

2.

Maskinelle enheder

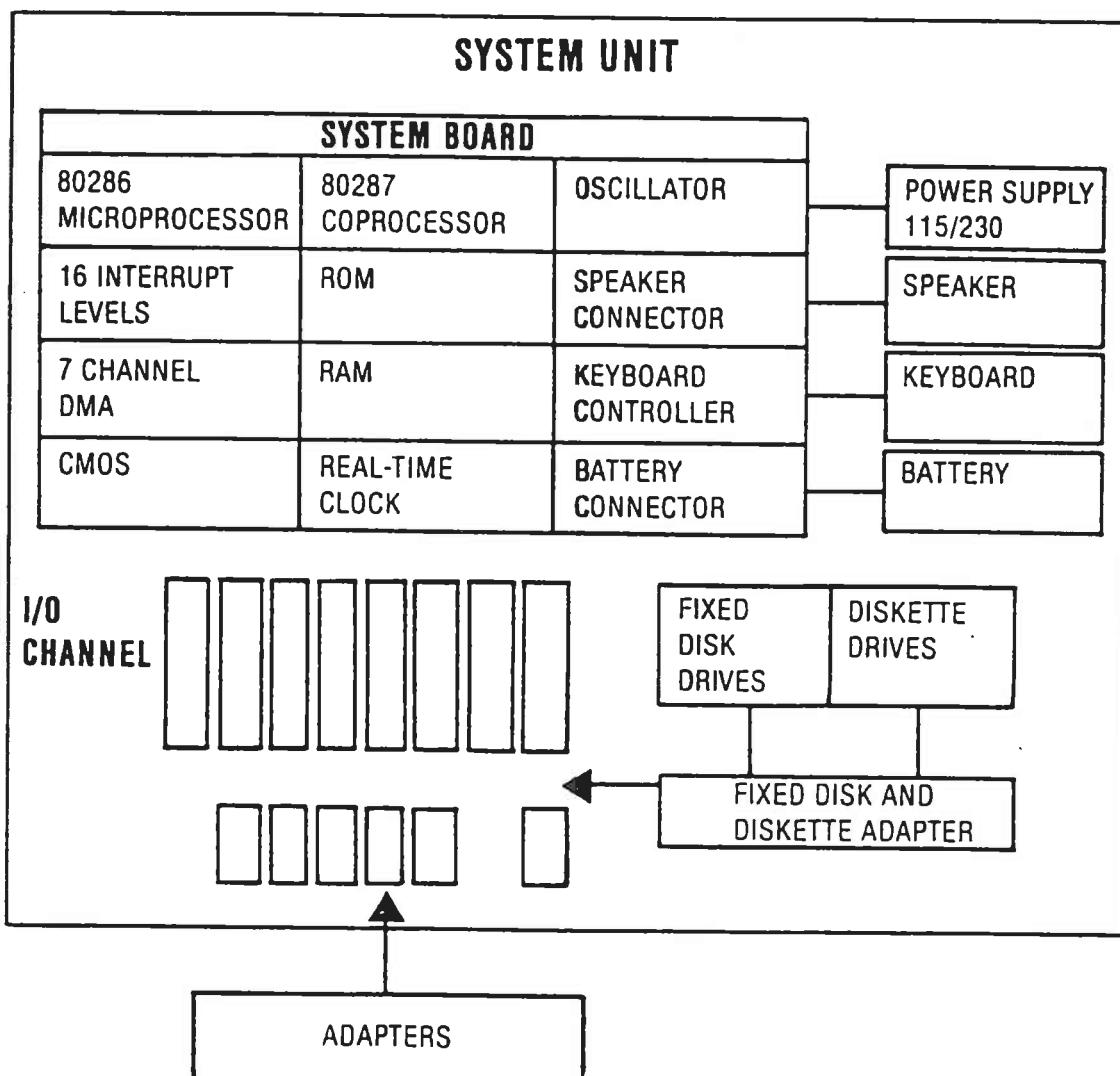
Indhold:

- 2.1 Systemenheden
 - a) Diagram: Systemenhedens komponenter.
 - b) Note: systemkomponenter (ikke med endnu)
- 2.2 Skærm'en
 - a) Artikel: Dataskærme
 - b) Annonce: LCD skærm
 - c) Tabel over skærmopløsninger
 - d) Skærmhukommelsen
- 2.3 Tastaturet
 - a) Note: Tastaturet
 - b) Vejledning: Tastatur-demo-program DEMOKBD.COM
 - c) Artikel: Sådan kommer trætte PC-fingere i form
- 2.4 Printere & plottere
 - a) Uddrag: Printere & plottere
 - c) Oversigt over printerstyretegn til canon BJ.300/330
- 2.5 Diskette & harddisk
 - a) Artikel: Trimming af harddisk
 - b) Diagrammer: Den fysiske diskette
 - c) Note: Harddisk og diskette under DOS (filsystem)
- 2.6 Mus
 - a) Artikel: Genvej til computerens hjerne

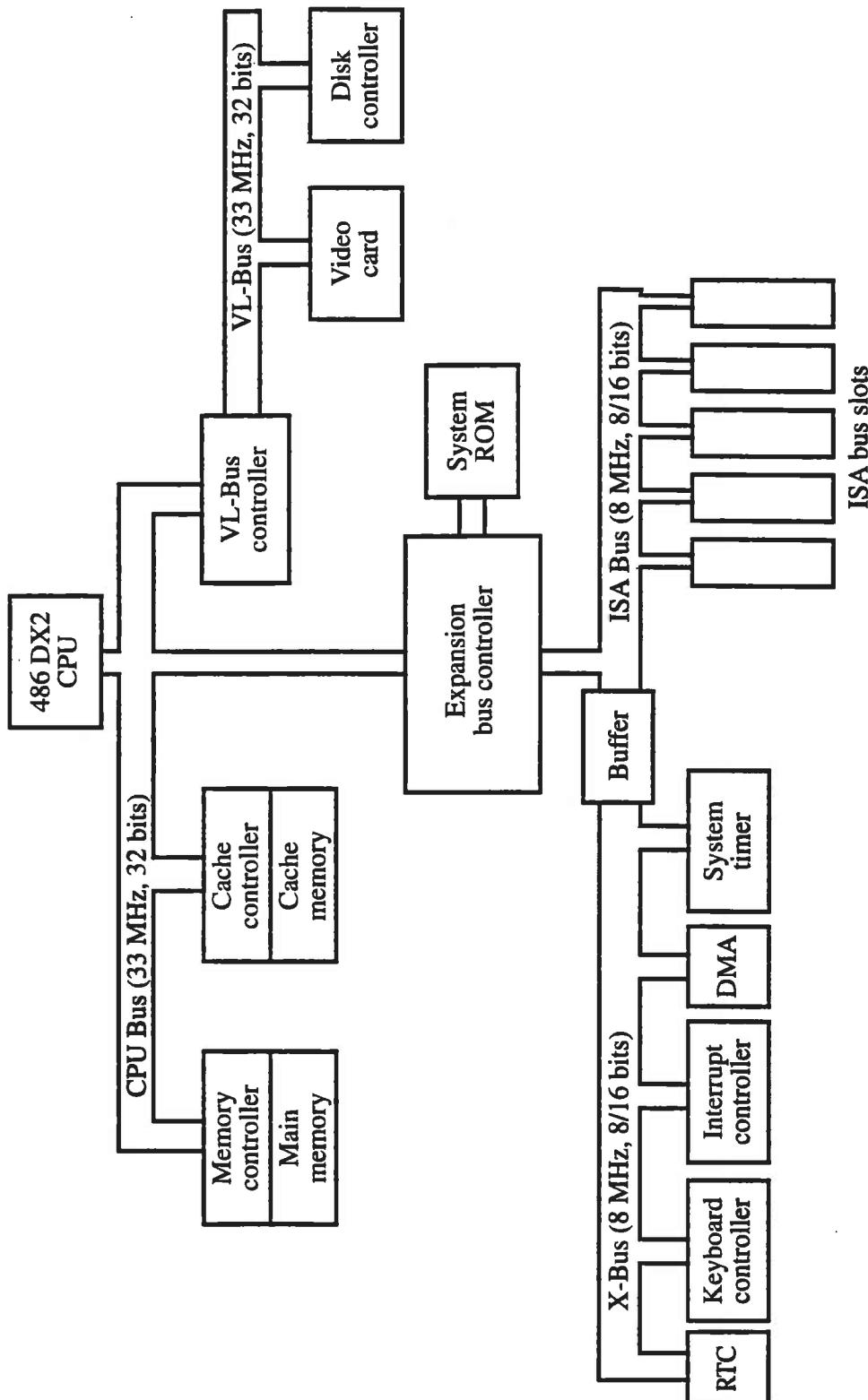
2.1.a

Systemenhedens komponenter

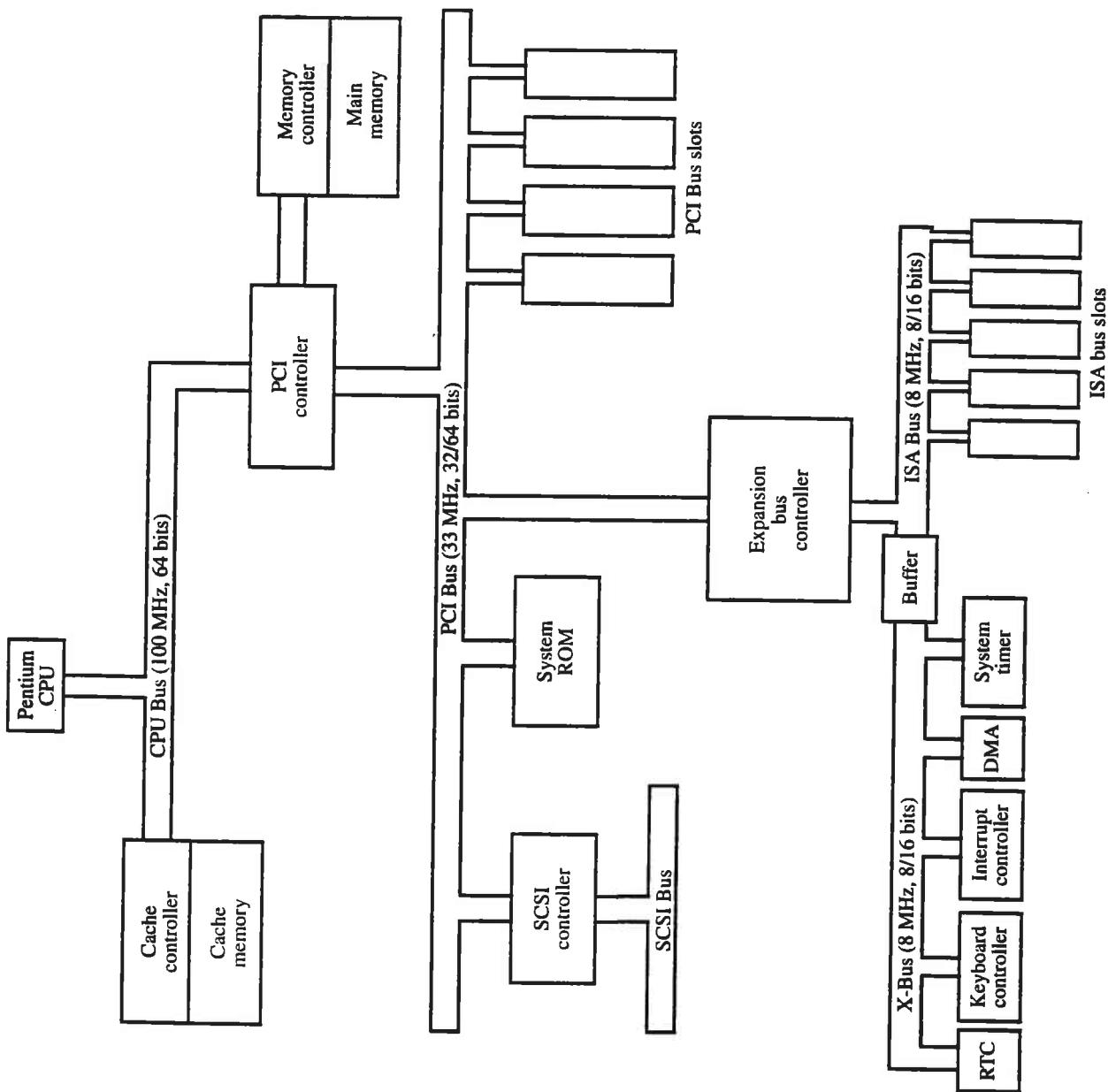
System Block Diagram



Block diagram showing the architecture of a basic ISA bus PC with a VL-Bus added to it



Block diagram showing the architecture of a PC that incorporates a PCI local bus.



2.2.a Dataskærme

Af Kaj Riis:

Den store vækst i salget af edb-løsninger til såvel privat som kommercial anvendelse, har naturligvis også ført et stigende salg af ydre enheder til installationerne med sig.

Da den egentlige kontakt med datamaten sker gennem blandt andet dataskærme, er det ikke underligt at der lægges større og større vægt på design og kvalitet af disse.

På samme måde stilles stadig større krav til de andre ydre enheder, der knytter sig til enhver form for databehandling

Fra de første dataskærmes fremkomst i starten af 60'erne og indtil i dag, er der sket en kolossal udvikling. En udvikling, der primært har givet sig udslag i et bedre billede, både hvad angår farver, oplosning og stabilitet.

Prisudviklingen er gået som inden for al anden elektronik, nemlig nedad. Dog ikke helt i samme tempo – et forhold, der også ses på fjernsynspriserne, der i forhold til f.eks. lønudviklingen ikke er faldet ret meget de seneste 10 år.

Som man ofte sammenligner kr. pr. Byte for maskiner og disketterdrev, kr. pr. tegn/sek. ved printere, sammenligner man nu kr. pr. pixel ved farveskærme.

I denne konstruerede måleenhed ses udviklingen i pris og ydeevne allerbedst.

Og hvad er så en monitor?

Hovedparten af mikroer til privat benytelse er tilsluttet husets fjernsyn. Mange mikroer er lavet specielt til dette, idet de højest arbejder med 40 tegn på hver linie. Og ca. 40 tegn er netop hvad et almindeligt fjernsyn kan gengive uden større problemer med at læse teksten. Samtidig er fjernsynets oplosning tilstrækkelig til den grafik, der er indbygget i de fleste hjemmecomputerne.

Problemerne dukker op, når der skal tilsluttes en mikro, der giver 80 tegn pr. linie, hvilket er det normale i al databehandling. Her er der simpelt hen ikke punkter nok på fjernsynsskærmen til, at de enkelte bogstaver kan dannes tydeligt nok. Punkterne opgives også som pixel eller linier.

Da UHF-signalet til et fjernsyn går igennem både modulator (i mikroen) og demodulator (i fjernsynet), før det når video-forstærkeren, sker der en svækkelse af signalets kvalitet. Samtidig er standards for et fjernsynsbillede lagt for mange år siden, således at op-

lösningen (antal punkter) ikke kan ændres.

Derfor benyttes specielle datamonitorer til f.eks. administrative opgaver og andre opgaver, hvor der stilles store krav til billedkvaliteten.

Monochrome monitorer

Monochrome monitorer har en oplosning på 18-25 MHz og er normalt baseret på videosignal som input. Den mest anvendte fosforbelægning er stadig væk P4 (sort/hvid) og P31 (sort/grøn). Den røværede skærm er i stadig fremgang, idet mange eksperter mener, det er bedre for øjnene ved længere tids skærmarbejde.

Hvor grafisk anvendelse kommer ind i billedet, benyttes ofte en fosfor med længere efterglød, kaldet P39 eller P42.

Farvemonitorer

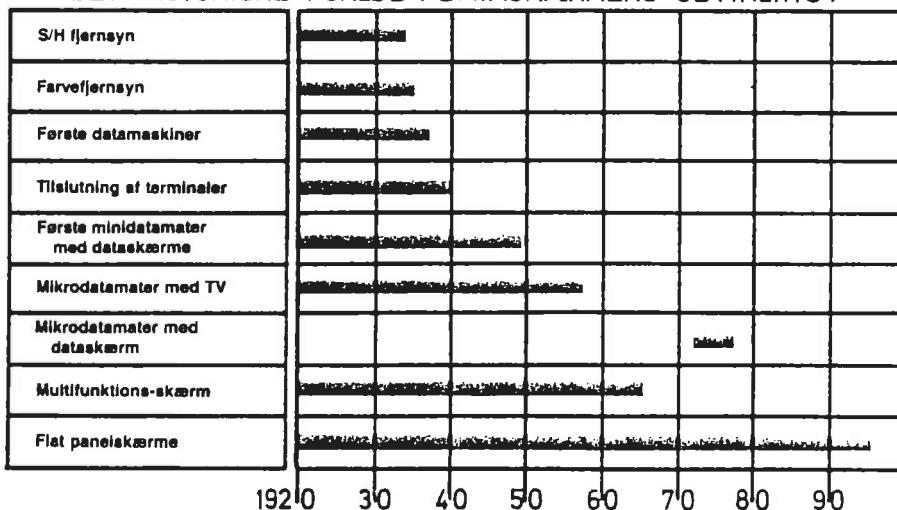
Farvemonitorer findes i et noget større spektrum end de monochrome. En oplosning svarende til lidt over farve-TV er på 380×280 punkter. Næste spring er normalt en fordobling af den horizontale oplosning. På skærme til grafisk databehandling er oplosningen ofte 1280×1024 . Samtidig benyttes 60 Hz scanningfrekvens og non-interlaced billedoverførelse for at sikre et helt stabilt billede.

Normalt stiger antallet af punkter naturligvis med skærmsørrelsen. For farvemonitorer på 9" til 20" ligger båndbredden på 15-40 MHz, oftest dog på 18-20 MHz, hvilket giver et godt billede med op til 2000 tegn. Skal man højere op, kræves 40 MHz til 4000 tegn eller f.eks. 960×720 punkter.

Et tredje væsentligt element er størrelsen af de enkelte punkter samt afstanden mellem disse. Dette måles i mm og kaldes pitch. Oftest mindskes pitch-sørrelsen samtidig med forøgelsen af antal punkter. Mest anvendte størrelse er 0.31 mm.

Når der tale om datamonitorer, kommer der uvægerligt en masse udtryk og begreber ind i billedet. Begreb, de fleste ikke er bekendte med, eller ikke anvender. I det følgende gives derfor en gennemgang af grundprincipperne i opbygning af en monitor.

DET HISTORISKE FORLØB I DATASKÆRMENS UDVIKLING :



Dataskærme...

Lidt monitorteknik

En monochrom monitor består af et billedrør med tre hovedelementer: 1) En elektronkanon, der afsender en hurtig stråle af elektroner mod billedrørets inderseite, 2) en *afbøjningsspole*, der afbøjer elektronstrålen mod et givet punkt på et givet tidspunkt og 3) en skærm, der er belagt med én af de nævnte fosfortyper, som gløder i en brøkdel af et sekund, på det sted elektronstrålen rammer.

Det sted, hvor belysning finder sted, kaldes et *punkt* eller en *pixel* (picture element). Det er antallet af disse, der giver oplosningen. Styrken for belysning afhænger af den spænding, der afgives til katodestrålerøret. Da strålen gennem afbøjningen kommer over hele skærmen, skal dens intensitet ved hver pixel med kolossal præcision sættes ON og OFF, afhængig af det billede, der skal dannes på skærmen.

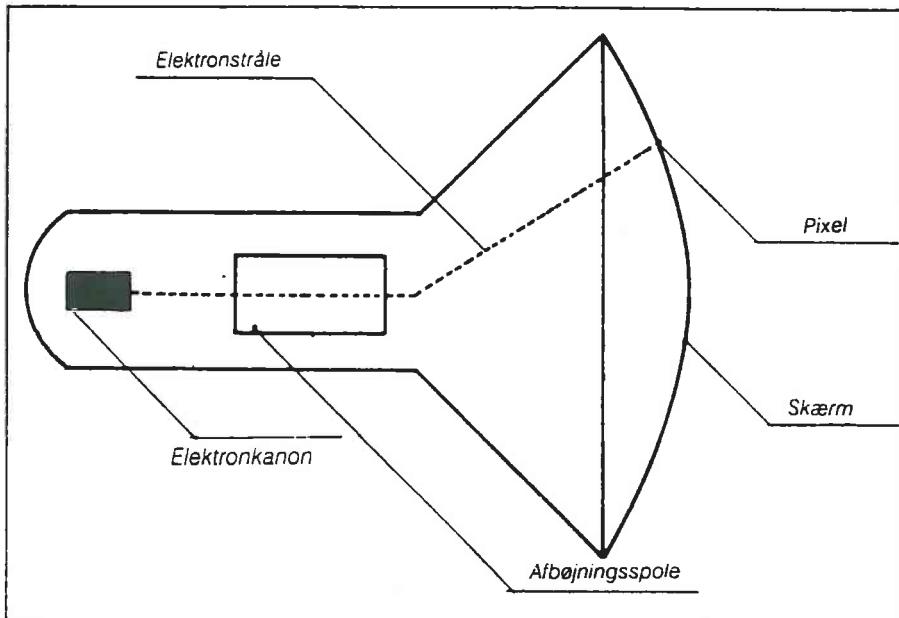
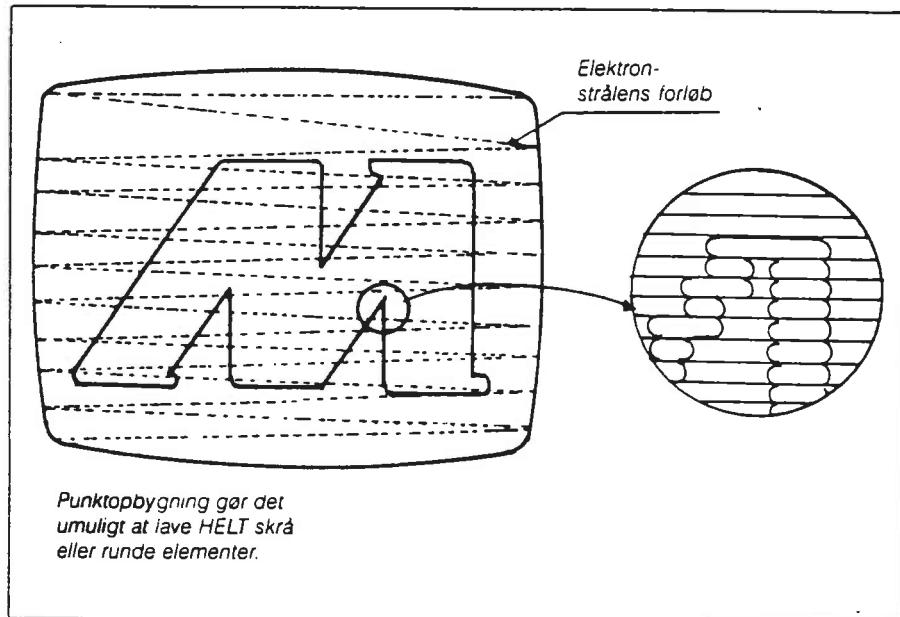
Da opglødningen af fosforbelægningen hurtigt forsvinder, skal elektronstrålen løbe gennem hver pixel på hele skærmen konstant. Dette sker 30-60 gange pr. sek.

Positioneringen af strålen styres af afbøjningsspolen, der består af to magnetiske spoler, der »kontrollerer« henholdsvis horizontal og vertikal retning. I et RASTER-SCAN system, som næsten alle monitorer er baseret på, starter strålen i øverste venstre hjørne og bevæger sig derefter mod højre og lidt nedad, indtil højre kant rammes. Samme bevægelse følges nu den anden vej og der fortsættes indtil nederste højre hjørne er nået, hvorefter strålen igen starter forfra. Dette gennemløb sker som nævnt mindst 30 gange pr. sek. (= 30 Hz).

Hvis denne »refresh.rate« er for lille, hvad 30 gange/sek. er, ses en tydelig blinken på skærmen. Til data- og tekstbehandling benyttes derfor oftest en hastighed på 50-60 Hz eller mere.

Og så til farvemonitorer

I farvemonitorer indgår væsentlig flere elementer og mere styring end i de monochrome. En betydelig forskel er, at der indgår *tre elektronkanoner* i stedet for én. En for hver af farverne rød, grøn og blå. Heraf betegnelsen RGB-



monitor. En meget vanskelig proces danner den vigtige fosforbelægning, hvor hver pixel – punkt – dannes og består af tre »dots«, én for hver basisfarve. Afhængig af hvilken farve, der skal komme til udtryk i det enkelte punkt, styres intensiteten på hver af de tre elektronstråler, der rettes mod det aktuelle punkt. For at alle tre stråler skal ramme præcis det samme sted, ligger der bag skærmen en såkaldt »shadow-mask«, som er en ganske tynd

metalfilm med gennemperforerede huller.

Denne lidt mere komplicerede opbygning medfører desværre, at farvemonitorer er dybere end tilsvarende størrelse s/h.

Afhængig af katodestrålerørets placering i billedrøret, skelnes mellem delta-gun og in-line gun, hvor sidstnævnte type er langt den mest udbredte.

Dataskærme . . .

Som det ses af fig. 4 er det primært pitch-størrelsen i shadow-masken, der afgør monitorenens oplosning. Men som tidligere nævnt betyder også båndbredden (i MHz) en masse. Båndbredden er den hastighed, hvormed de enkelte punkt-oplysninger modtages af monitoren til styring af de enkelte farver og til fastsættelse af hastigheden for strålernes gennemløb af alle punkter på skærmen.

Hvis en monitor med oplosning på 1024×1024 punkter skal gennemløbes 30 gange pr. sek., kræves en båndbredde på mere end 30 MHz. D.v.s. mere end 30 millioner »bevægelser« pr. sek.

Mere end én mill. farver

Hvor mange farver, der kan vises på en skærm, afhænger af hvorledes de tre elektronstrålers intensitet kan styres. I de enkleste tilfælde opnås otte farver, styret af tre informationsbits pr. punkt: Ingen farve, rød, grøn, blå, rød/grøn, rød/blå, blå/grøn og rød/blå/grøn (= hvid).

Kræves 16 farver, hvilket betragtes som minimum til mange opgaver, skal man bruge fire bits til styringen.

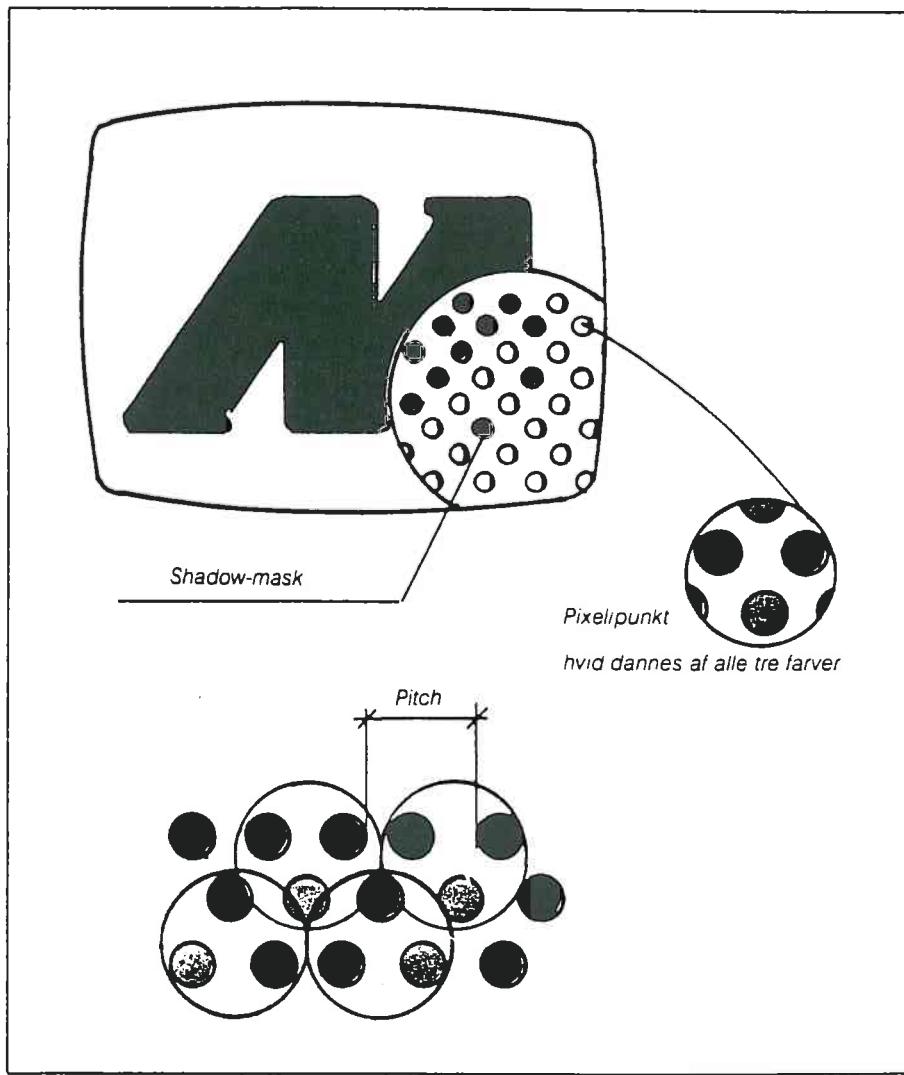
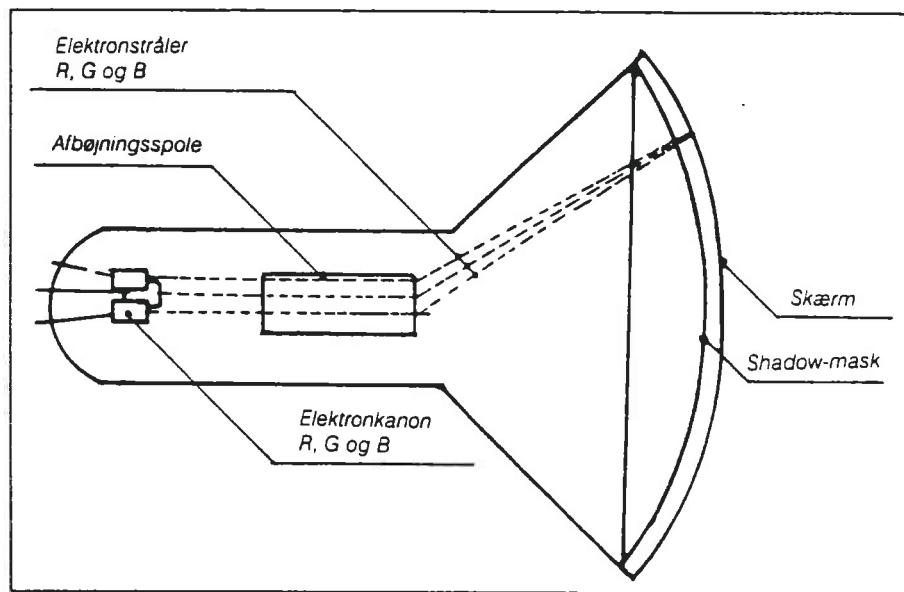
Til monitorer til grafisk databehandling anvendes ofte 8-bits koder til styring af HVER af de tre farvekanoner. Dette giver $256 \times 256 \times 256$ gradueringer (lysstyrke) på farveskalaen (mere end 16 mill.).

RGB og Composite Video

Hovedforskellen mellem de to typer er om de tre elektronstråler kontrolleres af ét sammensat (composite) videosignal eller om de kontrolleres af tre signaler (RGB). Composite video svarer meget til den metode, der anvendes i et almindeligt farve-TV. Det siger næsten sig selv, at en større præcision kan opnås ved RGB-udgaven, som da også benyttes ved stort set alle professionelle systemer.

Priser

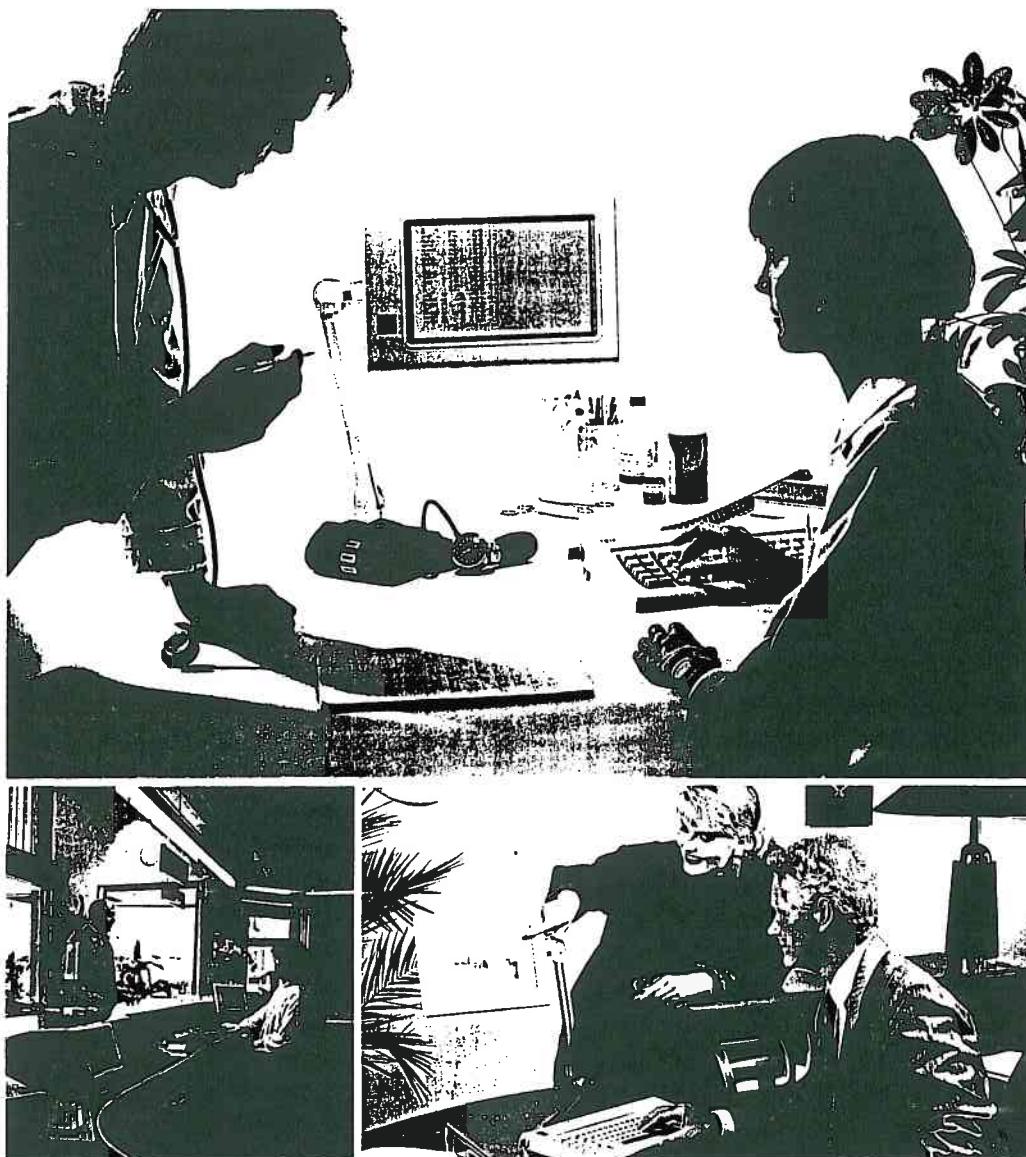
Farvemonitorer fås fra ca. kr. 3000 og opefter. Jo flere farver og jo større oplosning, desto højere pris. En ultra-high resolution monitor kan koste op til knap 100.000 kr. □



2.2.b

LCD Flat-screen

CTC
COMPUTER



- Et alternativ til CRT-skærme
- Pladsbesparende
- Ergonomisk
- Skånsom mod øjene

Spar plads

LCD-skærmen fylder kun 5% af en standard CRT monitor og vil gøre ethvert kontormiljø mere raffineret. Den enkle elegante arm, der holder LCD-skærmen, gør det muligt at placere skærmen netop der hvor det er mest behageligt. LCD-skærmen kan også ideelt monteres uden arm direkte på et kontormøbel.

Skånsom mod dine øjne

Når du begynder at arbejde med LCD-skærmen, vil du blive overrasket over hvor hurtigt du vender dig til den. En konstant klage fra CRT-brugerne er anstrengelsen af øjnene på grund af flimmer fra displayet og det store kontrastniveau mellem tekst og baggrund. CTC's LCD-skærm er fri for flimmer og har en kontrast som er nær den man er vant til når man læser en bog.

Ingen stråling

Undersøgelser af sundhedsrisikoen ved stråling fra traditionelle CRT-skærme er endnu uklaret. Sundhedsspørgsmålet er af stigende interesse og et diskussionsobjekt for både brugere, producent, medier og relevante autoriteter.

LCD-skærme giver ingen stråling.

2.2.c

Skærmopløsninger

en oversigt over nogle af de almindeligste korttyper og de tilsvarende monitorer. Oversigten er langtfra fuldstændig, da der er adskillige computerfabrikanter, som har deres egne videokort og skærme som passer til, men hvor det er ganske besværligt at finde ud af, hvilke data grejet overholder.

Korttype	Mode	Opløsning	Antal farver	Linie-/billedfrekvens
MA	tekst	720 × 350	2	18,4 kHz/50 Hz
HM	tekst	720 × 350	2	18,8 kHz/50 Hz
HM	grafik	720 × 348	2	18,8 kHz/50 Hz
CGA	tekst	320 × 200	16	15,7 kHz/60 Hz
CGA	grafik	320 × 200	4	15,7 kHz/60 Hz
EGA	tekst	720 × 350	2	21,8 kHz/60 Hz
EGA	grafik	640 × 350	16	21,8 kHz/60 Hz
VGA	tekst	720 × 400	16	31,5 kHz/70 Hz
VGA	grafik	640 × 480	16	31,5 kHz/60 Hz
8514/A	grafik	1024 × 780	256	35,5 kHz/43 Hz
VGA og 8514/A hører til IBM's PS/2-serie				

Oversigt over forskellige grafik-modes

2.2.d

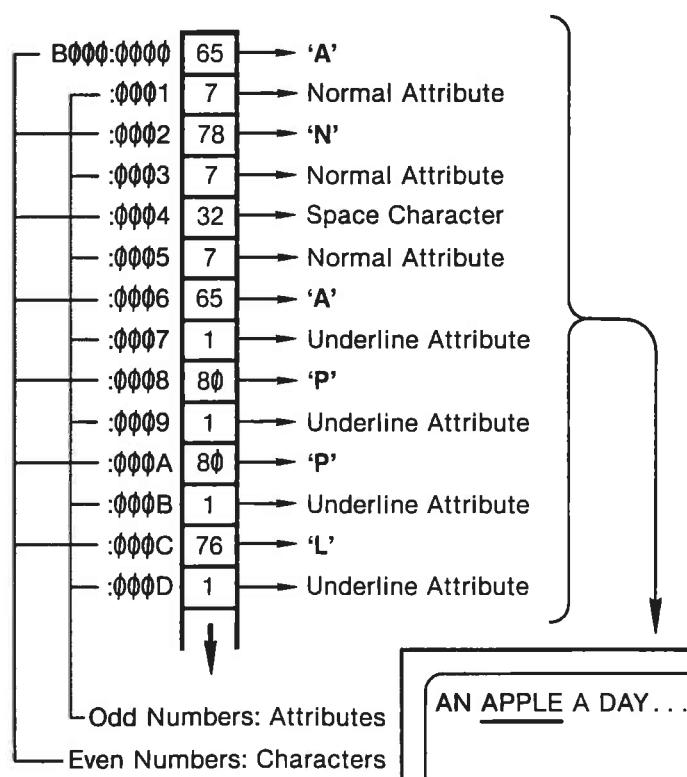
Skærmhukommelsen

Fra programkode til skærmbillede

Der er to principielt forskellige metoder til at overføre information til en skærm. Den ene metode er at udstede kommandoer (typisk som escape sekvenser) i stil med: Vis følgende tekst: "Velkommen til ..." på linie 3, søjle 5 på skærmen i omvendt video. Den anden er, at lade adapteren skandere et lagerområde, som et program kan skrive i, og vise det fundne på monitoren ("Bit mapping"). Det sidste er det almindeligste til PC'er.

Ehvert program har normalt fuld adgang til det lager (skærmbufferen), der skanderes, da det er en del af 1 MB adresserummet. Der benyttes op til 128 KB i adresserummet i området A000:0000 – C000:0000. Den monokrome buffer begynder således på adresse B000:0000 og CGA bufferen på adresse B800:0000 (se fig. 2.3.1). EGA og senere standarder benytter også B800:0000 til tekst, men til grafik er det mere indviklet, da der måske skal bruges mere end 128 KB.

Arbejdes med tekst, fylder hvert tegn 2 byte i bufferen, nemlig en byte til tegnets ASCII kode på lige adresse og en byte til tegnets attributter på efterfølgende ulige



Memory mapping on the monochrome adaptor.

	0	0	0	0	Sort Understregning			
	0	0	0	1	Grøn			
	0	0	1	0				
	0	0	1	1				
	0	1	0	0				
	0	1	0	1				
	0	1	1	0				
Forgrunds- farver	0	1	1	1				
	1	0	0	0	Sort Understreget, høj intensitet			
	1	0	0	1				
	1	0	1	0				
	1	0	1	1				
	1	1	0	0	Grøn, høj intensitet			
	1	1	0	1				
Blink-----1	1	1	1	0				
Ikke blink----0	1	1	1	1				
		R	G	B	I	R	G	B
Sort		0	0	0				
		0	0	1				
		0	1	0				
Grøn, omvendt video (dog sort ved understreg- ning)		0	1	1	Baggrunds- farver			
		1	0	0				
		1	0	1				
		1	1	0				
		1	1	1				
		1	1	1				

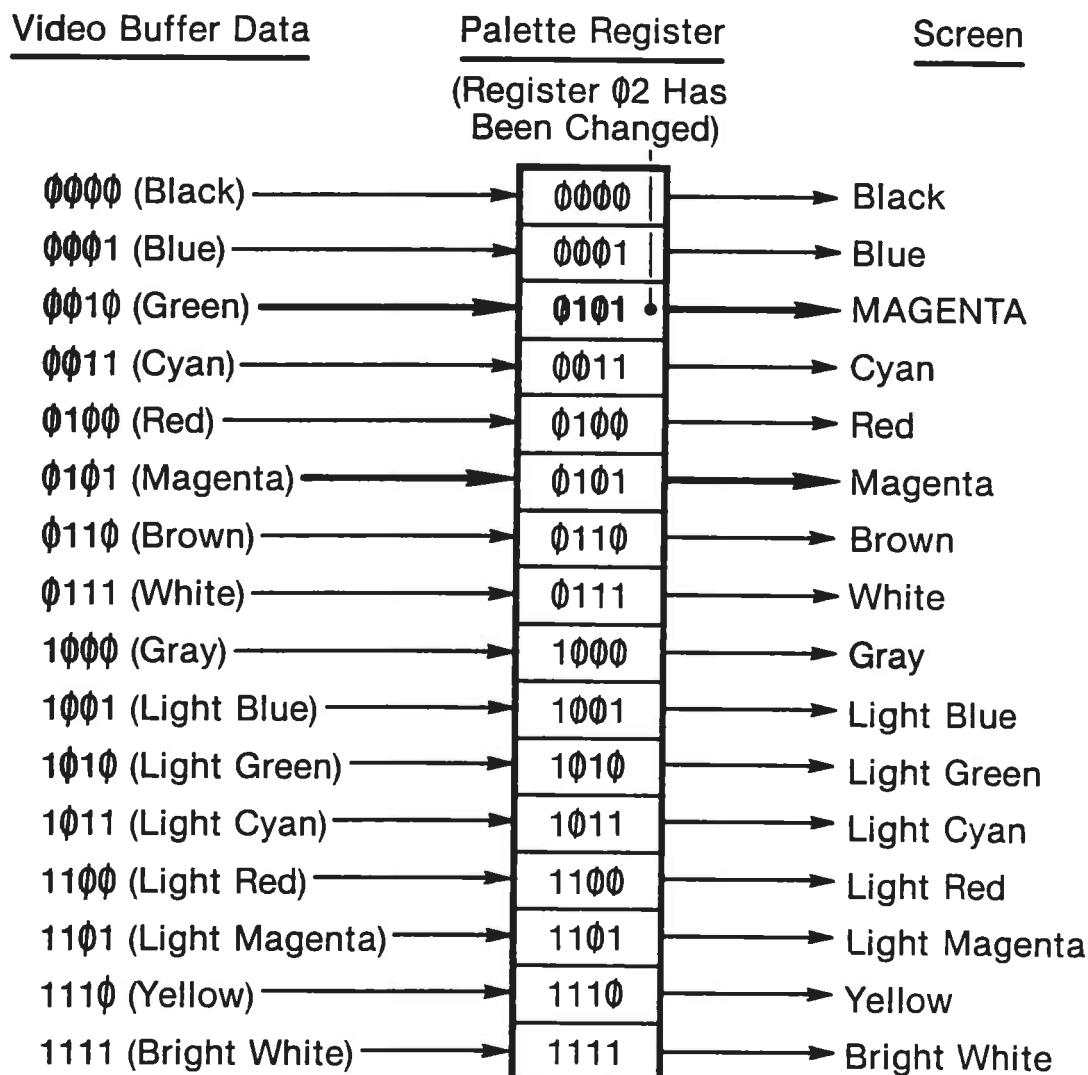
Figur 3.3.2. Attribut-bittenes betydning for IBMs monokrome skærm. Bemærk, at kombinationen: understreget forgrund + grøn baggrund, ikke giver omvendt video, men blot understregning.

	0	0	0	0	Sort			
	0	0	0	1	Blå			
	0	0	1	0	Grøn			
	0	0	1	1	Cyan			
	0	1	0	0	Rød			
	0	1	0	1	Magenta			
	0	1	1	0	Brun			
Forgrunds- farver	0	1	1	1	Hvid			
	1	0	0	0	Grå			
	1	0	0	1	Lyseblå			
	1	0	1	0	Lysegrøn			
	1	0	1	1	Lys cyan			
	1	1	0	0	Lyserød			
	1	1	0	1	Lys magenta			
Blink-----1	1	1	1	0	Gul			
Ikke blink----0	1	1	1	1	Kraftigt hvid			
		R	G	B	I	R	G	B
Sort		0	0	0				
Blå		0	0	1				
Grøn		0	1	0				
Cyan		0	1	1	Baggrunds- farver			
Rød		1	0	0				
Magenta		1	0	1				
Brun		1	1	0				
Hvid		1	1	1				

Figur 3.3.1. Attribut-bittenes betydning ved tekst på en RGBI monitor. Under VGA standarden kan I-bitten benyttes til noget helt andet end intensitet, nemlig til at vælge mellem to tegnsæt indlæst i RAM; hvert tegn på skærmen kan altså hentes fra en af to tegntabeller, svarende til 512 forskellige tegn. Det samlede antal tegntabeller, der kan vælges fra er 8.

Farvespalette

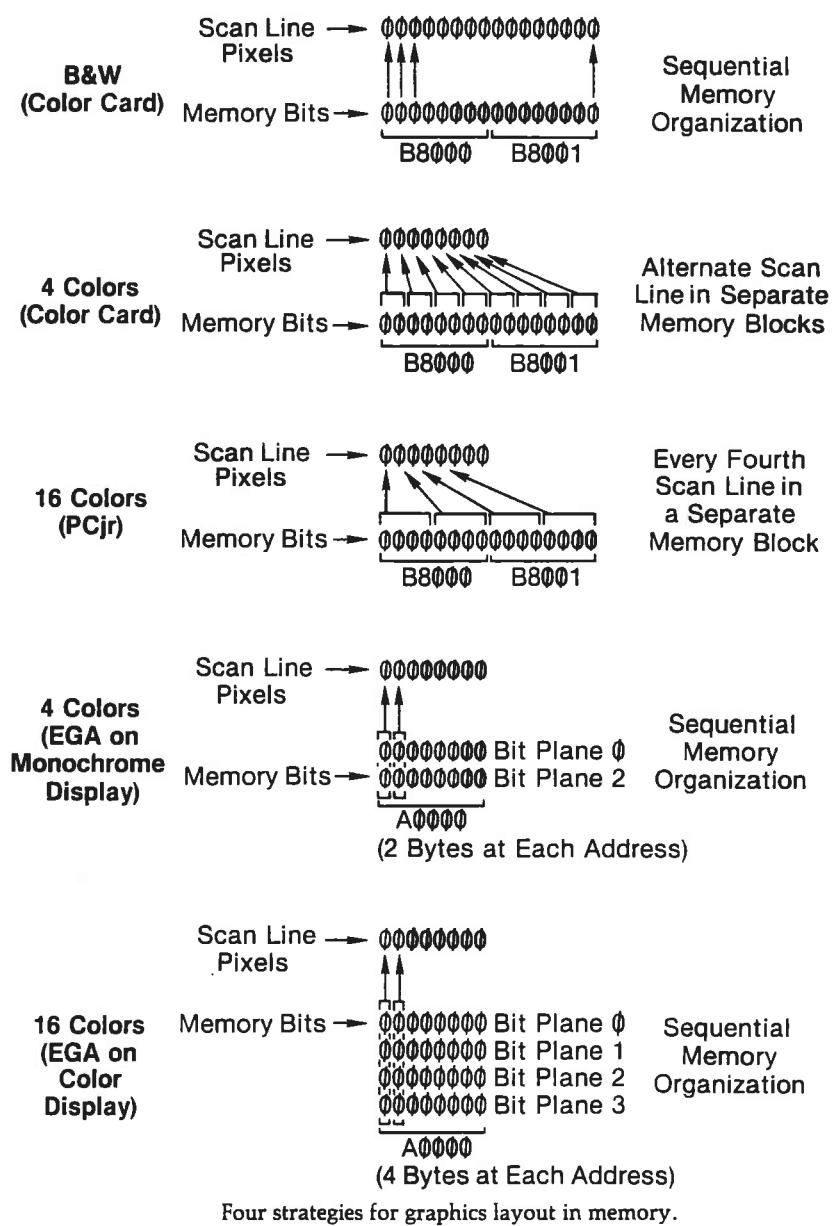
På EGA og VGA skærmkort anvedes skærmattibutter ikke direkte, men der sker et farvevalg ud fra en palette. Der kan ved ændring af paletten vælges andre farver f.eks 7 røde nuancer istedet for rød, grøn, blå o.s.v.



Displaying "green" as magenta.

Grafikskærmen

Der findes flere måder, at opbygge skærmhukommelsen på, når skærmen skal anvendes til grafik.



2.3.a

Tastaturet

Bjørk Busch

Keyboard'et har sin egen indbyggede microprocessor, på IBM-PC'er og mange compatible maskiner en INTEL 8048 chip.

Denne processor har til opgave, kontinuerligt at "scanne" tastaturet for at opdage, om et tastetryk har fundet sted (eller om en taste er blevet "sluppet" igen).

Når et tastetryk (eller -slip) er konstateret, genererer 8048 en SCANKODE for den pågældende taste og anbringer denne kode i en buffer i selve tastaturet. Denne buffer har plads til max. 19 scan-koder plus en "stopklods-kode", som er FFH. Når alle pladser er fyldt, og der finder yderligere indtastninger sted, så ignoreres disse tastninger.

En SCAN-KODE er et simpelt løbenummer for tastaturets taster. Som regel starter nummereringen med tasten "Esc" som scan-kode nr 1, tallet 1 og tegnet ! (udråbstegn) er fælles om scan-kode nr 2 og tallet 2, tegnet " og tegnet @ deler alle 3 samme taste og er fælles om scan-kode nr. 3. Kun undtagelsesvis genereres der forskellige scan-koder for samme taste.

Taster, som har samme tegn eller funktion, genererer hver sin scan-kode.

Tallet 1 fra hoved-tastaturet giver en anden kode end tallet 1 fra nummer-tastaturet, og tegnet '*', som ofte findes i flere taster, genererer også hver sin scan-kode.

O.s.v. Der findes naturligvis optegnelser om disse scan-koder i computerens tekniske manualer.

Nedenfor er vist en oversigt over en tastatur med tilhørende scankoder.

Der gøres op mærksom på, at tildelingen af scan-koder til de enkelte taster i høj grad er fabrikat-afhængig. Scan-koden for et taste-"slip" sættes lig med tastens scan-kode + 128.

Hexadecimal scan codes for an Enhanced Keyboard.

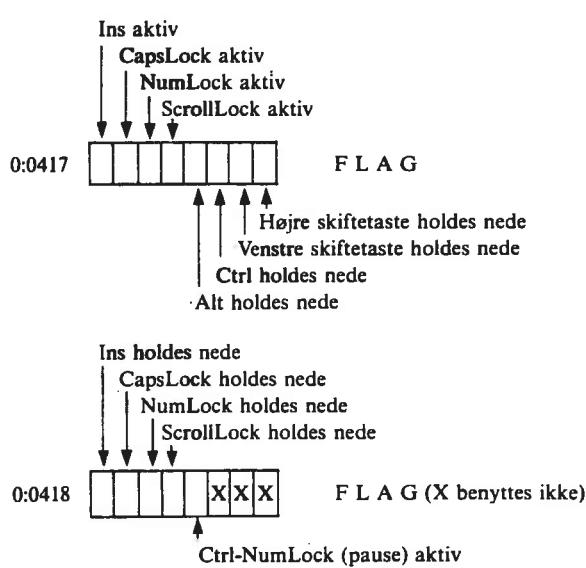
01 Esc	3B F1	3C F2	3D F3	3E F4	3F F5	40 F6	41 F7	42 F8	43 F9	44 F10	57 F11	58 F12	*12 P.S.	46 S.L.	E1 Pow					
29 ~	02 !	03 @	04 #	05 \$	06 %	07 &	08 ^	09 >	0A (0B)	0C -	0D =	0E Backspace	* 52 Ins	* 47 Hom	* 48 Pup				
0F Tab	10 Q	11 W	12 E	13 R	14 T	15 Y	16 U	17 I	18 O	19 P	1A [1B]	2B \	45 N.L.	* 35 /	37 *	4A -			
3A Caps Lk	1E A	1F S	20 D	21 F	22 G	23 H	24 J	25 K	26 L	27 :	28 ;	1C Enter	* 53 Del	* 4F End	* 49 Pdn	47 7	48 8	49 9	4E +	
2A Shift	2C Z	2D X	2E C	2F V	30 B	31 N	32 M	33 <	34 >	35 ?	36 Shift		* 48 :	4B 6	4C 5	4D 6				
1D Ctrl	3B AH	39	Space Bar				* 38 Alt	* 1D Ctrl	* 48 -	* 50 !	* 4D -		4F 1	50 2	51 3	* 1C 3End	52 0	53 .		

* = Scan Code is preceded by EO

Samtidig med, at 8048 anbringer scan-koden på plads i bufferen, så udsteder den et hardware-interrupt (nr. 9H) til hoved-processoren 8088.

Den bios-rutine, som dette interrupt initierer, udfører nu flere på hinanden følgende opgaver:

- Der sættes et "READY" -signal op på en af I/O portene (portene 60H til 63H er på IBM-pc reserveret til keyboardet) hvilket betyder "Kom!, hvad har du at sende".
- 8048 svarer med at sætte den scan-kode op, hvis tur det er til at blive behandlet, op i en af de andre i/o-porte.
- Bios-rutinen tager nu scankoden ind via systembussen og analyserer den. Har den værdien FFH, så er det tegn på, at tastaturets buffer er løbet fuld, og bipperen aktiveres som en advarsel til brugeren.
- Hvis scan-koden betyder, at en af følgende taster har været aktiveret:
 <venstre SHIFT>, <højre SHIFT>, <CAPS LOCK>, <CTRL>, <ALT>, <NUM LOCK>, <SCROLL LOCK>, <PAUSE> el. evt. tilsvarende taster (fabrikat-afhængigt) - så opdateres en særlig bitmaske på adresse 0:417H-418H (IBM), som indeholder keyboardets STATUS-FLAG.
 Status-flagenes har følgende betydning:



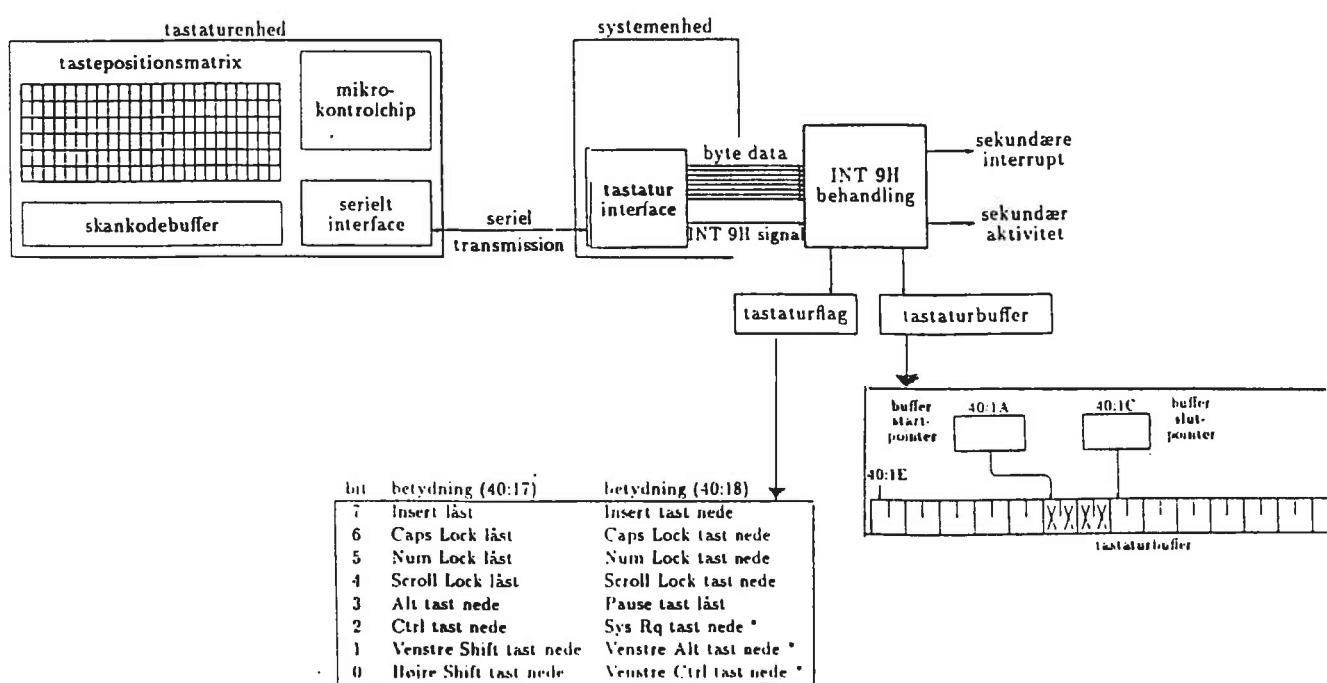
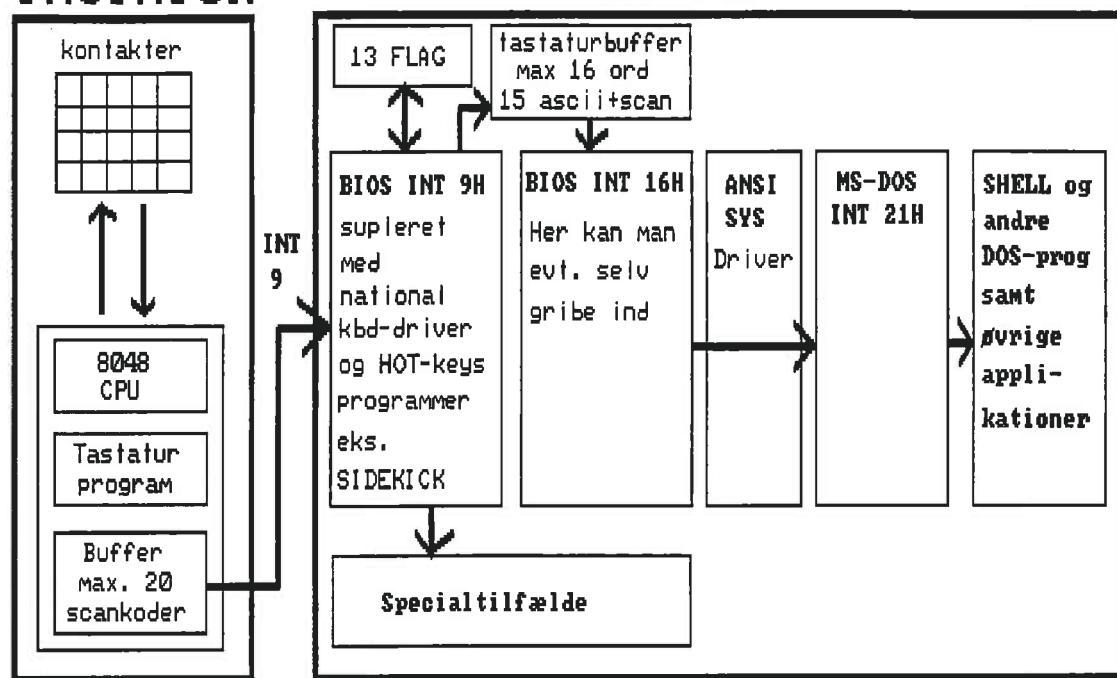
- Bemærk INSERT tastet, som er lidt speciel, idet den **BÅDE** sætter et flag og giver en SCAN-kode i tastaturbufferen. Ins-flaget bruges sjældent, idet programmerne selv tager hånd om denne funktion, når de indlæser scan-koden fra tastaturbufferen.
 For nogle programmer er det antallet af tryk på INSERT, der afgør hvor mange pladser, der skal rykkes. Denne funktion kan ikke opnås ved, at teste den aktuelle status.

- Udfra alle andre scan-koder genererer bios-rutinen nu en 2-bytes kode, hvor scan-koden anbringes i HI-order-byten og det tilsvarende ascii-tegns værdi anbringes i LOW-order byten. Ascii-tegnets værdien findes ved opslag i systemenhedens skrift-tegn tabel med et index, der først er blevet beregnet udfra scan-koden og udfra det bitmønster, som fremgår af status-flagene. Hvis der er tale om tastetryk, der stammer fra tastaturets special-taster (F-tasterne, pil-tasterne mv.), så har disse ingen ascii-repræsentation, og 2-bytes-kodens Low-order-byte får derfor værdien 00H.
- Bios-rutinen noterer det nøjagtige tidspunkt for hvert eneste tastetryk. Hvis et tastetryk ikke er blevet ophævet (ved modtagelsen af den tilsvarende scan-kode for "slip" af tasten) inden et halvt sekund, så vil bios-rutinen af sig selv automatisk generere en scan-kode for taste-tryk for hver tiendedel sekund, der går, inden taste-slippet konstateres. Dette gælder for langt de fleste af tastaturets taster, men ikke for dem alle. Prøv dig frem og lav en liste over de taster, der ikke kan "repetere".
- De genererede 2-bytes koder anbringes i en keyboard-buffer, der er 16 ord lang og organiseret som en cirkulær kø (ring-kø). Ringen styres af to pointere på hver et ord et HOVED og en HALE. HOVEDET peger på det næste tegn, som ligger klar til modtagelse af programmet. HALEN peger på det sted hvor et nyt indtastet tegn skal placeres. Hvis HOVED og HALE peger på samme tegn er bufferen tom, hvilket betyder, at der KUN kan placeres 15 tegn i bufferen. Hvis der blev placeret 16 tegn ville HOVED og HALE nemlig igen pege på samme tegn, og der kunne ikke skelnes mellem en fuld og en tom buffer. Der kan altså "huskes" i alt 15 tastetryk, som er til rådighed for DOS eller for et brugerprogram i "FIFO"-rækkefølge. Princip og operationer for en cirkulær kø er illustreret senere.
- Hermed er bios-rutinen, der kaldtes fra interrupt nr. 9H, færdig med sit arbejde og kommandoerne returneres til programmet.

Når brugerprogrammet (eller DOS) ønsker at "se" og anvende de tegn koder, der er placeret i keyboard-bufferen, anvender det (via dets læse-instruktion) en anden bios-rutine, som kaldes fra software-interruptet 16H.

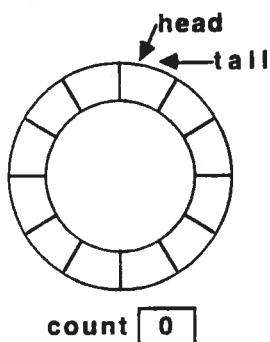
Tastaturet og behandlingen frem til modtager

TASTATUR PC - SYSTEMENHED

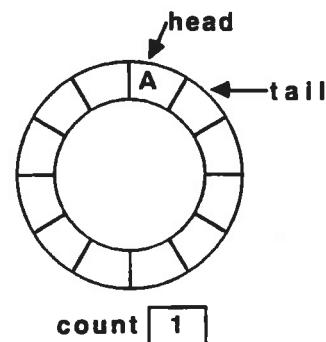


Tastaturlufferen fungerer som en cirkulær kø:

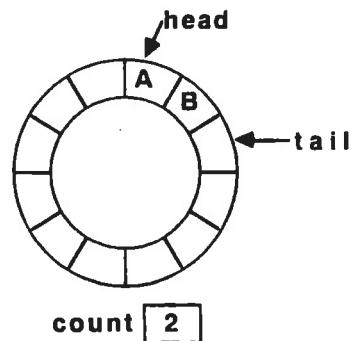
1/ Initial situation (empty queue)



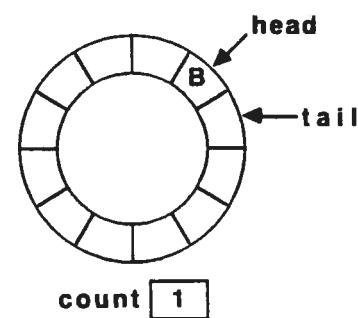
2/ After enqueue (value = A)



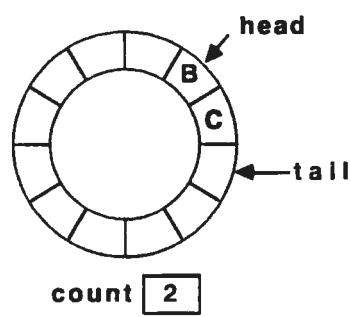
3/ After enqueue (value = B)



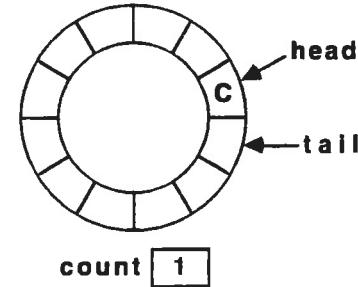
4/ After dequeue (A dequeued)



5/ After enqueue (value = C)



6/ After dequeue (B dequeued)



Enqueue Operation	Dequeue Operation
<code>ring[tail] := value</code> <code>advance tail</code> <code>count := count + 1</code>	<code>value = ring[head]</code> <code>advance head</code> <code>count := count - 1</code>

Circular Queue Operation

2.3.b

Vejledning til tastatur-demo-programmet DEMOKBD.COM

Bjørk Busch

Demo af BIOS-ens keyboardbuffer og tastaturflag (BBB 09/88)

Programmet viser hvorledes, der arbejdes med keyboardbufferen - opsamlingen af indtastninger - tidsforskydningen - og den senere læsning fra et program

Programmet standses ved at trykke på CTRL + ALT + SHIFT(en af dem) samtidig. Ved tryk samtidig på CTRL + ALT vil programmet foretage en læsning af et tegn.

Lad være med at taste for hurtigt, da programmet ikke reagerer med det samme. Tegn, som ligger klar til indlæsning fra program, blinker med normal farve. Tegn, som er under indlæsning / lige indlæst, har en anden farve.

Indtast hastighed 1-9 (norm 5)

hast:1 Demo af BIOS-ens keyboardbuffer og tastaturflag (BBB 09/88) 369

Programmet viser hvorledes, der arbejdes med keyboardbufferen - opsamlingen af indtastninger - tidsforskydningen - og den senere læsning fra et program

Programmet standses ved at trykke på CTRL + ALT + SHIFT(en af dem) samtidig. Ved tryk samtidig på CTRL + ALT vil programmet foretage en læsning af et tegn.

Lad være med at taste for hurtigt, da programmet ikke reagerer med det samme. Tegn, som ligger klar til indlæsning fra program, blinker med normal farve. Tegn, som er under indlæsning / lige indlæst, har en anden farve.

Hoved:0022 Hale:0028	Flag1: 00001110	Flag2: 00000011
Indhold af Keyboard Buffer		
ASCII kode	b d . 1 . j j j	
ASCII kode HEX	00 00 00 00 62 64 0D 00 31 0D 6A 6A 6A 00 00	
SCAN/extented kode	19 19 19 19 19 30 20 1C 19 02 1C 24 24 24 19 19	
Hoved !		
	Hale !	

2.3.c

Sådan kommer trætte PC-fingre i fin form

Tusindvis af mennesker bliver invalideret af at lave forkerte og ensformige bevægelser, når de skriver ved computer-tastaturet

Arbejdsskader er noget, de fleste forbinder med hårdt fysisk arbejde, tunge løft eller indånding af sundhedsskadelige stoffer på arbejdspladsen. Men et stort antal mennesker, hvis daglige arbejde hovedsagelig består i at skrive på et computer-tastatur, plages af svære lidelser i fingre og håndled. De ensformige, små bevægelser ved tastaturet kan forårsage lammelser, der kan forkoble hænderne i en sådan grad, at man ikke kan bruge dem normalt længere.

Lammelserne skyldes bl.a., at muskler og væv omkring håndroden hæver og klemmer de nerver, som har forbindel-

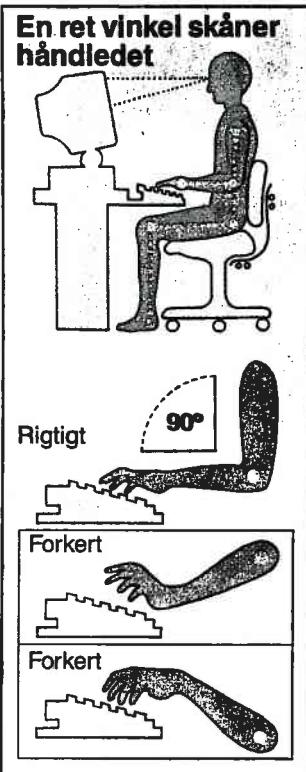
se til fingrene. De varige skader kan heldigvis forebygges, inden de bliver invaliderende, bl.a. ved hjælp af en række øvelser, der styrker hænder og håndled. Mange kontorarbejdere bliver nærmest en slags tastatur-ateleter, der skriver i rasende fart mange timer i træk. Men kun de færreste holder fingrene i form og sørger for at strække de overanstrengte muskler.

Tastatur-ateleter bør holde håndled og hænder lige i en 90° vinkel fra albuen og undgå at bøje og vride håndledene, mens de skriver. Det belaster også fingrene unødvendigt, hvis man har for vane at

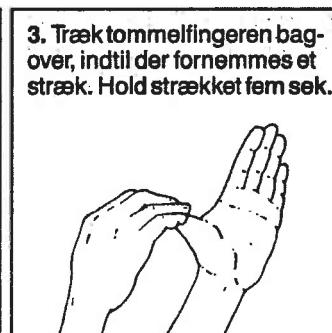
slå hårdt på tasterne. De fleste tastaturer reagerer nemlig på ganske blide anslag.

Hvis det er muligt, bør man tilrettelægge sit arbejde, så man veksler mellem at skrive på tastaturet og udføre andre funktioner – hvor man vel at mærke også sørger for at undgå at belaste håndledene.

Dr. Marvin J. Dainoff, der leder et ergonomisk forskningscenter i Ohio, USA, har udviklet træningsprogrammer for folk, som tilbringer mange timer dagligt ved et tastatur. Ifølge Dainoff bliver mange overrasket over, hvor plagsomme skader arbejdet ved en PC kan medføre. □



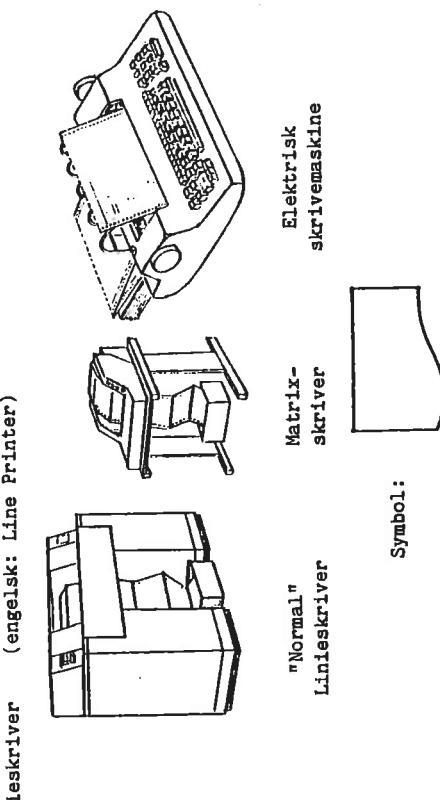
Hænder og håndled belastes mindst, hvis de holdes i en 90° ret vinkel ud fra albuen.



Simple øvelser udført dagligt kan skåne edb-arbejdere, sekretærer og skribenter for varige skader i hænder og håndled.

2.4.a

**Printere & plottere
fra
Grundlæggende databehandling
John Hansen & Carl Tange**



Hvilken type linjeskriver, der skal vælges til en aktuel maskinkonfiguration, afhænger helt af anvendelsesområderne for det pågældende anlæg. Hvis anlægget skal bruges til informationsøgning, som ved politiets motorregister, eller til beregningsopgaver med en lille ind- og uddatamængde, er en elektrisk skrivemaskine tilstrækkelig. Drejer det sig derimod om et edb-servicebureau, der f.eks. kører bogholderopgaver og lønssystemer, er der behov for en af de hurtige linjeskrivere.

Fælles for alle typer linjeskrivere er, at de er i stand til at producere skriftlige uddata styret af programmet i centralenheden. Dette betyder, at man er nødt til selv at fastlægge, hvilke uddata man ønsker skrevet, og den form, de ønskes skrevet på, d.v.s. opbygningen af den enkelte linje med mellemrum, tekster og variable oplysninger. Man er selv nødt til at definere overskriftslinier, hvilke blanke linjer der ønskes i udskriften o.s.v. Alle disse oplysninger skal så koddes i programmet, der så kan styre udskrivingen på linjeskriveren i overensstemmelse med brugerens ønsker.

Linjeskriveren kan anvendes til udskrivning på blanke lister (læporello-liste), hvor man selv skal sørge for tekster og lign., samt på fortrykte formulær (lønsejler, kontoudtog og lign.), hvor den faste tekst er fortrykt. Normalt kan en linjeskriver skrive 10 tegn pr. tomme (vandret), og 6 linjer pr. tomme (lodret).

Linjeskriveren er den ydre enhed, der størger for udskrivning i skriftlig form. Der findes en stor mængde forskellige linjeskrivere på markedet, varierende fra enkle, elektriske skrivemaskiner, der kan skrive 8-10 tegn pr. sekund, og op til avancerede linjeskrivere baseret på elektroniske principper eller laserskrivere, der kan skrive op til 20.000 linjer pr. min.

Tekniske krav Set fra et brugersynspunkt, er de tekniske krav til en linjeskriver forholdsvis enkle:

- Linjeskriveren skal være udstyret med det ønskede skriftbilledet.
- Linjeskriveren skal evt. kunne skrive med både store og små bogstaver (normalt skrives udelukkende med store bogstaver).
- Linjeskriveren skal kunne skrive med en hastighed, der står i forhold til den forevntede uddatamængde.
- Skrivebredden (antal anslag pr. linie) skal være tilstrækkelig (normalt 120-132 anslag pr. linie).
- Mulighed for at kunne skrive på løsblade istedet for endeløse baner. Der findes dog specielle formulærssæt, der giver denne mulighed.
- Mulighed for at kunne udskrive med det ønskede antal kopier.

Hvorledes linjeskriveren rent teknisk er opbygget, er set fra brugerens synspunkt, ret ligegyldigt, blot den opfylder de kapacitetsmassige krav, der stilles i form af ønsker til udskriftens udseende, mængden af udskrift m.v.

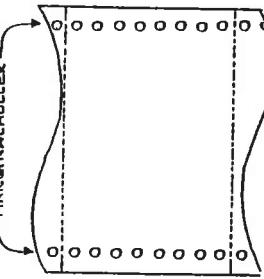
Gennemgang af nogle typer linjeskrivere findes på de efterfølgende sider.

Linjeskriveres teknik

Som tidligere nævnt findes der et meget stort antal forskellige linjeskrivertyper, varierende fra en elektrisk skrivemaskine til meget avancerede "sideskriver" baseret på anvendelse af laser-stråler. Der findes nogle egenskaber fælles for en række linjeskrivere, og disse egenskaber skal gennemgås nedenfor inden gennemgangen af de specifikke skrivetyper

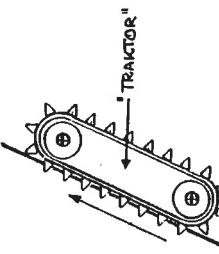
Papir-fremføring

Papir til udskrivning på linjeskrivere er normalt i endeløse baner, og er udstyret med marginalhuller.



Papir MED MARGINALHULLER

Disse huller anvendes til at trække formularen genem linjeskriveren, idet linjeskriveren er udstyret med et eller flere sæt "traktorer", der griber fat i formularens marginalhuller, og dermed styrer formularen under udskrivningen.

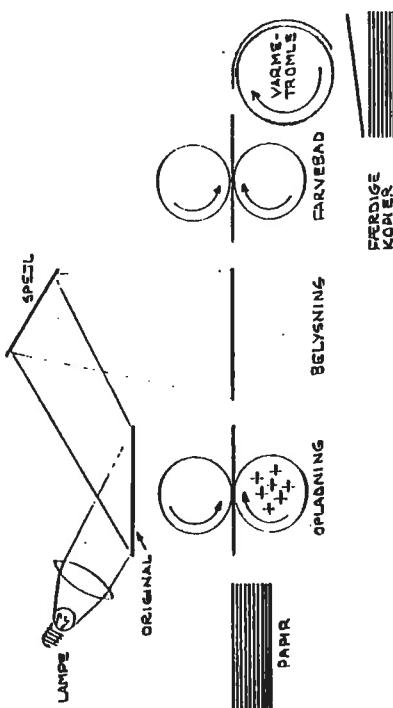


skip-funktion

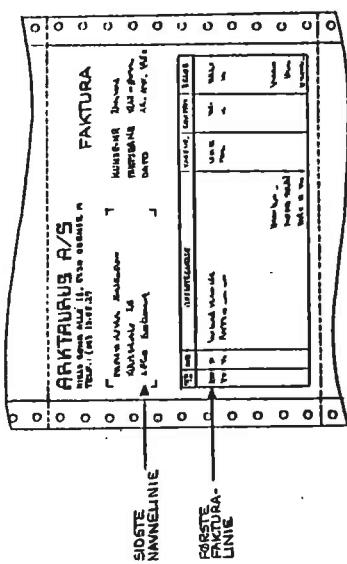
Mange af de hurtige linjeskrivere er udstyret med en "skip-funktion", hvilket vil sige en mulighed for, med høj hastighed, at køre formularen frem til den næste linje, der ønskes skrevet på.

Grundprincippet er baseret på det forhold, at papir der er opladt med en statisk, elektrisk spænding, aflades hvis det blyses.

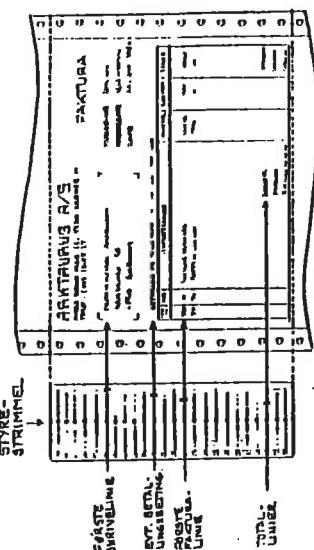
Man oplader et stykke papir med en elektrisk spænding. Dermed projiceres et billede over på det opladet papir af den original, der ønskes kopieret (bogstaver, tegninger o.s.v.). Hvor originalbilledet er mørkt (bogstaver, rammer m.v.) sker der intet med den elektriske spænding, mens papiret aflades alle andre steder. Tilbage bliver et billede på papiret af originalen i form af den elektriske spænding. Papiret sendes nu gennem et farvebad, hvor et meget fint, sort farvepulver "klauber" fast til de elektriske spændinger, og dermed danner en kopi af originalen. Endelig brandes farvepulveret fast til papiret, hvorefter kopien er færdig.



Man skal f.eks. i ovenstående faktura springe over 5 linjer fra sidste navnlinie til første linje på selve fakturaen. Man kan lade linjeskrivenere skrive 5 blanke linjer, men man kan også lade den "skipte" frem, hvilket er en langt hurtigere funktion. Skipmekanismen ligger i linjeskriveneren, men styres fra programmet i centralenheden. Til linjeskriveneren fremstilles enten en styre-strimmel (carriage-tape) svarende til den konkrete formulartype, der skal udskrives, og hvoraf der er angivet hvormange linjer, der skal springes over ved hvert hop, eller linjeskriveneren kan programmeres til at foretage disse hop. I dette tilfælde findes der et program for hver formulartype.



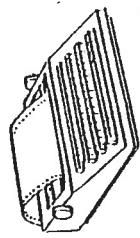
Carriage-tape



Elektrostatisk kopiering

Det elektrostatiske princip er en metode, der anvendes ved nogle af de moderne, hurtige linjeskrivere (og iøvrigt ved nogle typer fotokopieringsmaskiner).

Skrivemaskinen Den enkleste type linjeskriver er en elektrisk skrive-maschine.



Symbol:

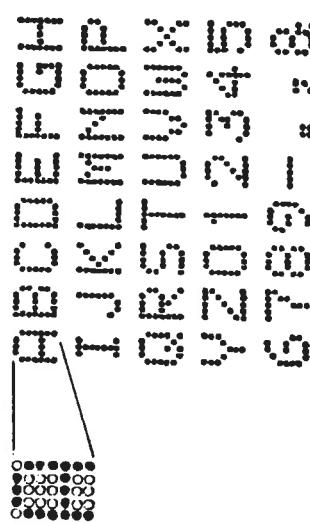


Udskrivningskapaciteten er ret ringe, ca. 8-10 tegn pr. sekund, således at denne type skrivenhed kun kan anvendes ved relativ små ud datamængder. Det skal dog anføres, at denne skrivenhed kan skrive med både store og små bogstaver, et forhold der langt fra gælder alle de hurtige linjeskriver. Den producerer et pant tryk, og kan skrive med 3-4 læselige gennemslag. Den kan i flere tilfælde fås med udskriftlige skriftbilleder (kuglehovedmaskinen), hvilket gør den velegnet til flere formål, bl.a. ved elektronisk tekstbehandling (ETB).

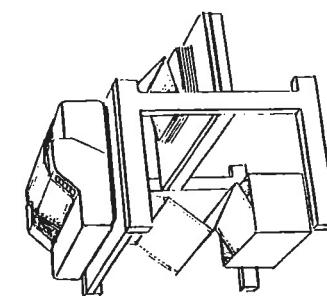
Teknikken bag en matrix-printer er et skrivehoved, der indeholder en række "nåle" sidende i en firkantet ramme (matrix).



Ved at pressse et antal af disse nåle frem mod et farvæbånd, ind mod papiret fremkommer de enkelte tegn.



Matrix-printeren Matrix-printeren er en linjeskriver-type med en udskrivningskapacitet på 80-500 linjer/minut.



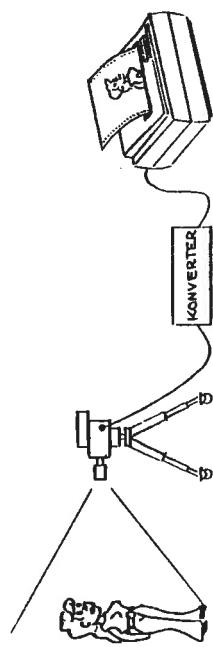
Som følge af sin konstruktion giver matrix-printeren ikke alene mulighed for at skrive med store og små bogstaver, men også mulighed for forskellige skriftbilleder i samme udskrivning, idet det udelukkende er et spørgsmål om, at den "tegengenerator", der producerer nålene frem, kan rumme de nødvendige kombinationer.

0123456789:!=?)@ABCDEFHIJKLMNOPQRSTUVWXYZ
O 1 2 3 4 5 = > ? @ B C D E Z H I J K ^ ~ _
!@2456789:!=?)@ABCDEFHIJKLMNOPQRSTUVWXYZ_!@HJH-J
012345:!=?)@ABCDEFHIJKLMNOPQRSTUVWXYZabcde fghijklmn@{
O 1 2 3 4 5 A B C D O A B C D E F G H I J K L M N @ { }

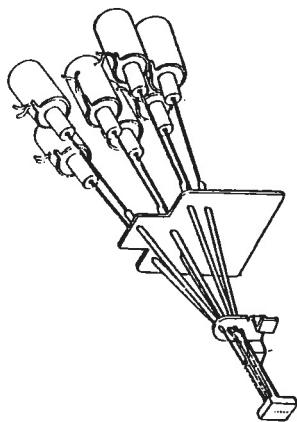
Eksempler på forskellige skriftbilleder
fra en matrix-printer

Billedfremstilling

Udover forskellige skriftbilleder kan matrix-printeren også, ved hjælp af sine nåle, "tegne" billede. Et billede kan fremstilles som en kombination af sorte og hvide prikker i forskellig tæthed (jf. avisbilleder). Matrix-printerens nåle kan anvendes i vilkårlige kombinationer, og dermed frembringe det ønskede billede. Teknikken kan anvendes både i forbindelse med et kamera, og udfra lagrede data på f.eks. magnetbånd.



Ved skrivning af en linie bevæger skrivehovedet sig hen forbi linien, og skriver de enkelte tegn 1 de ønskede positioner undervejs. Ved ønske om større skrivehastigheder kan matrix-printeren udstyres med to skrivehoveder, der hver skriver 1/2 linie, og hvor skrivningen af de to halvdele sikrer samtidigt. Matrix-printeren kan også indrettes således, at der skrives under både frem- og tilbage løb af skrivehovedet.



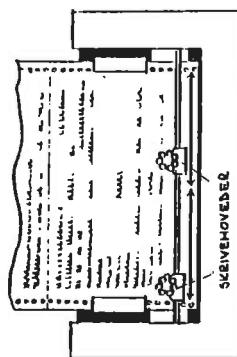
Udskrivningen sker normalt på papir i endeløse baner, idet formularen trækkes igennem linieskriveren ved hjælp af papirets marginalhuller.

Tekniske data

Matrix-printeren kan skrive fra 80 til 132 tegn pr. linie, og skriver normalt 6 linjer pr. tomme. Skrivehastighederne varierer fra 80 til 500 linjer pr. minut. Skrivekvaliteten er ikke overvældende god i øjeblikket, men da printertypen er meget populær p.g.a. prisbillighed og sin robusthed, kan det forventes, at den gradvis vil blive bedre. Der vil normalt kunne udskrives formularer med 1-2 kopier.

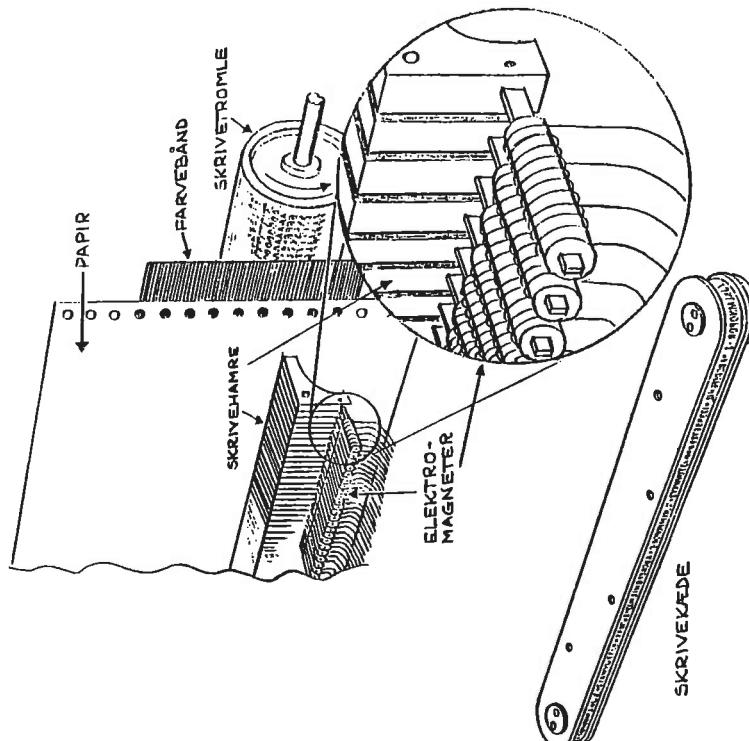
Tromle- og kædeskrivere

De to "klassiske" former for hurtige linieskrivere er kæde- og tromleskrivere. Begge typer skriver med hastigheder på 1.000 - 2.000 linjer pr. minut og 132 - 150 anslag pr. linie. Begge printertyper er baseret på den teknik, at papiret trykkes ind mod et farvebland og videre ind mod selve typen ved hjælp af en elektrisk styret hammer. Forskellen mellem de to typer er, at selve tryktyperne er monteret på henholdsvis en skrivetromle og en skrivækede.



Skrivning med kædeskriver

Ved kædeskriveren, hvor kæden roterer vandret forbi skrivelinen, skrives de enkelte tegn efterhånden som de passerer de positioner, hvor det aktuelle tegn skal skrives. Der udloses da en hammer, hvorfedt tegnet trykkes på papiret. Først passerer A'et forbi linien, dernæst B o.s.v. Når kæden har roteret en omgang er linien færdig. For at sætte skrivehastigheden i vejret, indeholder nogle skrivekæder alle tegn tre gange således, at kæden kun skal dreje 1/3 omgang for at skrive en linie.



Karaktersæt

Ved tromleskriveren er karaktersættet fast (antal typer og udseende (skriftbilleder)), idet det er meget vanskelligt at udskrifte selve tromlen. Ved kædeskriveren kan det relativt let lade sig gøre at udskrifte kæden med en anden, hvorved man kan skrive med forskellige skriftbilleder og øv. med forskellige tegn. Begge skrivertyper anvender i øjeblikket fortrensvis store bogstaver som skriftbilleder, men det må forventes, at der i tiden fremover vil ske en udvikling i retning af mulighed for både store og små bogstaver.

0123456789012345678901234567890*SEN NO. 05

Karaktersæt fra en tromleskriver

Det anvendte farvebånd er indfarvet silkebånd, ligesom ved en alm. skrivemaskine. Farvebåndet skal have samme bredde som antallet af skripositioner. Ved udskrivning af optisk læselig skrift (OCR) anvendes et farvebånd af en særlig god kvalitet, der kan give et meget præcist aftryk (engangsfarvebånd). Ved udskrivning af magnetisk læselig skrift (MICR) anvendes et farvebånd, hvor tryksærtten indeholder et magnetiserbart materiale. Sidstnævnte anvendes dog kun meget sjældent herhjemme.

Mekaniske linieskriverne kan udskrive formulærer med 3-4 læselige kopiar, hvor kopieringen sker ved hjælp af engangskarbonpapir indlagt mellem de enkelte blankeletter, eller ved hjælp af en kemiisk præparerings af blanketterne (NCR-papir).

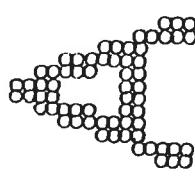
Farvebånd

Skrivningen sker på den måde, at der fra centralenheden afleveres en linie tekst til linieskriveren. Denne kontrollerer om papiret er fremme ved den korrekte linie, eller om det skal krybs frem. Når papiret er i den rigtige position og tromlen som roterer konstant har A-rækken udfør skrivelinen, udloses de hæure, der svarer til de positioner på linien, hvor der skal skrives A. Dernæst kommer B-rækken og de relevante hæure udloses o.s.v. Når tromlen er drejet en omgang, har alle tegn, bogstaver, tal og specialetegn passeret skrivelinen, og er blevet trykt, hvis de har været i den aktuelle linie.

Kopier

Laserprinter
En af de nyeste og hurtigste printertyper er laserprinter. Den er baseret på det elektrostatiske kopieringsprincip, hvor den anvendte lyskilde er en laserstråle.

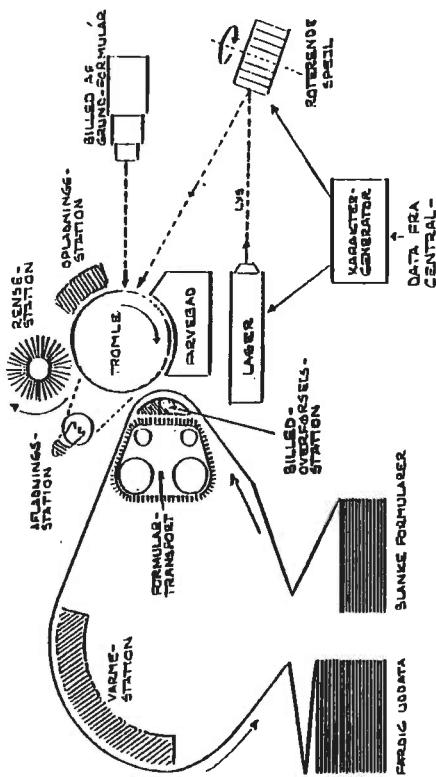
Hjertet i en laserprinter er en tromle, der oplades elektrostatisk. På denne tromle opbygges den ønskede tekst ved hjælp af en laserstråle, styret fra en "karaktergenerator". De enkelte tegn er opbygget af punkter, som laserstrålen gennem en række impulser "tegner" på tromlen.



Disse belyste punkter aflades elektrisk, og når hele siden er færdig, passerer tromlen gennem et farve-
tonerbad, hvor farven lægger sig på de afladede område
mens de ikke-belyste (=elektriske) områder af
tromlen frastøder farven. Billedet fra tromlen over-
føres nu til papiret, hvor farven bændes fast. Til
slut aflades papir og tromle for den resterende
elektriske spænding, og tromlen rennes for farve
inden overførelse af næste billede.

Samtidig med overførslen af de enkelte tegn i ud-
skriften til tromlen via laserstrålen, kan man fra
en anden lyskilde via et negativ overføre et billede
af selve formularen (stregen, fortekster m.v.) til
tromlen således, at resultatet fremtræder som trykt
på en fortrykt formular. Dette kan reducere omkost-
ningerne til fortrykte formulærer ganske væsentligt,

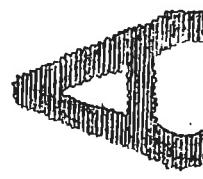
samt foregå fleksibiliteten i anvendelse af disse,
idet man kan andre formulærudsendet efter behov og
ikke efter hvor stort man har liggende.

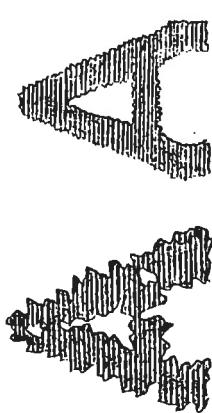


Princippet i en laserprinter

Skriftbilledet I kraft af sin konstruktion kan laserprinteren anvende vilkærlige skriftbilleder. Det er et spørgsmål om nødvendigt programmering af karaktergeneratoren, der dannet de enkelte tegn ved hjælp af punkter.

Kvalitet Skrivekvaliteten af en laserprinter er bedre end for de traditionelle linieskrivere. Farvetæheden pr. tegn er væsentlig større, hvilket resulterer i et lettere læseligt skriftbillede.





Mekanisk
printer

Laser
printer

Hastighed

Laserprinteren kan udskrive 20.000 linjer pr. min. hvilket er mere end 10 gange hurtigere end de normalt anvendte linjeskrivere. Samtidig er den, som nævnt, i stand til at producere fortrykte formulærer ud fra blanke lister, hvilket gør formulærprisen billig. Den kan dog ikke, i sagns natur, udskrive med kopier. Man er her nødt til at skrive det antal eksemplarer, der ønskes, som originaler.

Udover de nevnte linjeskrivertyper findes der en række andre af forskellige konstruktioner:

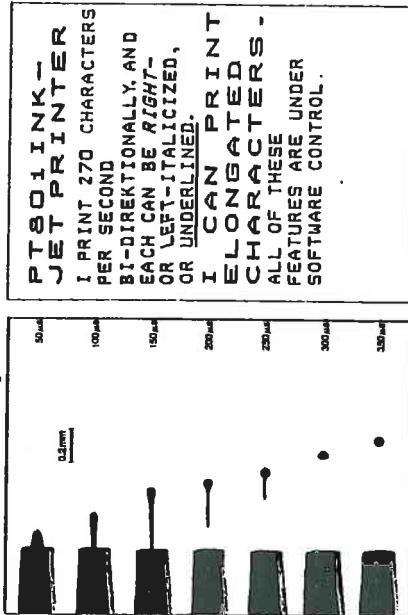
Typehjulsskriven, baseret på et eller flere typehjul, hvert indeholdende hele tegnsættet. Efterhånden som der skrives hen ad linien, flyttes typehjul og hammer til de aktuelle positioner.

Typestangsskriven, indeholdende hele karaktersættet på en typestang der enten vandret eller lodret passerer skrivelinien.

Den elektrostatiske printer, hvor der fra en film-matrix overføres billeder af de enkelte tegn til en elektrisk opladet formular, hvorefter der påføres farve, som slutteligen brændes fast til formularen.

Termoskriven, der ved hjælp af et skrivenhoved med varmetråde overfører skriften til varmefølsomt papir.
Blækskriven. (Jet Ink Printer) Ved hjælp af en dyle udsendes en fin stråle blækpartikler. Denne stråle styres elektrisk, og kan derigennem bringes

til at tegne de enkelte tegn på papiret. Der udsendes ca. 100.000 blækpartikler pr. sekund, og skriveren kan skrive op til 1000 tegn/min. Da styringen af blækpartiklerne sker via et program, kan skriveren arbejde med alle størrelser (i det omfang programmet kan styre det). Skriveren har også den store fordel at arbejde omtrænt lydløst. Overflødige blækpartikler suger automatisk tilbage igen. Blækskriveren kan skrive med forskellige farver.



Eksempel på blækskriver (jet ink printer), teknik og skrivekvalitet.

Kurveteegnere (engelsk: Plotter)	<p>I en anden konstruktion, plantegnere, er papiret placeret plant på et tegnebord. Pennen kan her bevæge sig i to retninger, vandret og lodret, og kan dermed generere den ønskede udskrift.</p>	<p>Plantegnere</p> <p>Kurveteegnener er en ydre enhed, der kan producere uddata i form af stregtegninger, kurver, diagrammer og lign. Kurveteegnener anvendes i forbindelse med grafisk databehandling, hvor man ønsker resultatet tegnet efter endt konstruktion via en dataskærm.</p>	<p>Pennen</p> <p>Princippet i en kurveteegner er, at en pen kan bevæge sig vandret og lodret hen over en papirlade, styret af de data, der ønskes afbildet. Resultatet af disse bevægelser er kurver af vilkårligt udseende, altså alle de former for grafisk afbilding, man kan ønske. Kurveteegnener kan også skrive den til afbildningen hørende tekst i samme arbejdsgang.</p>	<p>Symbol:</p> 	<p>Princippet i en kurveteegner er, at en pen kan bevæge sig vandret og lodret hen over en papirlade, styret af de data, der ønskes afbildet. Resultatet af disse bevægelser er kurver af vilkårligt udseende, altså alle de former for grafisk afbilding, man kan ønske. Kurveteegnener kan også skrive den til afbildningen hørende tekst i samme arbejdsgang.</p>	<p>Anvendelse</p>	<p>I den mest almindelige type, tromletegneren, ligger papiret på en tromle, der kan rotere. Pennen er monteret på en bjælle, og kan bevæge sig tværs over papirbunden. Ved at kombinere de to bevægelser, opnår man at kunne tegne i vilkårlige retninger. Under flytninger kan pennen have fra papiret.</p>
		<p>Tromletegner</p>			<p>Endelig kan kurveteegnener opbygges efter et elektrostatisk princip, hvor der ikke direkte tegnes på papiret, men hvor tegningerne opbygges af elektriske punkter, der så gennem fremkalder-proces omdannes til en læshar kurve. Denne type kurveteegner kræver speciel papir.</p>		
					<p>Selve pennen kan være en tuschpen, spritpen eller en avanceret udgave, en "bjælk-sprøte". Afhængig af størrelsen kan en kurveteegner arbejde med flere farver, ved faste penne op til 4 farver, ved blæk-sprøjten, 4 grundfarver der kan kombineres til over 100 nuancer. Pennen kan udskriftstilpasses den ønskede stregtykkelse.</p>		
					<p>Kurveteegnener kan arbejde med en nøjagtighed bedre en 0,1 mm, d.v.s. udstyret kan tegne mere nøjagtigt, end det er muligt manuelt.</p>		
					<p>Tegnehastigheden afhænger stærkt af udstyrets art og størrelse, og af hvor individet tegningen er, men en "normal" tegning kan udskrives på 5-15 min. Det kan derfor være hensigtsmæssigt at udstyre kurveteugnen med et internt lager og en processor, således at den ikke skal belaste CPU'en under hele tegneprocessen.</p>		
					<p>Sammen med en grafisk dataskærm og relevant programmel udgør kurveteugnen et overordentligt godt udstyr til konstruktion og rentegning. Når man er færdig med at gennemarbejde en tegning på dataskærmen, kan den målsettes automatisk og udskrives på kurveteugnen. Tegninger og lign. kan lagres på et af datamatens ydre lagre, således at man ved ændringer kan kalde tegningen frem på skærmen, foretage ændringer, og få en ny rentegning udskrevet.</p>		

2.4.b

Control Extensions

1 Control Sequence

The BJ-300/330 printers support the following 10 extension control sequences.

CSP	Code Count-Low Count-High [Parm ₁ ... Parm _n]
or	
ESC X'5B'	Code Count-Low Count-High [Parm ₁ ... Parm _n]
X'1B5B'	Code Count-Low Count-High [Parm ₁ ... Parm _n]

Where:
CSP (Control Sequence Prefix) is a 2-byte value that introduces the control sequence. CSP is assigned ASCII values 27 and 91 (1BH, 5BH), which represent ESC [.

Code is a 1-byte value that designates a specific control function.

Count-Low is a 1-byte value that is the least significant 8 bits of the parameter count bytes in the control sequence function.

Count-High is a 1-byte value that is the most significant 8 bits of the parameter count bytes in the control sequence function.

Parm₁ to Parm_n are the parameters that contain the control settings. These parameters can be decimal or hexadecimal in format and each can consist of one or more bytes. Some controls do not have parameters.

1.1 Control Specifications
Format controls include the basic ASCII control codes from the C0 set, the escape sequence functions and the control sequence functions.

1.2 Single-byte Controls

The BJ-300/330 printers support the following single-byte controls.
When Character Set 1 is active, most of the control codes exist from ASCII 80H to 9FH. These control codes are enclosed in parentheses below.
The supported control codes are as follows.

Code	Decimal (*)	Hex (*)	Function
NUL	0, 127 (128)	00, 7F (80)	Null
BEL	1 (135)	07 (87)	Bell
BS	8 (136)	08 (88)	Backspace
HT	9 (137)	09 (89)	Horizontal Tab
LF	10 (138)	0A (8A)	Line Feed
VT	11 (139)	0B (8B)	Vertical Tab
FF	12 (140)	0C (8C)	Form Feed
CR	13 (141)	0D (8D)	Carriage Return
SO	14 (142)	0E (8E)	Shift Out, double-wide on
SI	15 (143)	0F (8F)	Shift In, condensed mode on
DC1	17 (145)	11 (91)	Device Control 1, printer select (XON for serial interface)
DC2	18 (146)	12 (92)	Device Control 2, cancel condensed mode and return to 10 CPI
DC3	19 (147)	13 (93)	Device Control 3, same as NUL (XOFF for serial interface)
DC4	20 (148)	14 (94)	Device Control 4, cancel double-wide if set with SO
CAN	24 (152)	18 (98)	Cancel
SP	32	20	Space
RSP	256	FF	Required Space

(*) The control codes in parentheses can only be used when character set 1 is selected. These characters represent text characters in character set 2.

Note: For PC compatibility, the above control codes from ASCII 00H to 1FH can also be invoked by preceding the control code with ESC (1BH).

2 Escape Sequence

This section describes the control syntax and semantics of the escape sequence functions.
The format of the escape sequence functions is as follows.

ESC F n1...n

"F" can be any one of the following:

- ASCII 0 (00H) to 31 (1FH) (same as for single-byte control codes, see the previous section)
- ASCII 45 (2DH)
- ASCII 48 (30H) to 126 (7EH) (excluding 59 (3BH) and 91 (5BH))
- ASCII 51 (5BH) (see section "Control Extensions: Control Sequence" on page 1-23).

If "F" is not one of the above, the escape sequence will terminate without performing any operation.

The BJ-300/330 printers support the following escape sequences.

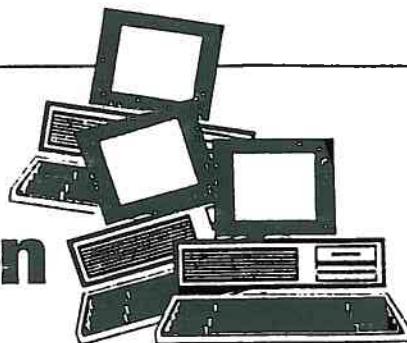
Escape Sequence	ASCII Code Dec	Hex	Function
ESC - n	45	2D	Auto Underscore (1=on, 0=off)
ESC 0	48	30	Set 1/8 Inch Line Spacing
ESC 1	49	31	Set 7/72 Inch Line Spacing
ESC 2	50	32	Invoke Variable Line Spacing
ESC 3 n	51	33	Set Graphics Line Spacing (n/216: changed with ESC V)
ESC 4	52	34	Set Top of Form
ESC 5 n	53	35	Set LF Auto Mode on CR (1=on, 0=off)
ESC 6	54	36	Select Character Set 2
ESC 7	55	37	Select Character Set 1
ESC :	58	3A	Set 12 CPI
ESC =	61	3D	Character Font Image Download
ESC A n	65	41	Set Text Line Spacing (n/72)
ESC B n...n0	66	42	Set Vertical Tabs n1...n
ESC C 0m	67	43	Set Page Length (m inches)
ESC C n	67	43	Set Page Length (n lines)
ESC D n..n0	68	44	Set Horizontal Tabs n...n
ESC E	69	45	Begin Emphasized Print
ESC F	70	46	End Emphasized Print
ESC G	71	47	Begin Double Strike Print
ESC H	72	48	End Double Strike Print
ESC I n	73	49	Change Font (n selects the font)
ESC J n	74	4A	Relative Move Base Line (n/216: changed with ESC V)
ESC K n1 n2 n...n	75	4B	Normal Density Bit Image Graphics (letter quality speed)
ESC L n1 n2 n...n	76	4C	Dual Density Bit Image Graphics (letter quality speed)
ESC N n	78	4E	Set Skip Over Perforation
ESC O	79	4F	Reset Skip Over Perforation
ESC P n	80	50	Proportional Spacing (1=on, 0=off)
ESC Q n	81	51	Deselect on Positive Query Reply
ESC R	82	52	Set Default Tab Racks (vertical and horizontal)
ESC S n	83	53	Start Subscript (1) or Superscript (0)
ESC T	84	54	End Subscript or Superscript
ESC U n	85	55	Set Print Direction (0=bi-directional, 1=uni-directional)
ESC W n	87	57	Double wide Continuous Mode
ESC X n1 n2	88	58	Set Horizontal Margins (n1=left margin, n2=right margin)
ESC Y n1 n2 n...n	89	59	Dual Density Bit Image Graphics (letter quality speed)
ESC Z n1 n2 n...n	90	5A	High Density Bit Image Graphics (letter quality speed)
ESC [@ n,n,m...m4	91-94	5B-5D	Set Presentation Highlight

Note: The BJ-300/330 printers support the original Canon commands, FS C B and FS C J.

Escape Sequence	ASCII Sequence	ASCII Code Dec	Hex	Function
ESC [f n,n,m,m4	91 70	5B 46		Page Presentation Media
ESC [l n,n2	91 73	5B 49		Select Font
ESC [k n,n2	91 75	5B 4B		Set Initial Condition
ESC [s n,n2	91 83	5B 53		Set Vertical Margins
ESC [t n,n2	91 84	5B 54		Set Code Page
ESC [b n,n2	91 92	5B 5C		Set Vertical Units
ESC [d n,n2	91 100	5B 64		Set Print Quality
ESC [g n,n2	91 103	5B 67		High Resolution Graphics
ESC [- n,n2	91 45	5B 2D		Score Selection
ESC [n,n2	92	5C		Print All Characters (including below 20H)
ESC ^	94	5E		Print Single Character Under Hex 20
ESC _ n	95	5F		Continuous Overscore Mode (1=on, 0=off)
ESC d n,n2	100	64		Relative Move Inline Forward (n/120th inch)
ESC e n,n2	101	65		Relative Move Inline Backward (n/120th inch)
ESC i	106	6A		Stop (and go offline)
ESC m	110	6E		Select Aspect Ratio

2.5.a

TÆT PÅ PC'EN



T

Trimming af harddisken

Anvisninger på, hvordan man holder harddisken "velsmurt".**Af Martin Rørbye Angelo**

Når man arbejder med sin PC, synes det i de første måneder som om alt går med en svimlende fart. Men snart vænner man sig til det, oginden længe synes alt at tage en forfærdende tid. Hvis ikke man vindste bedre, ville man tro, at ens PC ikke var den samme, som den man købte. Den er meget langsommere.

Der er naturligvis rigtigt, at man vænner sig til hastighed - *performance*, men det er også rigtigt, at ens PC bliver langsommere med tiden. Det er ikke fordi den regner langsommere, men fordi man, hvis man ikke tænker på det, kan benytte sin PC på en sådan måde, at der bliver mere og mere arbejde for PC'en for at udføre de samme opgaver.

Vi vil i denne artikel se på, hvorledes man kan trimme sin PC - dog mest i forbindelse med harddisken. Det tager tid at hente data ind fra harddisken, det tager tid at læse programmerne ind, og det er den værste slags tid, der findes: *ventetid*.

Øget hastighed uden omkostninger

Der er en række ting, man kan gøre for at bringe overførstiden af data på sin harddisk ned uden at indkøbe nye omkostningskrævende komponenter. Noget kræver dog anvendelse af (ikke særlig dyr) software, men det meste er en række grundlæggende regler for god opførsel overfor sin harddisk.

De 14 faktorer, der bestemmer harddiskens hastighed

Der er 14 grundlæggende faktorer, der bestemmer den hastighed, hvormed harddisk-systemet henter information fra disken til arbejdslageret.

De første fire er betinget af det "isenkram", der sidder i PC'en. Isenkrammet kan naturligvis udskiftes, men det koster penge.

De næste to er bestemt af *low level*-formateringen - dog begrænset af "isenkrammet". De sidste otte er kun bestemt af brugerens, og det er derfor her, vi skal finde den gratis hastighedsforøgelse.

Det tager tid at flytte data gennem elektronikken, men den er minimal. Det altaførende tidsforbrug ligger i læse/skrivehovedets bevægelser. De fleste af de følgende råd og vejledninger omhandler derfor forskellige måder at begrænse læse/skrivehovedets bevægelser på.

CPU-hastighed (1). Højere CPU-hastighed øger hele PC'ens reaktionshastighed og derfor også dataoverførsel fra harddisken. Det kræver dog investeringer i *accelerator-boards* eller i udskiftning af CPU'en.

Hukommelses- og hjælpe-kredselementerne forbliver de oprindelige (langsomme) versioner, så man får ikke nær så meget ud af at udskifte fx en 80286-CPU med en 80386SX-ditto som ved at købe en 80386SX-maskine. Men det er stadigvæk en glimrende og prisbillig måde at skaffe sig ekstra hastighed på i en lidt ældre PC.

Middelsøgetid (2). Jo kortere middelsøgetid en harddisk har, jo hurtigere bevæger den læse/skrivehovedet, dvs. jo hurtigere finder den de enkelte sektorer. Der er dog intet, der kan ændre middelsøgetiden ved et givet drev. Hvis man ønsker at nedsætte middelsøgetiden, må man købe en ny, hurtigere harddisk.

Data transfer rate (3). En hurtig harddisk kan forsinkes ved langsom overførsel gennem harddisk-controlleren (se PC World nr. 6/92 si. 96f). Ændring af en given PC kræver derfor indkøb af et nyt controller-board. Udskiftning af controlleren fra fx en ST-506/412-type til en RLL-type kan dog øge hastigheden væsentligt, samtidigt

med at harddisk-kapaciteten øges - ofte med 50 pct. Således kan en 40 MByte harddisk med ST-506-controller blive til en 60 MByte med RLL-controller - uden andlen investering. Men det fordrer en *low level*-omformatering samt at harddisken er konstrueret med tilstrækkeligt gode tolerancer til at kunne lagre den øgede datamængde.

Interleave (4). Når der læses fra harddisken, kan systemet normalt læse data hurtigere end controllerkortet kan modtage og behandle dem. Bl.a. derfor anvendtes tidligere meget *interleave factor*, der er det antal sektorer, der springes over, hver gang der er læst én sektor. Nyere hurtigere maskiner har ikke det problem og anvender derfor ofte interleave 1:1. Ved at optimere interleaving, kan den effektive søgetid bringes ned. Dette kan gøres ved fx Norton's funktion Calibrate.

Track buffering (5). Man kan læse et helt *track* (spor) ind i en buffer, selv om PC'ens controller ikke kan håndtere 1:1 interleaving. Da sporet da læses sekventielt, går det væsentlig hurtigere, end hvis læsningen skete i den rækkefølge, interleave-faktoren ellers foreskrev. Track-buffering er imidlertid i reglen en egenskab ved det enkelte controllerkort, hvorfor et nyt må købes, hvis man vil øge hastigheden således.

Antal sektorer pr. cylinder (6). Jo flere plader pr. cylinder der er, jo mindre skal læse/skrivehovedet (rettere: læse/skrivehovederne) bevæge sig under læsning og skrivning. Det betyder, at med en given kapacitet vil en harddisk med flere plader være hurtigere end én med færre, men større plader. Men med en given harddisk kan dette jo ikke ændres. Jo flere sektorer pr. spor, jo flere data kan læses uden at skulle flytte læse/skrivehovedet til næste spor. Vil man øge antal sektorer pr. spor, kræver det dog udskiftning af controllerkortet.

RAM-disk support (7). Øste læste filer kan lægges i et RAM-lager, derved nedsættes den mængde data, der skal læses fra harddisken.

DOS buffer-definitlion (8). Såfremt antallet af DOS-buffere er valgt optimalt, nedsættes behovet for at læse de samme filer igen og igen.

Caching (9). En mere avanceret metode til RAM-lagring af øste benyttede databidder end DOS' "buffers" er data-caching. Caching-programmer leveres fx som standard med DOS 3.3 eller senere

Path-kommandoen (10). Ved omhyggelig (begrænset) anvendelse af DOS' path-kommando, kan søgetiden efter lokalbiblioteker holdes nede.

Fastopen-kommandoen (11). Ved at køre *fastopen*, vil DOS huske, hvor de sidst anvendte lokalbiblioteker er placeret og behøver derfor ikke at lede efter dem, men kan gå direkte til dem. Det er en ny kommando i DOS 3.3.

Design af træstrukturen (12). Ved at udforme sit bibliotekstræ med omtanke, kan man holde sin harddisk på samme hastighed, som da den var ny.

Lokalblibliotekslayout (13). Ved at sikre sig, at lokalbibliotekerne er lagt i en eller få cylindre, kan harddiskens hastighed ligeledes holdes på samme niveau, som da den var ny.

Fil-fragmentering (14). Jo mindre fragmenteret en fil er, jo færre cylindre den er spredt over, jo mindre bevægelse skal læse/skrivehovedet gennemføre for at læse den.

Hvorledes øger jeg min harddisk performance?

Af disse 14 faktorer kan man måske anvende de ti sidste på en eksisterende PC uden at skulle investere i nyt isenkram, og det er disse ti faktorer, vi vil behandle.

Alle reglerne i de følgende tre afsnit går derfor ud på ét eneste: at formindskе antallet af mekaniske bevægelser, som læse/skrivehovedet skal foretage.

Optimering på low-level formateringsniveau

Når den "blanke" harddisk formate-

Harddiskene bliver mere og mere kompakte. Denne Maxtor MXT har en kapacitet på 1,2 Gb og en gennemsnitlig søgetid på 8 ms.

res til anvendelse under DOS-styresystemet, vælges en række parametre, der har væsentlig indflydelse på harddiskens tidsforbrug. Hvis hastighedsforøgelse er et krav eller hvis parametrene simpelt hen er valgt forkert fra starten, kan man forsøge sig med en ny low level-formatering.

Interleave

Sektorer, skrevet i rækkefølge på harddisken, kommer ikke til at ligge i rækkefølge, fordi systemet "fletter" disse sektorer for at give tid til at behandle en læst sektor. Inden den næste sektor læses. Dette kaldes interleave. En Interleave-faktor på 6:1 betyder, at der er fem "fremmede" sektorer mellem en bestemt sektor og den næste sektor, der skal læses, når hele den pågældende fil læses.

Såfremt faktoren er rigtigt valgt, betyder det, at den pågældende PC anvender samme tid til at behandle den læste sektor, som det tager harddisken at rotere en vinkel svarende til knapt fem sektorer og derfor er klar til at læse, når sektor nr. seks kommer ind under læse/skrivehovedet.

Såfremt Interleave-faktoren er valgt for stor, skal systemet altså vente længere end nødvendigt. Såfremt Interleave-faktoren er valgt for lille, skal harddisken dreje en hel omgang, før næste sektor læses med tilhørende tidsspild.

Hvis man har mistanke om, at harddiskens interleave-faktor er sat forkert, må man først konsultere den dokumentation, man sikk med eller konsultere sin forhandler. En XT havde som standard 6:1, mens en AT har 3:1. Der er i dag så mange forskellige kombinationer af processorer og hurtige eller langsomme harddiske, at det ikke er til at sige, hvilken Interleave-faktor man kan forvente en nyleveret PC har. Grundet den stadig større andel af hurtige harddiske og controllers, er andelen af PC'er med interleave 1:1 stadig stigende.

Vil man eksperimentere, må man



Foto: Dansk Perfekt Teknik A/S

low level-reformatore harddisken, high level (DOS) -formatore og så køre en hastighedstest (som Benchmark e.l.) for at se, om man øgede hastigheden.

Det er et næsten uoverkommeligt arbejde. Der findes programmer til optimering af interleave, der samtidig vil reformatore harddisken til den optimale interleave, såfremt det ønskes - uden at slette indlæste data. Den reformater én cylinder ad gangen og gemmer data imens i arbejdslageret. Et sådant program er fx Norton Utilities 5.0 eller senere, der indeholder et program kaldet *calibrate* som gør netop dette. Programmet skal dog genkende formateringen for at kunne gennemføre analysen og reformateringen, så det er ikke alle PC'er, der kan anvende dette eller tilsvarende programmer; men nok de fleste.

Som hovedregel er der kun væsentlige hastighedsforøgelser at hente her, hvis interleave-faktoren er sat for lavt. Hvis den er sat en eller to sektorer for høj, er det marginalt, hvad der indtjenes ved en ændring. Husk at tage backup, før disse øvelser begyndes.

Optimering i DOS-miljøet

I det følgende vil vi se på programmer og kommandoer, der under DOS-styresystemet vil kunne øge den hastighed, hvormed harddisken øjensynligt opererer.

Når vi siger øjensynligt, er det fordi disse hjælpemidler mindsker behovet

for at læse fra og skrive til harddisken eller mindsker søgetiden efter et bestemt bibliotek. Den egentlige læse- og skrivetid ændres ikke herved.

RAM-disksupport

En RAM-disk er en del af PC'ens almindelig elektroniske arbejdslager, der er afsat til at opbevare data i en sådan form, at det for brugeren ligner en disk.

RAM-diske blev mere almindelige, da 80286-baserede PC'er kom frem, idet 80286 kan adressere 16 MByte, mens 8088 kun kan adressere 1 MByte. DOS, der oprindeligt var skabt til 8086 eller 8088, kan heller ikke adressere mere end den ene MByte, hvorimod 80286 har mulighed for at adressere de 15 MByte, der ligger udenfor DOS' rækkevidde.

Det er det, der kaldes *extended memory* - engelsk for udvidet arbejdslager. Den er extended (forlænget) ud over DOS' rækkevidde. Fra DOS 3.0 findes mulighed for at sætte en RAM-disk op i denne extended memory, kaldet *udisk* (IBM DOS) for virtual disk eller *ramdisk* (Microsoft DOS).

Herudover kan en PC forsynes med hukommelse, der ikke direkte kan adresseres af CPU'en, men gennem et 64 KByte"-vindue", der ligger inden for den første ene MByte, og derfor kan ses af såvel 8088 som 80286. Denne form for hukommelse kaldes *expanded memory* - engelsk for ekstra arbejdslager. Kommunikationen gennem dette 64 KByte-vindue varetages af et hukommelsesstyringsprogram - ▷

TÆT PÅ PC'EN

ofte refererer til som EMMS (*Expanded Memory Manager System*).

Da denne form for hukommelse er tilgængelig via DOS, kan den derfor anvendes direkte af de fleste programmer, som almindelige PC-brugere anvender.

Også i Expanded Memory kan man sætte en RAM-disk op, det gøres dog ikke af DOS, men derimod af det pågældende EMMS, men for brugerne er de to systemer ikke til at skelne fra hinanden. Begge de to typer RAM-diske mister deres information, når der slukkes for PC'en, hvorfor de kun kan anvendes til program- og data-stumper, der skal lagres midlertidigt under programafviklingen. Således opretter nogle en RAM-disk alene for at kunne lægge DOS' eksterne kommandoer herop, hvorved afviklingen af DOS-kommandoer bliver væsentlig hurtigere.

RAM-diske er derfor bedst egnet til lagring af dele af programmer eller hele programmer, mens de afvikles. Således vil regnskabsprogrammet PCPlus afvikles meget hurtigere fra RAM-disk, idet programmet er opbygget på en sådan måde, at det ustandseligt skal ud på harddisken og hente nye dele af programmet.

Ved at lægge hele programmet og tilhørende data op i en 3 MByte RAM-disk, benyttes harddisken ikke under hele brugen, og det batch-program, der lægger program og data op i RAM-disken (når programmet startes), sørger også automatisk for at lægge data tilbage på harddisken ved afslutning.

Batch-programmet lægger ligeledes en kopi af data på et andet diskdrev ved afslutning af brugen, hvorved egentlig backup kan foretages relativt sjældent, idet headcrash på to forskellige harddiske er så uædvanligt, at man kan roligt kan se bort fra det.

Visse tekstbehandlingsprogrammer giver brugerne mulighed for at specificere, hvorledes programmet skal finde den midlertidige hukommelse, det har brug for under redigering (kalder fx Virtual Memory Setup). Default er ofte harddiskens rod-bibliotek, mens det vil spare en del harddisk-læse- og skriveaktivitet, hvis en RAM-disk specificeres.

DOS buffer-definition

DOS har faktisk selv et system til minimering af harddisk (eller floppy-disk) læsninger. Det er *buffers=nn*, i config.sys, hvor nn er et tal fra 1 til 99. Tallet angiver, hvor mange sektorer på 512 Byte, der gemmes i hukommelsen og derfor ikke behøver blive læst fra disken, når de skal anvendes igen. Tilsvarende vil dette antal sektorer ikke blive skrevet på disken, for buffer-pladsen skal benyttes af andre data eller en bestemt tid er forløbet uden diskaktivitet.

Herved undgås at en sektor, hvor der måske kun rettes nogle få Byte ad gangen, skrives på disken, hver eneste gang en rettelse finder sted; men først skrives, når programmet kalder nye sektorer ind.

Hvis kommandoen *buffers=* ikke anvendes, sætter XT'er den til 2 og AT'er den til 3. Dette afspejler hukommelsens begrænsning fra PC'ens

barndom, hvor de var født med 64 KByte alt iberegnet.

I dag, hvor man næppe kan købe en PC med under et par MByte, er *buffers=15* eller *buffers=20* et mere optimalt valg. Hver buffer fylder 528 Byte - 512 Byte til hukommelsen og 16 Byte til at holde styr på den med.

Caching

Det bedste ved RAM-diske og DOS-buffers kombineres i *disk-caching*. Når et program giver besked om at læse en programdel eller datadel fra harddisken, søges i caching-området (i arbejdslageret) først. Disk-caching kan holde MegaByte af program og data i hukommelsen, såfremt extended eller expanded memory benyttes.

Forskellen mellem DOS-buffere og disk-cache er blot, at da disk-cache opbevarer hele programblokke og

TÆT PÅ PC'EN

ikke kun sektorer, er søgningen væsentlig hurtigere og mere logisk.

Derfor forsinker en disk-cache på 1 MByte ikke søgningen efter en bestemt programdel væsentligt, mens en tilsvarende DOS-buffer dels ikke kan lade sig gøre (maksimum er buffers=99 svarende til ca. 50 KByte) og dels ville tage evigheder, da bufferne søger sektor for sektor.

Derudover vil DOS-bufferne forbruge af den maksimale 1 MByte, som DOS kan adressere, hvorafl vi som brugere kun kan anvende de 640 KByte, hvorimod en disk-cache kan lægges i såvel extended som expanded memory.

Eksempelvis leveres IBM-cache med nogle af PS/2-systemerne. Det ligger som en skjult fil på systemdisketten og kan sætte en disk cache på 16-512 KByte inden for DOS-hukommelsen (system memory) eller fra 16 KByte-15 MByte i extended memory (IBM-cache kan ikke håndtere expanded memory).

Cache-programmer leveres med andre hjælpeprogrammer som Windows og Norton Utilities. I DOS 5.0 introduceredes en cache kaldet smartdrv.sys. Man kan sætte cache-hukommelsen så stor man vil, men på et tidspunkt bliver den tid, der skal til at søge gennem cache-hukommelsen, hver gang man søger en information på harddisken, så stor, at den samlede hastighed falder. En værdi over 256 KByte, men under 1 MByte, placeret i extended hukommelse synes at være et godt startpunkt.

Path-kommandoen

Praktisk taget alle, der anvender harddisk, kender DOS-kommandoen path, der beder DOS om at lede i en række specifiserede biblioteker, såfremt en kommando ikke findes i det bibliotek, kommandoen indtastes fra.

De fleste opbygger en under tiden særdeles kompliceret path (søgesti) i logisk rækkefølge: Først rodten i C:, dernæst lokalbiblioteker - ofte i alfabetisk orden - ned ad én streng, ned ad den næste osv. Dernæst disk D:, hvis flere logiske drev anvendes, osv. De færreste tænker på, at DOS rent faktisk søger i samme orden. Så søgestien bør opbygges således, at de mest anvendte lokalbiblioteker ligger først fx den, der indeholder com-

mand.com og DOS' eksterne kommandoer. Sidst i søgestien kommer så de lokalbiblioteker, der indeholder "elegante", men sjældent brugte batch-filer. Søgestien må i øvrigt højest være 127 tegn lang.

Fastopen-kommandoen

Fastopen-kommandoen er en nyskabelse i DOS 3.3. Ved enten direkte eller via autoexec.bat at kalde kommandoen fastopen c: = nnn, vil DOS huske de sidste nnn-klyngenumre på drev c:. Første gang man beder om at læse filen C:bib1\bib2\bib3\bib4\fil.dat, må DOS lede gennem hele denne path. Næste gang ved den, hvor på harddisken filen fil.dat ligger og kan derfor gå direkte til den første klynge og sparer derved en del søgetid.

Fastopen er nnn et tal fra 10 til 999. Hvis Fastopen eksekveres uden angivelse af nnn, sætter DOS nnn = 034.

Bibliotekets træstruktur

Når DOS søger en fil, må den springe fra lokalbibliotek til lokalbibliotek i sin søgen ned igennem træet. Derfor kan følgende "leveregler" udledes: Først må filer, der ikke anvendes ret meget, lægges langt nede i træstrukturen, mens filer, der adresseres ofte fx data, bør ligge nærmere rodten. Ofte lægges programmer i første niveau under rodten, og data lægges et eller to niveauer herunder. Man børde, for at minimere søgetiden, gøre det modsatte.

Af hensyn til forenklet backup, kan det være en fordel at have sine programmer på fx logisk drev E: og alle data på logisk drev D:, idet logisk drev C: reserveres til systemfiler såsom DOS og hjælpeprogrammer, der ikke bruges dagligt. Herved mindskes risikoen for, at et vildtløbende program kan ændre noget og umuliggøre at harddisken boot'er.

Da data ikke fylder ret meget, såfremt, der ikke er tale om grafik-filer, kan mange klare sig med et mindre drev til data og et noget større til programmer.

Tekstbehandlingsprogrammer kan fx godt i dag fyldte over 5 MByte, mens datafiler hertil næppe kommer

op over nogle få MByte for de flestes vedkommende. Derfor foreslås det, at programmet lægges på fx e:\txtprg\..., mens data lægges på d:\txtdata\..

Da man ikke behøver at lave backup af programmerne, man har jo originaldisketterne, kan man nøjes med den simple kommando: lav backup af D:

Dernæst må størrelsen på lokalbiblioteket holdes nede. Et bibliotek optager 32 Byte på harddisken. En 512 Byte-sektor kan derfor indeholde 16 biblioteker. En standard formateret harddisk under DOS 3.0 eller senere har klynger (harddiskens mindste adresserbare enhed) på fire sektorer. En klynge kan altså indeholde op til 64 navne på filer eller lokalbiblioteker i én klynge. To af disse går til DOS' egen information, kendt som ".." og "...", så for at sikre sig, at alle navne i et bibliotek holdes inden for én klynge, bør man ikke have mere end 62 navne i sit bibliotek.

Ved at holde alle navne i én klynge, sikrer man sig, at læse/skrivehovedet ikke skal flyttes til en anden cylinder under søgningen og sparer derved søgetid. Sorteres bibliotekerne regelmæssigt, og holdes bibliotekerne først på harddisken - før program og datafiler, opnås optimal hastighed. Denne sorterings kan foretages meget enkelt og hurtigt med hjælpeprogrammer som Norton Utilities. Dets Speed Disk-program spørger i hvilken rækkefølge, man vil have sine filer lagt efter defragmenteringen.

Endelig, som sidste regel: Hold antallet af niveauer så langt nede som muligt. Helst kun to eller tre niveauer. Hvis mange data skal hentes fem eller syv niveauer nede, kan læse/skrivehovedets søgetid alene løbe op i sekunder.

Ved at holde data og programmer adskilt på hver sit logiske drev, spares ofte et par niveauer, hvor man ellers ville fristes til at lægge biblioteket som et underbibliotek under selve programmet.

Hvordan lægges lokalbiblioteker ind?

Hvis lokalbibliotekerne lægges fysisk samlet på harddisken - i kun et par cylindre langs harddiskens ydre kant ▷

TÆT PÅ PC'EN

- kan læse/skrivehovedets søgetid reduceres meget. Det betyder, at søgning fra lokalbibliotek til lokalbibliotek kan foregå inden for samme cylinder eller nogle få nabo-cylindre, hvilket mindsker den afstand, læse-/skrivehovedet skal bevæge sig.

Man skal derfor skabe hele sit bibliotekstræ - just efter at harddisken er formateret - inden, der lægges data eller programmer ind. Derved lægges alle biblioteker automatisk i harddiskens kant, og lokalbiblioteker lægges først i bibliotekets navneliste. Når bibliotekstræt skabes, skal man derfor først oprette alle lokalbiblioteker i én streng og dernæst oprette hele den næste path.

Det er ikke den måde, man normalt tænker på. Ofte vil man først oprette sine hovedbiblioteker som fx c:\tekst, c:\regneark, c:\database, c:\regnskab, c:\pascal, c:\utility osv. for dernæst at oprette de lokalbiblioteker, der er nødvendige i hvert bibliotek. Vil man maksimere sin harddisk-hastighed, er det ikke vejen frem.

Fil-fragmentering - undgå at sprede filer for meget

Fragmentering og defragmentering er nok den del af harddisk-optimeringen, de fleste har stiftet bekendtskab med gennem programmer som Nortons SpeedDisk og lignende.

Når data skrives på en nyformateret harddisk, lægges de i klynger, der ligger lige efter hinanden (eller i rækkefølge svarende til interleave-faktoren). Det næste sæt data lægges i klynger lige bag det første sæt.

Når nu det første sæt data vokser, lægges de overskydende data bag de nu skrevne data, dvs. at det første sæt data ikke længere ligger i samlet rækkefølge. Efterhånden som harddisken bruges, og data skrives og slettes, filer vokser og andre forsvinder, bliver det fænomen, at filer ikke ligger pænt i rækkefølge, mere og mere udbredt. Man taler om, at filerne er brutt op i stumper eller med et nudansk ord: fragmenteret.

Det siger sig selv, at jo flere cylindre en fil er spredt over, jo større læse- og/eller skrivetid er nødvendigt. Især for filer, der indlæses på én gang, er tidsforøgelsen åbenbar. Det

gælder fx regneark, tekst-filer og programmer. Hvor stumper af filer hentes af og til som fx databaser, føles det derimod ikke så meget.

Fragmentering er dog den mest betydnende faktor i forringet harddisk-hastighed. Defragmentering bør foretages regelmæssigt, hvis man vil holde sin harddisk "afkokset".

Ud over at holde harddisk-hastigheden oppe, opnås en større sandsynlighed for at kunne redde filer, der ved en fejtagelse er blevet slettet.

Diverse *unerase*-programmer virker perfekt på absolut ufragmenterede filer, så alene af den grund bør man holde sin harddisk oprydlet.

Der er principielt to forskellige måder at defragmentere sin harddisk på: Den første er at lave backup til disketter eller en anden harddisk ved simpelt hen at anvende copy *. * eller backup.

Dernæst kan man evt. reformattere harddisken, hvilket dog ikke er strengt nødvendigt og så genindlæse filerne med copy-eller restore afhængigt af, hvordan backup'en blev lavet. Det er en temmelig besværlig måde, men den kræver ikke andet end DOS og disketter.

Det kan dog gøres væsentlig lettere, hvis man råder over to PC'er med tilstrækkelig harddisk-kapacitet og overfører data fra den ene til den anden med FastLynx eller LapLink e.l. hjælpeprogrammer.

Den hurtige, enkle og "rigtige" måde er at anvende et af de mange hjælpeprogrammer, der har defragmenteringsprogrammer. Et eksempel er Norton Utilities Speed Disk-funktion.

Hvornår dette skal gøres, kan DOS-kommandoen chkdsk give en ide om. Ved at checke en af sine mest brugte store filer, gives bl.a. antallet af blokke denne er brutt i; fx med kommandoen chkdsk c:\veks\volstoi.txt.

Man kan ikke gøre ret meget for at hindre eller mindske fil-fragmentering. Arbejder man med DOS 2.x eller tidligere, er sandsynligheden for fragmentering væsentlig større end ved DOS 3.0 og senere. Årsagen er, at de tidlige versioner af DOS sylder disken op udefra og derfor skriver i de første klynger, der er ledige, også selv om det er en løstliggende klynge fra en enkelt lille fil, der er slettet, som

kun giver én ledig klynge. DOS 3.0 og senere arbejder sig ind mod diskens midte, og først når disken er fyldt med sammenhængende filer, udnytter DOS 3.3 de mellemliggende klynger.

Man kan med andre ord mindske (eller måske undgå) fragmentering ved at anvende DOS 3.0 eller senere, og sikre sig rimelig reserveplads på sin harddisk.

Arbejder man med en 40 MByte harddisk og har 1 MByte ledig kapacitet, kan man være sikker på en voldsom fragmentering. Har man derimod 20 MByte ledig, er man rimelig sikret mod fragmentering.

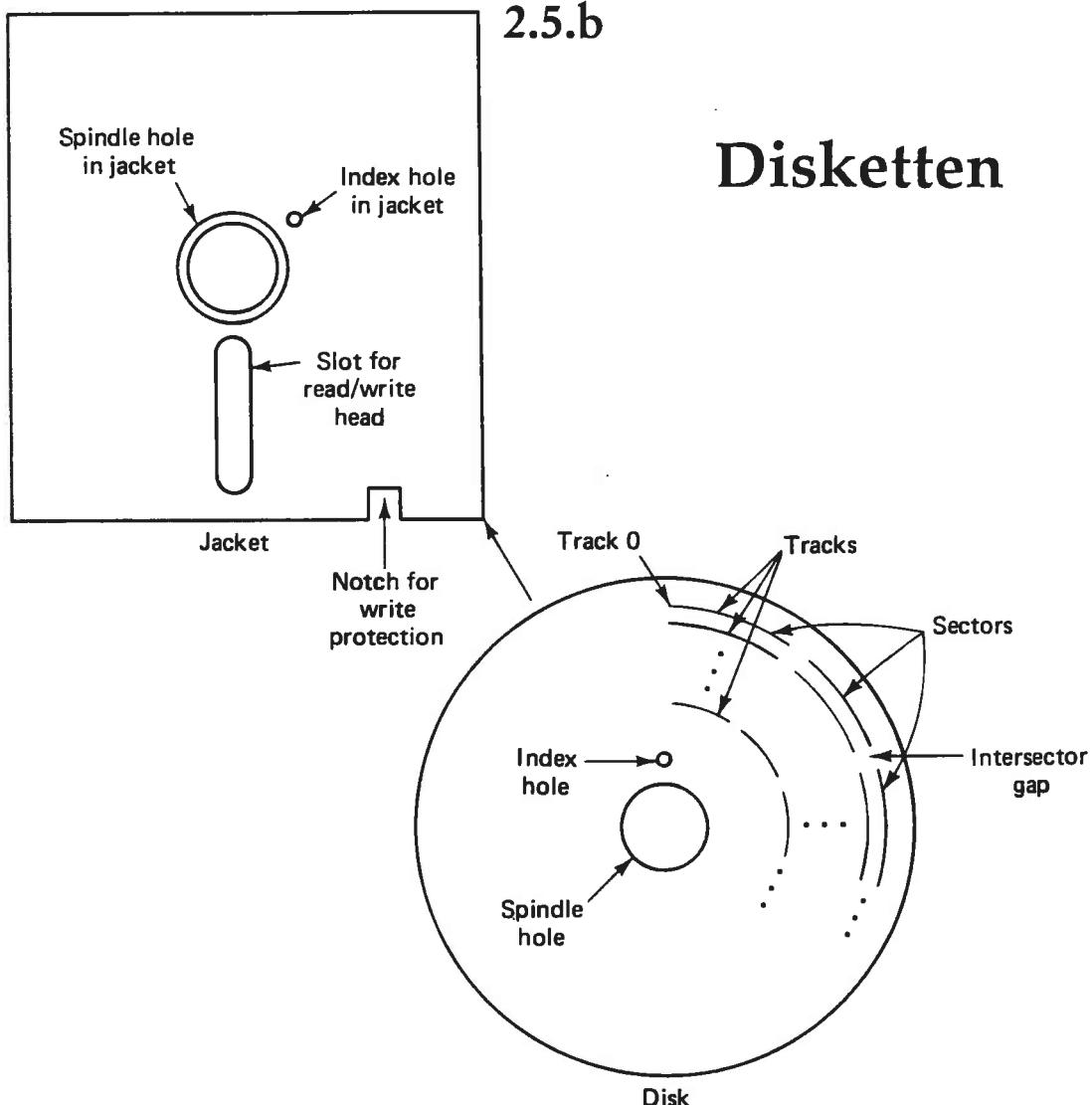
Hvis man sletter gamle filer for at skaffe sig denne rimelige plads på harddisken, må man dog defragmentere harddisken, før man skriver på harddisken igen ellers vil de tilfældige huller fra de slettede filer blot blive fyldt med kraftig fragmentering til følge.

Afsluttende bemærkninger

Mange af ovenstående anvisninger kan ikke følges 100 pct. De er mest ment som en art rettesnor. Ved at henlede opmærksomheden på en række fænomener, man i almindelighed ikke går og tænker på, kan man komme langt uden de store tiltag. Og egentlig er det vel sjovere at få sin "gamle" PC til at virke fornuftig og kvik, end at ofre stakkevis af kroner på en ny.

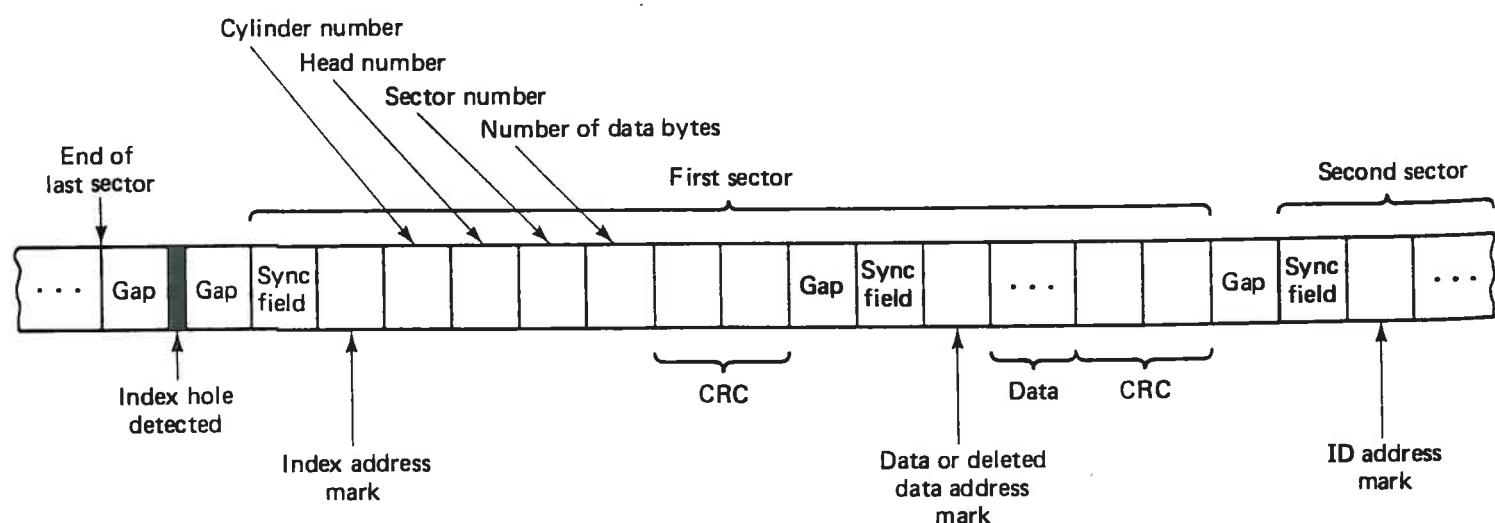
□

2.5.b



Construction of a diskette.

Track and sector format for a soft-sectored diskette.



2.5.c

Harddisk og diskette under dos DOS-filsystem

Bjørk Busch

Denne lille note har til hensigt, at udbygge forklaringen til hvorledes disketter og harddiske anvendes under DOS.

Princip for sektornummereringen under BIOS og DOS, eksemplificeret på en 5,25" 360KB (DSDD) diskette.

Bios nummerering af sektorer:

Afhængig af fysisk medie.

Opdelt i spor, side, sektor (3. dim. tabel)

Spor	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	.
Side	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	.
Sektor	1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9	1
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
	19	.																	

Dos sektor-nummerering:

Uafhængig af fysisk medie.

Fortløbende med start i 0 (1. dim. tabel)

Princip for opdeling af diskette i områder, eksemplificeret ved en 5,25" 360KB (DSDD) diskette.

Brug af dos sektorer (fysisk pladsering):

0	1-2	3-4	5-11	12-	...	(dec.)
---	-----	-----	------	-----	-----	--------

BOOT	FAT1	FAT2	ROOT	DATA	(filer/underkat.)
------	------	------	------	------	-------------------

Den samme opdeling findes for andre disketter og for et logisk drev på en harddisk (eks C:), idet størrelsen af de 2 FAT-tabeller og ROOT kan variere, og dermed give andre sektor-numre.

BOOT-sektoren indeholder et opstarts-program, samt informationer om hvorledes den er formaterer og hermed opbygget.

På en data-diskette vil BOOT-programmet normalt blot udskrive en meddelelse om, at disketten/det logiske drev ikke indeholder systemfilerne (DOS-kernen). På en system-diskette/-drev vil programmet opstarte den første del af DOS systemet.

Strukturen opbygges med programmet FORMAT.

Opdeling af harddisk under DOS:

Partitions tabel	Primær DOS partition	Extended DOS partition
------------------	----------------------	------------------------

Partitionstabellen er placeret på harddiskens første spor, første side, første sektor. Her ligger også et lille opstarts program, men det er forskelligt fra DOS-BOOT programmet.

Partitionstabellen beskriver opdelingen af harddisken i områder, også kaldet partitioner. Det er herved også muligt at placere partitionerne i en anden rækkefølge fysisk på harddisken.

Den primære DOS partition bliver til drev C: og er opbygget på samme måde som disketten med DOS-BOOT sektor, FAT tabeller m.m.

De fysiske (BIOS) sektor adresser er selfølgeligt anderledes.

Der behøver ikke være en Extended DOS-partition, idet den primære kan optage hele pladsen, eller der kan være ledig plads til evt. andre systemer.

Opdeling af Extended DOS-partition på harddisk:

Ext. Part. tabel	Logisk DOS Drev	Logisk DOS Drev
------------------	-----------------	-----------------

Den Extendede partition indeholder en ny partitionstabel, der beskriver hvorledes dette område er opdelt yderlig.

Opdelingen sker her i logiske drev, og hvert af disse er igen opbygget på samme måde som disketten, med DOS-BOOT sektor, FAT tabeller m.m.

Opdelingen af harddisken i partitioner kan ske med programmet FDISK.

Logisk sammenhæng mellem FAT, ROOT og DATA arealer:

KATALOG (ROOT)		FAT	
Filnavn	Start		
FIL1	002	000	2 første elm. i FAT bruges ikke normalt.
FIL2	006	001	1. byte = mediedescriptor
FIL3	005		DATA CLUSTRE
UKAT	00A	002	DOS Sekt. (dec.)
		003	12, 13
		004	14, 15
		FFF	16, 17
		007	18, 19
		FFF	20, 21
		008	22, 23
		FFF	24, 25
		000	26, 27
		00B	28, 29
		FFF	30, 31
		000	32, 33
		FF7	34, 35
		000	36, 37

Her 2 sektor pr. cluster

UKAT er et underkatalog. Det administreres som en alm. fil med hensyn til pladsering.

Underkataloger er opbygget på samme måde som hovedkataloget (ROOT), idet der dog er følgende forskelle:

- Der optræder ingen LABEL i underkataloger.
 - Der optræder altid 2 filnavne i starten af underkataloger:
 - . Der refererer til underkataloget selv.
 - .. Der refererer til kataloget lige over.
- Ved reference til ROOT-kataloget, der jo ikke ligger i nogen af data-clustrene, står der 0 i startadressen.

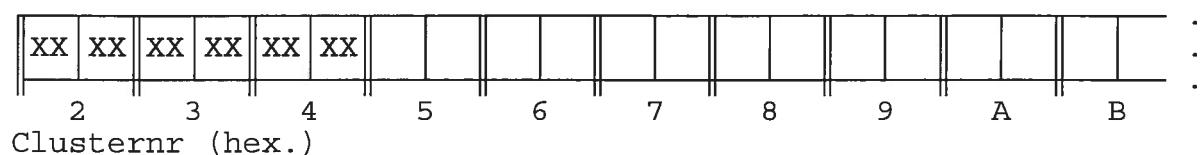
Der kan opstå fejl i FAT-tabeller og kataloger i spec. situationer. Der kan herved f.eks opstå "kæder" i FAT, uden tilhørende fil. Dos kommandoen CHKDSK kan blandt andet anvendes til kontrol af sådanne ting.

FIL1's fysiske pladsering (Sammenhængende) :

Start -> 002 -> 003 -> 004 (stop)

Dos sektor (dec.)

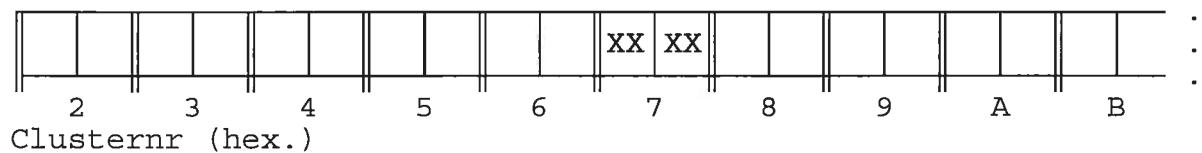
12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 .

**FIL2's fysiske pladsering (Sammenhængende) :**

Start -> 007 (stop)

Dos sektor (dec.)

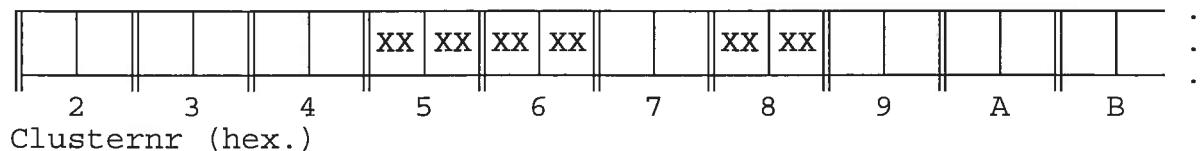
12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 .

**FIL3's fysiske pladsering (Spredt) :**

Start -> 005 -> 006 -> 008 (stop)

Dos sektor (dec.)

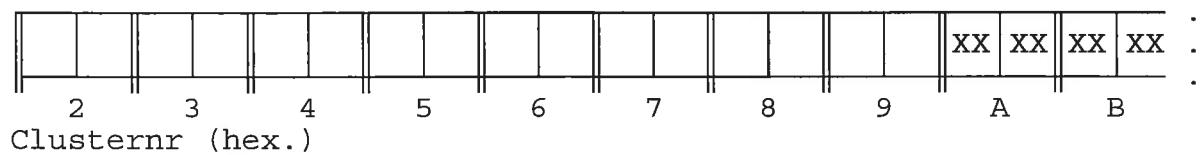
12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 .

**UKAT's fysiske pladsering (Sammenhængende) :**

Start -> 00A -> 00B (stop)

Dos sektor (dec.)

12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 .



Opbygning af FAT-tabel.

Start af FAT-TABEL, som den ser ude i et dump:

```
0000: FD FF FF    03 40 00    FF 7F 00    FF 0F 08    FF 0F 00
```

Mediekode: FD svarende til 360KB, 40 spor, 2 side og 9 sektorer. Koden efterfølges af FF FF, som ikke anvendes til noget.

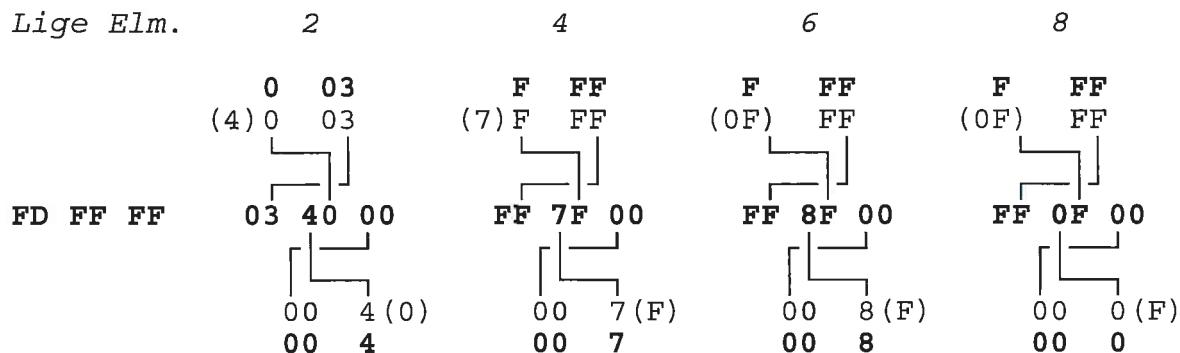
I DOS versioner før 3.30 blev data fra BOOT sektoren ikke anvendt for disketter, men formatet blev allene defineret ud fra 1. byte i FAT-tabellen.

Omsætning til 12 bits FAT tabel:

1. Det ønskede clusternr gøges med 1,5
2. Der rundes ned, og man har så offset relativt til tabel-start.
3. Der hentes 2 bytes fra denne adresse (den beregnede og den efterfølgende).
De 2 bytes ombyttes som normalt for word's.
4. For lige clustre anvendes de sidste 12 af de 16 bits, og for ulige clustre anvendes de første 12 af de 16 bits.

Hvis man fortolker FAT-tabellen fra start af, kan man tage 3 bytes af gangen, der så svarer til et lige og et ulige cluster.

- a) De første 2 bytes af de 3 ombyttes og venstre halve bytes smides bort (Left=Lige) hvorved man får det lige cluster.
- b) De sidste 2 bytes af de 3 ombyttes og højre halve bytes smides bort hvorved man får det ulige cluster.



Ulige Elm. 3 5 7 9

Ved 16 bits FAT tabeller er der 2 byte pr. cluster, her springes de første 4 bytes over svarende til cluster 0 og 1. FAT elementerne er alm. word så der skal stadig fortages alm. ombytning.

Opbygning af katalog-tabeller.

Hvert element i katalog-tabellen er på 32 byte, svarende til 2 linier i dumpet. Disse 32 bytes indeholder følgende:

Indhold	position	antal bytes
Filnavn	00 - 07	8
Filextention	08 - 0A	3
Attribute	0B - 0B	1
Reserveret	0C - 15	10
Tidspunkt	16 - 17	2
Dato	18 - 19	2
Clusternr	1A - 1B	2
Filstørrelse	1C - 1F	4

Attribute byten's 8 bit har følgende betydning:

Betydning / bit	7	6	5	4	3	2	1	0
Read only	1
Hidden file	1	.
System	1	.	.
Volume label	1	.	.	.
Subdirectory	.	.	.	1
Archive	.	.	1
Ubrugt	.	1
Ubrugt	1

Klokkeslettet er opdelt på følgende måde:

Bit 0 - 4	Sekunder/2 (0-29). Gang med 2 for sek.
Bit 5 - 10	Minutter (0-59)
Bit 11 - 15	Timer (0-23)

Bytes ligger som sædvanlig omvendt i lager og skal ombyttes.

Dato er opdelt på følgende måde:

Bit 0 - 4	Dag (1-31)
Bit 5 - 8	Måned (1-12)
Bit 9 - 15	Årstal - 1980. Adder 1980 til for år.

Bytes ligger som sædvanlig omvendt i lager og skal ombyttes.

Når en fil slettes ændres 1. bogstav i filnavn til **E5** og plads frigives i FAT-tabel. Pladsen i kataloget kan herefter genbruges. Nye filer indlægges på første ledige plads i kataloget.

Eksempel på FIL-element i ROOT og fortolkning af dette:

1. element i ROOT katalog, som det ser ud i dump.

```
0000: 4C 41 42 50 43 44 20 20 - 20 20 20 28 00 00 00 00  
0020: 00 00 00 00 00 00 61 80 - 12 17 00 00 00 00 00 00
```

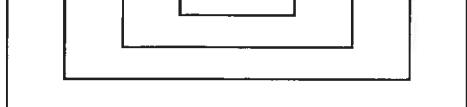
Fortolkning:

Filnavn:	4C 41 42 50 43 44 20 20 => LABPCD	
Extention:	20 20 20 => (blank)	
Attribute:	28 => ARC + LABEL	
Reserveret:	00 00 00 00 00 00 00 00 00 00 00	
Tidspunkt:	61 80 => 80 61 = 1000 0000 0110 0001 TTTT TMMM MMMS SSSS	
Time:	10000 => 16	
Min:	000011 => 3	
Sek:	00001 => 1*2 => 2	
Dato:	12 17 => 17 12 = 0001 0111 0001 0010 ÅÅÅÅ ÅÅÅM MMMD DDDD	
År:	0001011 => 11+1980 => 1991	
Måned:	1000 => 8	
Dag:	10010 => 18	
Startcluster:	00 00	
Filstørrelse:	00 00 00 00	En label optager kun plads i ROOT

2. element i ROOT katalog, som det ser ud i dump.

```
0030: 46 73 4C 32 20 20 20 20 - 54 58 54 21 00 00 00 00
0040: 00 00 00 00 00 00 20 17 - 5D 15 02 00 05 0A 00 00
```

Fortolkning:

Filnavn:	46 73 4C 32 20 20 20 20 => FIL1
Extention:	54 58 54 => TXT
Attribute:	21 => ARC + READONLY
Reserveret:	00 00 00 00 00 00 00 00 00 00 00
Tidspunkt:	20 10 => 10 20 = 0001 0000 0010 0111 TTTT TMMM MMMS SSSS
Time:	00010 => 2
Min:	000001 => 1
Sek:	00111 => 7*2 => 14
Dato:	5D 15 => 15 5D = 0001 0101 0101 1101 ÅÅÅÅ ÅÅÅM MMMD DDDD
År:	0001011 => 10+1980 => 1990
Måned:	1010 => 10
Dag:	11101 => 29
Startcluster:	02 00 => 00 02
Filstørrelse:	05 0A 00 00 => 00 00 0A 05 => 2565  Sidste cluster er ikke fyldt op.

De enkelte bytes i binære tal er lagt "baglæns" ud i lager.

Beskrivelse af boot-sector:

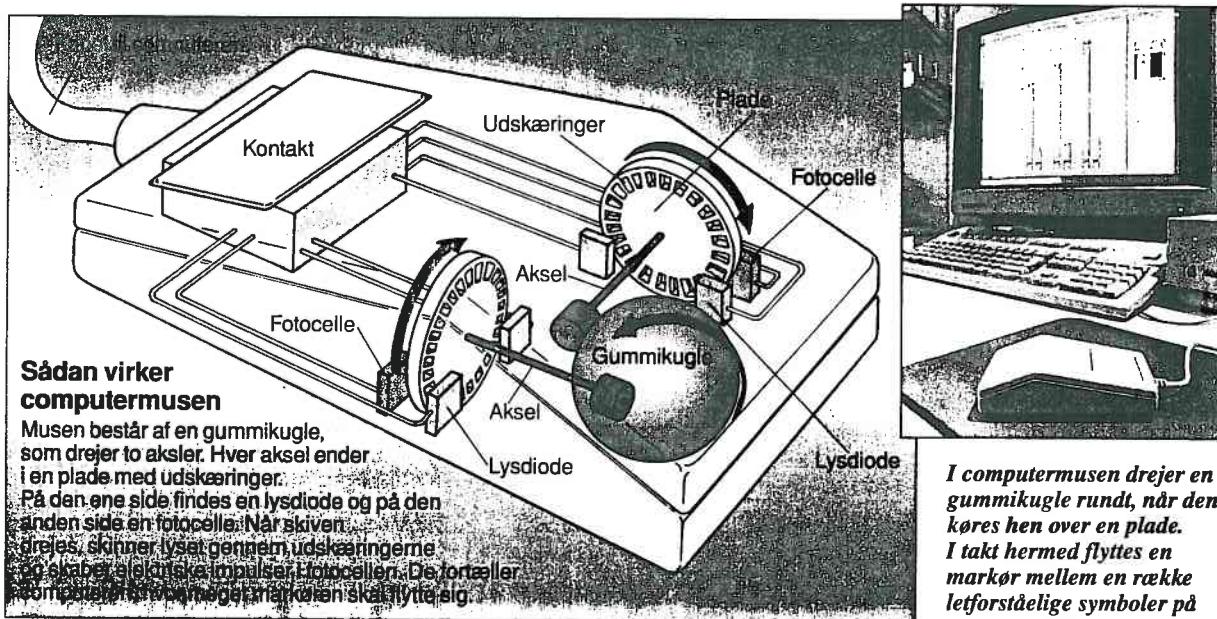
00 - 02	...	-	Jump under om beskrivelse af medie, til BOOT-PROGRAMMET.
03 - 0A	CHAR	-	Producentnavn i ASCII-format.
0B - 0C	WORD	-	Antal bytes pr. sektor.
0D - 0D	BYTE	-	Antal sektorer pr. cluster.
0E - 0F	WORD	-	Antal reserverede sektorer.
10 - 10	BYTE	-	Antal FAT tabeller
11 - 12	WORD	-	Antal entries i ROOT.
13 - 14	WORD	-	Antal sektorer i alt på medie.
15 - 15	BYTE	-	Medie descriptor - står også i første byte i FAT-tabel. Før DOS 3.30 anvendtes alene descriptor fra FAT-tabel når DOS skulle anvende en diskette og format var derfor fastlåst.
16 - 17	WORD	-	Antal sektorer pr. FAT-tabel.
18 - 19	WORD	-	Antal sektorer pr. spor.
1A - 1B	WORD	-	Antal sider.
1C - 1D	WORD	-	Antal skjulte sektorer (til DOS 3.xx)
1E - 1FF	...		Resten af BOOT-programmet.

Fra DOS 4.00 er der informationerne i BOOT-sektoren udvidet. Dette skyldtes blandt andet den begrænsning der ligger i at feltet med antal sektorer på mediet kun er et word, hvilket med en sektorstørrelse kun giver mulighed for 32 MB.

Det nye format bygger videre på det gamle og rummer også en beskrivelse (i klar tekst) af om FAT-tabellen er en 12-bits eller en 16-bits. BOOT-sektoren rummer også volume lablen, idet denne stadig også opbevares i ROOT-directory.

2.6.a

PC-MUS



Genvej til computerens hjerne

Teknik Det er langsomme-ligt og besværligt at styre et indviklet computerprogram ved hjælp af et tastatur. Opfindelsen af computermusen har gjort det meget nemmere at

bruge en PC. I musen findes en gummikugle, som styrer to aksler. Den ene styrer de vandrette bevægelser, den anden de lodrette. Når musen kører hen over en plade, sendes elektri-

ske impulser til computeren, og en lille pil flytter sig rundt på skærmen.

Pilen kan pege på forskellige symboler, som aktiveres ved hjælp af en tryknap på

musen. Derefter udføres den bestemte del af computerpro-grammet.

Betegnelsen mus blev pa-tenteret allerede i 1964 som det engelske »mouse«. □