indeholdt i programmet.

Da der i almindelighed er tale om at anvende en delmængde af pro-
cedurerne i COMGRAPH i et program, vil det som oftest være mere
hensigtsmæssigt at anvende programmet GRAFLINK til at udvælge og
indkorporere de ønskede rutiner. GRAFLINK arbejder således:



Fremgangsmåden ved udvikling af programmer bliver herefter:

1. Brugeren indtaster sit program på sædvanlig vis og kalder heri procedurer fra COMGRAPH efter behov.
2. Brugeren lagrer sit program på en diskette i en '.CML' fil.
3. En diskette med COMGRAPH.CML skal være indsat i COMET'en, ligesom GRAFLINK skal være tilgængelig. Endvidere skal der på den/de monterede diskette(r) være plads til en arbejdsfil noget større end brugerprogrammet.
4. LOAD GRAFLINK
RUN
5. Indtast svar på de af GRAFLINK stillede spørgsmål.
BEMÆRK: De anvendte filer kan placeres vilkårligt på en eller to disketter.
6. GRAFLINK vil nu:
 - a. oprette en arbejdsfil (BUGERP.XXX) med det samme navn som BRUGERP.CML, men med typebetegnelsen '.XXX'.
 - b. overføre indholdet af BRUGERP.CML til arbejdsfilen samtidig med at der udtrækkes informationer om kald af procedurer samt indbyggede procedurer i BRUGERP.CML.
 - c. bestemme, hvilke procedurer, der er eksterne, dvs. ikke er indbygget i BRUGERP.CML.
 - d. ved gennemlæsning af COMGRAPH.CML finde og overføre til arbejdsfilen eksterne procedurer. Herunder registreres eventuelle procedurekald, der findes i de overførte procedurer.
 - e. d. gentages indtil samtlige eksterne procedurer er overført, eller indtil det konstateres, at en given procedure hverken er at finde i BRUGERP.CML eller COMGRAPH.CML. Det sidste resulterer i en fejlmeddelelse.

9. - lukke samtlige filer samt slette BRUGERP.CML og ændre typebetegnelse for arbejdsfilen BRUGERP.XXX til '.CML'.

BEMÆRK: GRAFLINK indkorporerer IKKE funktioner fra COMGRAPH.

Foretager man ændringer i sit program, som kræver yderligere procedurer fra COMGRAPH, kan ovennævnte fremgangsmåde gentages, idet allerede indkorporerede COMGRAPH-procedurer ikke genindlæses.

Der er i COMGRAPH afsat plads til at behandle programmet med op til 50 kald af FORSKELLIGE procedurer samt 50 indbyggede procedurer. Disse tal kan let udvides ved at rette værditildelingen for variabelen MAXPROC forrest i programmet.

EKSEMPEL PÅ BRUG AF COMGRAPH

Som et eksempel på anvendelse af COMGRAPH-procedurerne vises på næste side hovedprogrammet af et program, der tegner følgende på grafiskskærmen:

1. Et kvadrat (rektangel), der indrammer skærbilledet.
Linietype: kontinuert.
Linietykkelse: 3.
2. Et kvadrat, hvis vinkelspidser falder på midtpunkterne af siderne i det under 1 tegnede kvadrat.
Linietype: punkteret.
Linietykkelse: 1.
3. En cirkel indskrevet i kvadratet fra 2.
Linietype: kontinuert.
Linietykkelse: 3.
4. To diagonaler i ovennævnte cirkel, som forbinder to og to cirkelens berøringspunkter med rektanlet fra 2.
5. Tekststrengene "COMAL-80", "COMET", "GRAFIK" og "COMGRAPH" skrevet i hver sit af de fire under 4 fremkomne 'lagkagesnit'
Karaktertype: lodret på vandret linie.
Karakterstørrelse: i X-retning: *3
 i Y-retning: *4

```

0010 // PROGRAM COMEX                NOV.82 / ULJ
0020 //
0030 // Eksempel på anvendelse af COMGRAPH-procedurer.
0040 //
0050 EXEC INITSTATUS
0060 EXEC GRAFIK(1) // Skift til grafiskskærm.
0070 EXEC CLEARSCREEN // Sletning af grafiskskærm.
0080 //
0090 // Tegning af ramme.
0100 //
0110 EXEC DISPREKTALL(0,3,1,1,510,510)
0120 //
0130 // Tegning af rhombe startende i nederste hjørne
0140 // Tegning foregår mod urets retning.
0150 //
0160 EXEC DISPLINEALL(1,1,255,3,508,255)
0170 // BEMÆRK: efter at have sat linietype ovenfor
0180 //          forbliver denne type i kraft.
0190 EXEC DISPVEKTOR(255,508)
0200 EXEC DISPVEKTOR(3,255)
0210 EXEC DISPVEKTOR(255,3)
0220 //
0230 // Tegning af cirkel
0240 //
0250 EXEC DISPCIRKELALL(0,3,255,255,381,381)
0260 EXEC DISPLINEALL(0,3,129,129,381,381)
0270 EXEC DISPLINEALL(0,3,129,381,381,129)
0280 //
0290 // Skrivning af tekster
0300 //
0310 EXEC DISPCHRALL(0,3,4,180,365,"COMAL-80")
0320 EXEC DISPCHR(110,241,"COMET")
0330 EXEC DISPCHR(295,241,"GRAFIK")
0340 EXEC DISPCHR(180,113,"COMGRAPH")
0350 //
0360 // Herefter ventes på indtastning af en vilkårlig
0370 // karakter fra tastaturet, før der fortsættes.
0380 //
0390 EXEC WAIT
0400 //
0410 // Ved fortsættelse returneres til VIDE0.
0420 //
0430 EXEC VIDE0
0440 END
0450 //
0460 // Herefter følger procedurer fra COMGRAPH

```


GRUPPE 1: INITIERINGER

I denne gruppe findes rutiner til valg mellem VIDEO og GRAFIK samt til initiering af rutine til aflæsning af grafikprocessorens status. Endvidere to 'venterutiner'.

INITGRAFIK

FORMAL: Skift til grafik. Sletning af grafikskærm. Valg af PEN - NEDE - CYKLISK skærm. Nulstilling af øvrige registre.
KALD: EXEC INITGRAFIK(SKRMNR)
SKRMNR: Grafikskærm (0 el. 1), der skal initieres.

GRAFIK

FORMAL: Skift til grafikskærm.
KALD: EXEC GRAFIK(SKRMNR)
SKRMNR: Grafikskærm (0 el. 1), der skiftes til.

VIDEO

FORMAL: Skift til videobillede.
KALD: EXEC VIDEO

TIME

FORMAL: Tidsforsinkelse.
KALD: EXEC TIME(SEC)
SEC: Længden af tidsforsinkelse i sekunder.
FUNKTION: Der returneres fra proceduren efter SEC sekunder.

WAIT

FORMAL: Vente på inddata fra tastatur. Indlæse en karakter fra tastatur.
KALD: EXEC WAIT(KAR)
KAR: Indeholder ved returnering fra WAIT en fra tastaturet indtastet karakter (incl. ESC).
FUNKTION: Efter kaldet venter der i WAIT indtil en karakter indtastes fra tastaturet. Samtlige karakterer incl. ESC, CTRL-, RETURN m.v. vil aktivere WAIT.

INITSTATUS

FORMAL: Initiering af rutine til aflæsning af GP's status.
KALD: EXEC INITSTATUS
FUNKTION: Proceduren må KUN KALDES EN GANG pr. programafvikling. Der oprettes maskinkode, som kan kaldes ved:
CALL STATUS
Efter dette kald returneres først til videre programudførelse, når GP er klar til at modtage ny kommando. Navnet STATUS må i øvrigt IKKE anvendes i programmer, der anvender EXEC INITSTATUS og CALL STATUS. Proceduren kalder STATUSINIT.

GRUPPE 2: FIGURER OG TEKST MED FULDSTÆNDIGE SPECIFIKATIONER

Denne gruppe omfatter procedurer til tegning af rette linier, rektangler og cirkler samt udfyldning af rektangulære områder og skrivning af tekststrengene.

Parameterlisterne til procedurerne omfatter fuld specifikation af linie- og karaktertype samt linietykkelse og karakterstørrelse.

DISPLINEALL

FORMAL: Tegning af rette linier med specifikation af linietype og -tykkelse.

KALD: EXEC DISPLINEALL(TY, BR, X1, Y1, X2, Y2)

TY: Linietype: 0 for kontinuert linie
1 for punkteret linie
2 for stiplede linie
3 for stiplede/punkteret linie

BR: Liniebredde i antal punkter (1..n).

X1, Y1: Koordinater (0..511) for liniens startpunkt.

X2, Y2: Koordinater (0..511) for liniens endepunkt.

FUNKTION: Der tegnes BR rette linier symmetrisk omkring og parallelt med linien (X1,Y1) - (X2,Y2).

Proceduren kalder DISPLINE.

DISPREKTALL

FORMAL: Tegning af rektangler med specifikation af linietype og -tykkelse. Vandrette og lodrette sider.

KALD: EXEC DISPREKTALL(TY, BR, X1, Y1, X2, Y2)

TY: Linietype: se DISPLINEALL.

BR: Liniebredde i antal punkter (1..n).

X1, Y1: Koordinater (0..511) for rektanglets startpunkt.

X2, Y2: Koordinater (0..511) for rektanglets hjørne diagonalt modsat startpunktet.

FUNKTION: Der tegnes BR rektangler symmetrisk omkring rektanglet (X1,Y1) - (X2,Y2).

Proceduren kalder DISPREKT.

DISPCIRKELALL

FORMAL: Tegning af cirkel med specifikation af linietype og -bredde.

KALD: EXEC DISPCIRKELALL(TY, BR, X1, Y1, X2, Y2)

TY: Linietype - se DISPLINEALL.

BR: Liniebredde i antal punkter (1..n).

X1, Y1: Koordinater (0..511) for cirkelens centrum.

X2, Y2: Koordinater (0..511) for et punkt på cirkelperiferien.

FUNKTION: Der tegnes BR cirkler symmetrisk omkring cirklen med centrum i (X1,Y1) og radius:

$$R = \text{SQRT}((X2 - X1)^2 + (Y2 - Y1)^2)$$

Proceduren kalder DISPCIRKEL.

DISPCHRALL

FORMAL: Udskrivning af en karakterstreng med specifikation af karaktertype og -størrelse.

KALD: EXEC DISPCHRALL(TY, MX, MY, X1, Y1, S\$)
TY: Karaktertype: 0 for lodret på vandret linie
1 for skrå på vandret linie
2 for lodret på lodret linie
3 for skrå på lodret linie
MX, MY: Multiplikationsfaktor (1..16) for karakterstørrelse i X-retningen hhv. Y-retningen.
X1, Y1: Koordinater (0..511) for start af strengen.
S\$: Karakterstreng, der skal udskrives.

FUNKTION: Karaktererne i strengen udskrives i overensstemmelse med det internationale ASCII tegnsæt, dvs. uden danske nationale karakterer (Æ, Ø, Å, æ, ø, å).
Se også proceduren DISPCHRALLDK.
Proceduren kalder DISPCHR.

DISPCHRALLDK

FORMAL: Som proceduren DISPCHRALL, men med udskrift af danske nationale karakterer Æ, Ø, Å, æ, ø, og å i overensstemmelse med gældende standard.

KALD: EXEC DISPCHRALLDK(TY, MX, MY, X1, Y1, S\$)
Jvfr. proceduren DISPCHRALL.

FUNKTION: Karakterstrengen udskrives i overensstemmelse med dansk standard.
Proceduren kalder DISPCHRDK.

FILL

FORMAL: Udfyldning/sletning af et rektangulært område med vandrette og lodrette sider.

KALD: EXEC FILL(X1, Y1, X2, Y2, PENERASE)
X1, Y1: Koordinater (0..511) for det rektangulære områdes ene hjørnepunkt (startpunkt).
X2, Y2: Koordinater (0..511) for hjørnet diametralt modsat det rektangulære områdes startpunkt.
PENERASE: Valg mellem at udfylde eller slette:
0 for slette
1 for udfylde.

FUNKTION: Det rektangulære område udfyldes med vandrette linier med brug af PEN eller SLETTE funktion i overensstemmelse med PENERASE-parameteren.

GRUPPE 3: FIGURER OG TEKST MED GÆLDENDE SPECIFIKATIONER.

Procedurerne i denne gruppe omfatter tegning af rette linier, rektangler og cirkler med enkelt streg samt med den stregtype, der er gældende ved procedurekaldet.

Endvidere omfatter gruppen procedurer til udskrivning af karakterstrengene (m. og u. danske karakterer) med den karaktertype og -størrelse, der er gældende ved procedurekaldet.

DISPLINE

FORMAL: Tegning af ret linie af et punkts bredde og med den type, der er gældende ved procedurekaldet.

KALD: EXEC DISPLINE(X1, Y1, X2, Y2)
X1, Y1: Koordinater (0..511) for liniens startpunkt.
X2, Y2: Koordinater (0..511) for liniens endepunkt.

FUNKTION: Efter udførelsen af proceduren er X- og Y-registrene opdateret til koordinaterne for liniens endepunkt.
DELTAX- og DELTAY-registrenes indehold er:

DELTAX = ABS(X2 - X1)

DELTAY = ABS(Y2 - Y1)

DISPVEKTOR

FORMAL: Tegning af en ret linie med startpunkt svarende til indholdet af X- og Y-registrene ved kald af proceduren. Liniebredden er et punkt, og linietype svarer til den ved procedurekaldet gældende.

KALD: EXEC DISPVEKTOR(PX, PY)
PX, PY: Koordinater (0..511) for liniens endepunkt.

FUNKTION: Efter udførelse af proceduren er registrene opdateret som for DISPLINE.

DISPREKT

FORMAL: Tegning af et rektangel med en liniebredde på et punkt og linietype svarende til den ved procedurekaldet gældende.

KALD: EXEC DISPREKT(X1, Y1, X2, Y2)
X1, Y1: Koordinater (0..511) for et af rektanglets hjørner.
X2, Y2: Koordinater (0..511) for det diametralt modsatte hjørne i rektanglet.

FUNKTION: Proceduren kalder DISPVEKTOR.
Indholdet af X-, Y-, DELTAX- og DELTAY-registrene er undefineret efter udførelse af proceduren.

DISPCIRKEL

FORMAL: Tegning af en cirkel med en liniebredde på et punkt og linietype svarende til den ved procedurekaldet gældende.

KALD: EXEC DISPCIRKEL(X1, Y1, X2, Y2)
X1, Y1: Koordinater (0..511) for cirkelns centrum.
X2, Y2: Koordinater (0..511) for et punkt på cirkelperiferien.

FUNKTION: Cirkelns radius bliver:
 $R = \text{SQRT}((X2 - X1)^2 + (Y2 - Y1)^2)$

DISPCHR

FORMÅL: Skriver en karakterstreng med den karaktertype og -størrelse, der er gældende ved procedurekaldet.

KALD: EXEC DISPCHR(X1, Y1, S\$)
X1, Y1: Koordinater (0..511) for strengens startpunkt.
S\$: Den karakterstreng, der skal udskrives.

FUNKTION: Karakterstrengen udskrives i overensstemmelse med det internationale ASCII tegnsæt, dvs. uden danske nationale karakterer Æ, Ø, Å, æ, ø og å.
Se også proceduren DISPCHRDK.

DISPCHRDK

FORMÅL: Som proceduren DISPCHR, men med udskrift af de danske nationale karakterer Æ, Ø, Å, æ, ø og å i overensstemmelse med gældende standard.

KALD: EXEC DISPCHRDK(X1, Y1, S\$)
X1, Y1: Koordinater (0..511) for strengens startpunkt.
S\$: Den karakterstreng, der skal udskrives.

FUNKTION: Karakterstrengen udskrives i overensstemmelse med dansk standard.
Proceduren kalder DANSK.

GRUPPE 4: SKRIVNING TIL REGISTRE

I denne gruppe findes procedurer til skrivning til de enkelte registre samt til at sætte dele af registre, hvor dette er relevant.

SETXY

FORMAL: Udskriver værdier til X- og Y-registrene.
KALD: EXEC SETXY(X, Y)
X: Værdien (0..511), der udskrives til X-register.
Y: Værdien (0..511), der udskrives til Y-register.

SETX

FORMAL: Udskriver en værdi til X-registeret.
KALD: EXEC SETX(X)
X: Værdien (0..511), der udskrives til X-register.

SETY

FORMAL: Udskriver en værdi til Y-registeret.
KALD: EXEC SETY(Y)
Y: Værdien (0..511), der udskrives til Y-register.

SETDELTAXY

FORMAL: Udskriver værdier til DELTAX- og DELTAY-registrene.
KALD: EXEC SETDELTAXY(DX, DY)
DX: Værdien (0..255), der udskrives til DELTAX-reg.
DY: Værdien (0..255), der udskrives til DELTAX-reg.

SETDELTAX

FORMAL: Udskriver en værdi til DELTAX-registeret.
KALD: EXEC SETDELTAX(X)
X: Værdien (0..255), der udskrives til DELTAX-reg.

SETDELTAY

FORMAL: Udskriver en værdi til DELTAY-registeret.
KALD: EXEC SETDELTAY(Y)
Y: Værdien (0..255), der udskrives til DELTAY-reg.

SETCTRL2

FORMAL: Udskriver en værdi til CTRL2-registeret.
KALD: EXEC SETCTRL2(V)
V: Værdien (0..15), der udskrives til CTRL2-reg.

SETLINETYPE

FORMAL: Sætter linietype i CTRL2-regitsteret uden i øvrigt at ændre i dette registers indhold.
KALD: EXEC SETLINETYPE(TY)
TY: Linietype: 0 for kontinuert linie
1 for punkteret linie
2 for stiplet linie
3 for stiplet/punkteret linie.

SETCHRTYPE

FORMAL: Sætter karaktertype i CTRL2-registeret uden i øvrigt at ændre indholdet af dette register.

KALD: EXEC SETCHRTYPE(TY)
TY: Karaktertype: 0 for lodret på vandret linie
1 for skrå på vandret linie
2 for lodret på lodret linie
3 for skrå på lodret linie.

SETCHRSIZEXY

FORMAL: Udskriver multiplikationsfaktorer for karakterstørrelse i X- og Y-retning til CSIZE-registeret.

KALD: EXEC SETCHRSIZEXY(X, Y)
X: Multiplikationsfaktor (1..16) for X-retningen.
Y: Multiplikationsfaktor (1..16) for Y-retningen.

SETCHRSIZE X

FORMAL: Udskriver multiplikationsfaktor for karakterstørrelse i X-retningen til CSIZE-registeret.

KALD: EXEC SETCHRSIZE X(X)
X: Multiplikationsfaktor (1..16) for X-retningen.

SETCHRSIZE Y

FORMAL: Udskriver multiplikationsfaktor for karakterstørrelse i Y-retningen til CSIZE-registeret.

KALD: EXEC SETCHRSIZE Y(Y)
Y: Multiplikationsfaktor (1..16).

SETCTRL 1

FORMAL: Udskriver en værdi til CTRL1-registeret.

KALD: EXEC SETCTRL1(V)
V: Værdien (0..127), der udskrives til CTRL1.

SETPENSLETTE

FORMAL: Valg af PEN/SLETTE funktion (CTRL1-registeret).

KALD: EXEC SETPENSLETTE(F)
F: Funktion: 0 for SLETTE-funktion
1 for PEN-funktion.

SETUPDOWN

FORMAL: Valg af pen OPPE/NEDE funktion (CTRL1-registeret).

KALD: EXEC SETUPDOWN(F)
F: Funktion: 0 for pen/slette OPPE
1 for pen/slette NEDE.

SETSKÆRM

FORMAL: Valg mellem FLAD og CYKLISK skærm.

KALD: EXEC SETSKÆRM(S)
S: Skærmtype: 0 for FLAD skærm
1 for CYKLISK skærm.

SETRMW

FORMAL: Valg mellem Read-Modify-Write (RMW) ON eller OFF.
KALD: EXEC SETRMW(F)
F: Funktion: 0 for RMW OFF
1 for RMW ON

CLEARSCREEN

FORMAL: Blankstiller (sletter) den valgte grafiskskærm.
KALD: EXEC CLEARSCREEN

RESETXY

FORMAL: Nulstiller X- og Y-registrene.
KALD: EXEC RESETXY

RESETX

FORMAL: Nulstiller X-registeret.
KALD: EXEC RESETX

RESETY

FORMAL: Nulstiller Y-registeret.
KALD: EXEC RESETY

RESETALL

FORMAL: Blankstiller (sletter) den valgte grafiskskærm samt sætter karakterstørrelse (CSIZE) til minimal karakterstørrelse og nulstiller alle øvrige registre.
KALD: EXEC RESETALL

COMMANDO

FORMAL: Udsteder en kommando til GP og returnerer, når GP er klar til at modtage en ny kommando.
KALD: EXEC COMMANDO(C)
C: Kommando (0..255).
FUNKTION: Proceduren anvender CALL STATUS, hvorfor der før første kald af COMMANDO skal være foretaget et kald af proceduren INITSTATUS.

GRUPPE 5: INDLÆSNING AF REGISTRE

Ved hjælp af procedurerne i denne gruppe indlæses registerværdier eller dele af registerværdier hvor relevant.

HUSK: de variable, der anvendes i forbindelse med indlæsningsprocedurerne skal være erklæret før kald af procedurerne.

INPX

FORMAL: Indlæsning af værdien i det totale X-register.
KALD: EXEC INPX(X1)
X1: Indeholder efter kaldet værdien (0..511) af X-reg.

INPY

FORMAL: Indlæsning af værdien i det totale Y-register.
KALD: EXEC INPY(Y1)
Y1: Indeholder efter kaldet værdien (0..511) af Y-reg.

INPDELTAX

FORMAL: Indlæsning af værdien i DELTAX-registeret.
KALD: EXEC INPDELTAX(DX)
DX: Indeholder efter kaldet værdien (0..255) af DELTAX-registeret.

INPDELTAY

FORMAL: Indlæsning af værdien i DELTAY-registeret.
KALD: EXEC INPDELTAY(DY)
DY: Indeholder efter kaldet værdien (0..255) af DELTAY-registeret.

INPCTRL1

FORMAL: Indlæsning af værdien i CTRL1-registeret.
KALD: EXEC INPCTRL1(CTRL1)
CTRL1: Indeholder efter kaldet værdien (0..127) af CTRL1-registeret.

INPCTRL2

FORMAL: Indlæsning af værdien i CTRL2-registeret.
KALD: EXEC INPCTRL2(CTRL2)
CTRL2: Indeholder efter kaldet værdien (0..15) af CTRL2-registeret.

INPLINETYPE

FORMAL: Indlæsning af gældende linietype.
KALD: EXEC INPLINETYPE(TY)
TY: Indeholder efter kaldet linietype:
0 for kontinuert linie
1 for punkteret linie
2 for stiplet linie
3 for stiplet/punkteret linie

INPCHRTYPE

FORMÅL: Indlæsning af gældende karaktertype.
KALD: EXEC INPCHRTYPE(TY)
TY: Indeholder efter kaldet gældende karaktertype:
0 for lodret på vandret linie
1 for skrå på vandret linie
2 for lodret på lodret linie
3 for skrå på lodret linie

INPCHRSIZE

FORMÅL: Indlæsning af gældende karakterstørrelse i X- og Y-retningen.
KALD: EXEC INPCHRSIZE(MX, MY)
MX: Indeholder efter kaldet multiplikationsfaktoren (1..16) for karakterstørrelsen i X-retningen.
MY: Indeholder efter kaldet multiplikationsfaktoren (1..16) for karakterstørrelsen i Y-retningen.

INPPENSLET

FORMÅL: Indlæsning af valgt PEN eller SLETTE funktion.
KALD: EXEC INPPENSLET(PEN)
PEN: Indeholder efter kaldet: 0 hvis SLETTE valgt
1 hvis PEN valgt.

INPUPDOWN

FORMÅL: Indlæsning af valgt UP eller DOWN for PEN/SLET.
KALD: EXEC INPUPDOWN(UPDOWN)
UPDOWN: Indeholder efter kaldet: 0 hvis UP valgt
1 hvis DOWN valgt.

INPSKÆRM

FORMÅL: Indlæsning af valgt skærmtype (FLAD el. CYKLISK).
KALD: EXEC INPSKÆRM(SKÆRM)
SKÆRM: Indeholder efter kald: 0 hvis FLAD valgt
1 hvis CYKLISK valgt

LÆSSKÆRMALL

FORMÅL: Indlæsning af tilstanden af 8 punkter i vandret retning på skærmen ud fra givne X-Y-koordinater.
KALD: EXEC LÆSSKÆRM(X, Y, RES)
X: X-koordinat for første venstre punkt.
Y: Y-koordinat for vandret linie.
RES: Indeholder efter kaldet et decimalt tal, der repræsenterer de 8 punkters tilstand.
FUNKTION: RES er den decimale værdi af en byte, hvor bit nr. 0 svarer til pkt. (X,Y) og bit nr. 7 svarer til punktet (X+7,Y). De enkelte bit er sat til 1, hvis det tilsvarende punkt er OFF (mørkt), eller 0, hvis ON (lyst).

LÆSSKÆRM

FORMÅL: Indlæsning af tilstanden af 8 punkter i vandret retning på skærmen ud fra gældende X-Y-registre.
KALD: EXEC LÆSSKÆRM(RES)
RES: Indeholder efter kaldet et decimalt tal, der repræsenterer de 8 punkters tilstand.
FUNKTION: Se LÆSSKÆRMALL.

GRUPPE 6: FUNKTIONER FOR REGISTERVÆRDIER

Denne gruppe indeholder funktioner til indlæsning af registerværdier eller dele af registerværdier.

Funktionerne anvendes f.eks. på formerne:

```
XREG:=FNX  
PRINT FNCHRTYPE  
OUT 165,FNDELTA/2 // Halvering af DELTAX
```

FNPORT

FORMAL: Giver indhold af register allokeret til given port.
KALD: FNPORT(NR)
NR: Portnummer.

FNX

FORMAL: Giver værdi af X-registerer.
KALD: FNX

FN Y

FORMAL: Giver værdi af Y-registeret.
KALD: FNY

FNDELTA

FORMAL: Giver værdi af DELTAX-registeret.
KALD: FNDELTA

FNDELTA Y

FORMAL: Giver værdi af DELTAY-registeret.
KALD: FNDELTA Y

FNLINETYPE

FORMAL: Giver aktuelle linietype.
KALD: FNLINETYPE
Efter kald: 0 for kontinuert linie
1 for punkteret linie
2 for stiplet linie
3 for stiplet/punkteret linie

FNCHRTYPE

FORMAL: Giver aktuelle karaktertype.
KALD: FNCHRTYPE
Efter kald: 0 for lodret på vandret linie
1 for skrån på vandret linie
2 for lodret på lodret linie
3 for skrån på lodret linie

FNCHRSEX

FORMAL: Giver aktuel karakterstørrelse (1..16) i X-retningen
KALD: FNCHRSEX

FNCHRSIZEY

FORMAL: Giver aktuel karakterstørrelse (1..16) i Y-retningen.
KALD: FNCHRSIZEY

FNPENSLET

FORMAL: Giver gældende PEN eller SLETTE funktion.
KALD: FNPENSLET
Efter kaldet: 0 hvis SLETTE gældende
1 hvis PEN gældende

FNUPDOWN

FORMAL: Giver gældende pen/slette OPPE eller NEDE.
KALD: FNUPDOWN
Efter kaldet: 0 hvis OPPE gældende
1 hvis NEDE gældende

FNSKÆRM

FORMAL: Giver gældende skærmtype (FLAD el. CYKLISK).
KALD: FNSKÆRM
Efter kaldet: 0 hvis FLAD skærm gældende
1 hvis CYKLISK skærm gældende

FNLÆSSKÆRMALL

FORMAL: Giver tilstanden af 8 punkter i vandret retning på skærmen ud fra givne X-Y-koordinater.
KALD: FNLÆSSKÆRMALL(X, Y)
X: X-koordinat for første venstre punkt.
Y: Y-koordinat for vandret linie.
FUNKTION: Efter kaldet indeholder FNLÆSSKÆRMALL et decimalt tal, der repræsenterer de 8 punkters tilstand.
Jvfr. i øvrigt proceduren LÆSSKÆRMALL.

FNLÆSSKÆRM

FORMAL: Giver tilstanden af 8 punkter i vandret retning på skærmen ud fra gældende X-Y-registre.
KALD: FNLÆSSKÆRM
FUNKTION: Se FNLÆSSKÆRM og LÆSSKÆRMALL.

REGRESSIONS ANALYSE

ET EKSEMPEL PÅ ANVENDELSE AF
MPS-24 GRAFIKMODUL TIL COMET
OG
WATANABE DIGI-PLOT Model WX4671

Copyright (c) ICL A/S november 1982.

Denne dokumentation og tilhørende programmel stilles af ICL A/S til brugerens rådighed i forbindelse med levering af grafikprocessor MPS-24 til COMET microprocessoren.

Dokumentation og programmel må anvendes i forbindelse med den leverede grafikprocessor. Det er tilladt at foretage sikkerhedskopiering af programmet i fornødent omfang samt at anvende programmet i programmer til eget brug.

Enhver anden form for anvendelse af dokumentation og programmel - herunder kommerciel udnyttelse af dette samt distribution til tredje part - er ikke tilladt.

ICL A/S påtager sig ingen forpligtigelser med hensyn til fejlrettelser og/eller videreudvikling/opdatering af det leverede programmel samt medfølgende dokumentation.

Ophavsretten til dokumentationen og tilhørende programmel forbliver hos ICL A/S.

REGRESSIONS ANALYSE

1. PROGRAMNAVN: REGRES.CSB

2. FORMÅL.

Ved hjælp af programmet REGRES kan man foretage en regressionsanalyse af punkter, der plottes ind i et koordinatsystem på skærmen. Regressionsanalysen foretages ved at beregne et 1. eller 2. grads polynomium, som opfylder mindste kvadraters afvigelse. Kurven for det beregnede polynomium indtegnes på skærmen, og polynomiet udskrives.

Der er mulighed for at aflæse kurvens værdier, ligesom man kan fjerne/tilføje punkter og foretage en fornyet analyse. Endelig kan såvel koordinatsystem som plottede punkter og kurver samt formler tegnes på en plotter af typen WATANABE DIGI-PLOT model WX4671 med anvendelse af denne plotters mulighed for at tegne med flere farver.

3. PROGRAMAFVIKLING.

For afvikling af programmet REGRES kræves en COMET med grafikmodul MPS-24.

Program REGRES kan kun afvikles under den swappede COMAL-80 version COMAL80S.

Program REGRES kan indlæses fra en vilkårlig disketteenhed og der anvendes ingen filer ligesom der heller ikke CHAIN'es til andre programmer.

```
LOAD REGRES
RUN
```

4. SKÆRMBILLEDET.

Efter start af REGRES udtegnes et grafikbillede, som består af 4 delområder:

4.1. MENU

I øverste venstre hjørne af skærmen udskrives en menu med valgmulighed mellem 8 funktioner, som er nærmere beskrevet i afsnit 5.

Valg af funktion sker ved at bevæge et lysende felt op eller ned i menuen ved brug af tastaturets 'pil op' og 'pil ned'. Når det lysende felt er placeret over den ønskede funktion, tages (RETURN) for valg af funktion.

4.2. KOORDINATSYSTEM

Størstedelen af skærbilledet udgør et retvinklet X-Y-koordinatsystem med faste akseinddelinger.

Dette område fungerer som inddata- og uddataområde i forbindelse med den grafiske del af regressionsanalysen.

4.3. STEP, X & Y

I nederste venstre hjørne af skærbilledet findes en tabel, som fungerer i forbindelse med funktionerne:

- + PUNKTER
- PUNKTER
- FØLG KURVE

Under udførelse af disse funktioner bevæger brugeren en cursor rundt i koordinatsystemet ved hjælp af tastaturets fire pile:

- PIL OP
- PIL NED
- PIL MOD VENSTRE
- PIL MOD HØJRE

Cursorens aktuelle position i form af X- og Y-koordinater vises løbende i tabellen under hhv 'X' og 'Y'.

Ved en funktions start vil der under 'STEP' i tabellen stå '1', hvilket betyder at cursoren for hvert tryk på en af tastaturets pile bevæges et punkt i den anvendte pils retning. 'STEP-længden' er 1. Steplængden er variabel, og bestemmes ved at taste et af tallene 0 til 8. Tastes tallet N, hvor $0 \leq N \leq 8$, sættes steplængden til $20N$.

EKSEMPEL: tastes 5, vil der i tabellen under 'STEP' vises 32 (205), og et tryk på en pil bevirker, at cursoren bevæges 32 punkter i pilens retning.

Programmet tager højde for, at cursor ikke kan bevæges udenfor koordinatsystemets rammer.

4.4. FORMEL

Efter udførelse af en funktionerne:

1. GRAD
2. GRAD

hvorunder der beregnes et tilnærmende polynomium, udskrider i området under X-aksen formlen for det beregnede polynomium.

5. FUNKTIONER.

Som nævnt omfatter program REGRES i alt 8 funktioner:

+ PUNKTER	plot af punkter i koordinatsystem
- PUNKTER	fjernelse af punkter
1. GRAD	1. grads regressions
2. GRAD	2. grads regression
FØLG KURVE	aflæsning af kurveværdier
PLOTTE	udtegning på plotter
SLETTE	resette program
SLUT	afslutte program

Valg af funktion er beskrevet i afsnit 4.1.

5.1. '+ PUNKTER'

Valg af denne funktion bevirker, at brugeren kan bevæge en

cursor i form af en skråt opad mod venstre pegende pil. Pilen starter med at være placeret i (0, 0), og bevæges rundt i koordinatsystemet, som beskrevet i afsnit 4.3. I tabellen i nederste venstre hjørne vises løbende koordinaterne for pilespiden (øverste venstre punkt af cursor). Tastes (RETURN) vil det punkt, der p.t. udpeges af cursor blive markeret med et kryds (+) som et plottet punkt, der indgår i efterfølgende beregninger. Der kan plottes op til 25 punkter i koordinatsystemet. Flere punkter kan være sammenfaldende. Man returnerer fra funktionen '+ PUNKTER' til menuen for valg af ny funktion ved at taste 'Q' eller 'q'.

5.2. '- PUNKTER'

Ved hjælp af denne funktion kan man fjerne tidligere plottede punkter. Funktionen giver en cursor af form som er kvadrat, der bevæges i koordinatsystemet som beskrevet i afsnit 4.3. Når et punkt, der ønskes fjernet er helt omsluttet af cursoren, tages (RETURN). Punktet fjernes fra koordinatsystemet og udgår dermed af efterfølgende beregninger. Afslutning af funktionen sker ved at taste 'Q' eller 'q'.

5.3. '1. GRAD'

Ved valg af denne funktion slettes først eventuelt tidligere indtegnede kurver fra koordinatsystemet. Derefter beregnes et første grads polynomium, der tilnærmer de plottede punkter efter princippet for mindste kvadraters afvigelse og kurven (en ret linie) svarende til det beregnede polynomium indtegnes i koordinatsystemet. Derefter udskrives formelen for det beregnede polynomium i området under X-aksen, og der returneres til menu. Der skal være plottet mindst 2 punkter i koordinatsystemet forud for valg af denne funktion.

5.4. '2. GRAD'

Denne funktion er identisk med '1. GRAD' bort set fra, at det beregnede polynomium er af anden grad. Der skal være plottet mindst 3 punkter i koordinatsystemet forud for valg af denne funktion.

5.5. 'FØLG KURVE'

Denne funktion giver mulighed for at aflæse sammenhørende X-Y-værdier for den kurve, der p.t. er indtegnet i koordinatsystemet. Ved valg af 'FØLG KURVE' placeres en cursor af form som et kvadrat med et kryds igennem centrum på den aktuelle kurves venstre yderpunkt. Cursoren kan bevæges som beskrevet i afsnit 4.3., men kun i X-aksens positive eller negative retning. Cursoren vil for enhver indstillet X-værdi være placeret på den indtegnede kurve, og i tabellen i nederste venstre hjørne af skærbilledet kan man aflæse de sammenhørende værdier for X og Y, idet Y opgives med en decimals nøjagtighed. Der returneres til menu ved at taste 'Q' eller 'q'.

5.6. 'PLOTTE'

Valg af denne funktion kan kun ske, når der til COMET'en er sluttet en plotter af typen WATANABE DIGI-PLOT, og denne er tændt.

Valg af funktionen bevirker, at koordinatsystem samt de punkter og den tilhørende kurve, der p.t. er indtegnet i koordinatsystemet, tegnes ud på plotteren. Tegning af koordinatsystem sker dog kun første gang funktionen vælges. Endvidere vil formlen for det til kurven svarende polynomium blive udskrevet.

Vælges 'PLOTTE' funktionerne flere gange under en og samme afvikling af REGRES, vil der for hvert valg blive skiftet pen. Ligeledes vil afsatte punkter for hvert valg blive indtegnet med forskellige symboler.

Man bør således højst vælge 'PLOTTE' funktionen 6 gange under samme afvikling af REGRES.

5.7. 'SLETTE'

Denne funktion udtegner et skærbillede som ved start af programmet samt nulstiller samtlige variable i programmet. Funktionen virker med andre ord som en afslutning og genstart af programmet.

5.8. 'SLUT'

Medfører afslutning af programmet.