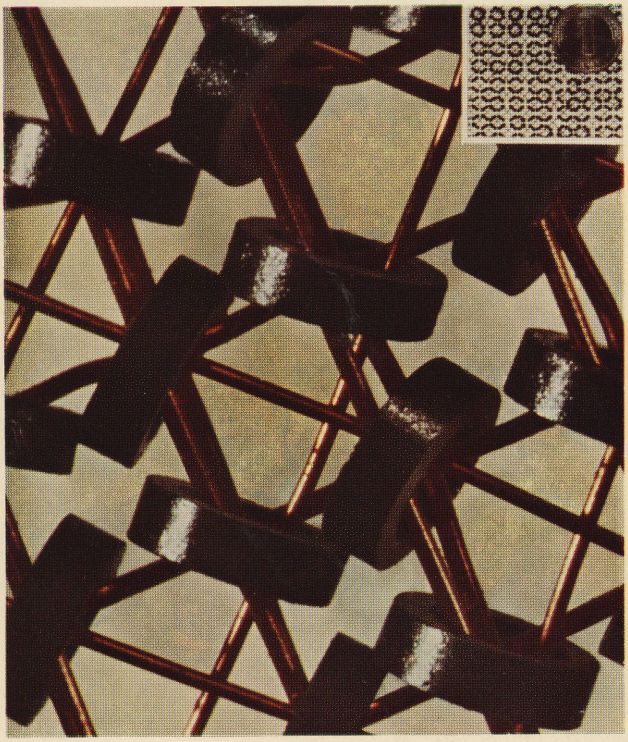
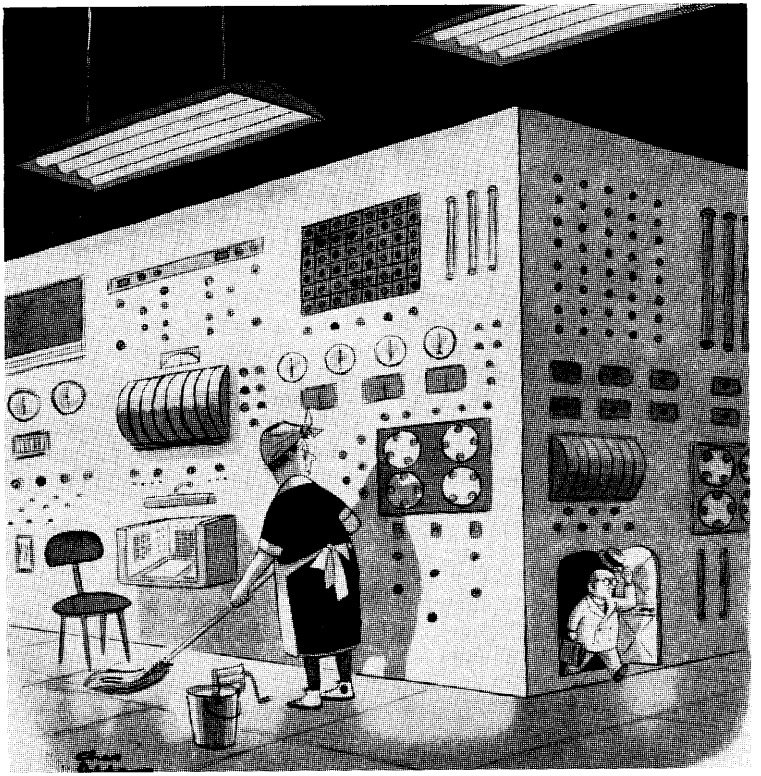
HVORDAN ELEKTRONREGNEMASKINEN VIRKER . . .



PER BRINCH HANSEN · REGNECENTRALEN

Særtryk, VOR VIDEN, april 1965.



TEKNIKKEN BAG ELEKTRONREGNEMASKINEN

Tænkende maskiner?

Af civilingeniør Per Brinch Hansen

»Kan du lægge tal sammen?« spurgte den hvide dronning.
»Hvad er en og en og en og en og en og en og en og en og en?«
»Det ved jeg ikke,« sagde Alice. »Jeg kunne ikke følge med.«
– Lewis Carroll.

Det er knap tyve år siden, amerikaneren *John von Neumann* udviklede den første *programstyrede elektronregnemaskine.* Og i dag er disse maskiner allerede i færd med at overtage meget af det routinearbejde, som man før mente kun kunne udføres af mennesker. Disse forbløffende maskiner beregner ugelønninger for store virksomheder, ajourfører checkkonti i banker, beregner præmieopkrævninger for forsikringsselskaber eller lægger en fuldstændig tidsplan for bygningen af et ejendomskompleks.

Hvordan elektronregnemaskinerne egentlig bærer sig ad med at løse disse opgaver, er et mysterium for de

fleste mennesker. I dagspressen omtales regnemaskiner dramatisk som »elektronhjerner« eller »tænkende maskiner«. Man får uvilkårligt det indtryk, at de besidder en næsten menneskelig intelligens.

I virkeligheden er disse maskiner hverken særligt indviklede eller særligt begavede. Enhver beregning, maskinen skal udføre, må i forvejen planlægges indtil mindste detaille af brugeren. Når maskinen herefter startes, vil den slavisk udføre de operationer, den i øjeblikket er indstillet til — og hverken mere eller mindre end det. Regnemaskinens *begrænsning* er, at den kun kan udføre meget trivielle beregninger. Dens *styrke*, at den kan gøre dette langt hurtigere og mere pålideligt end noget menneske.

Maskinens store regnehastighed har gjort det muligt at løse opgaver, der før blev opgivet som håbløse. I 1922 fik en englænder, ved navn *Richardson*, den idé at forudsige vejret fra dag til dag ud fra matematiske beregninger. Idéen var god nok, men desværre viste beregningerne sig at være så omfattende, at han måtte regne med papir og blyant i tre måneder på den første forudsigelse. På det tidspunkt havde den tabt det meste af sin værdi — og den var desuden forkert! I dag løses den samme opgave af en elektronregnemaskine på nogle få timer. Regnemaskinen, en CDC 1604, ejes af den amerikanske flåde. Flere tusinde vejrstationer, fordelt over hele den nordlige halvkugle, sender daglig meteorologiske observationer via fjernskriver til denne maskine. Efter at have udført det astronomiske antal af en halv milliard beregninger trykker maskinen selv det færdige vejrkort for den følgende dag.

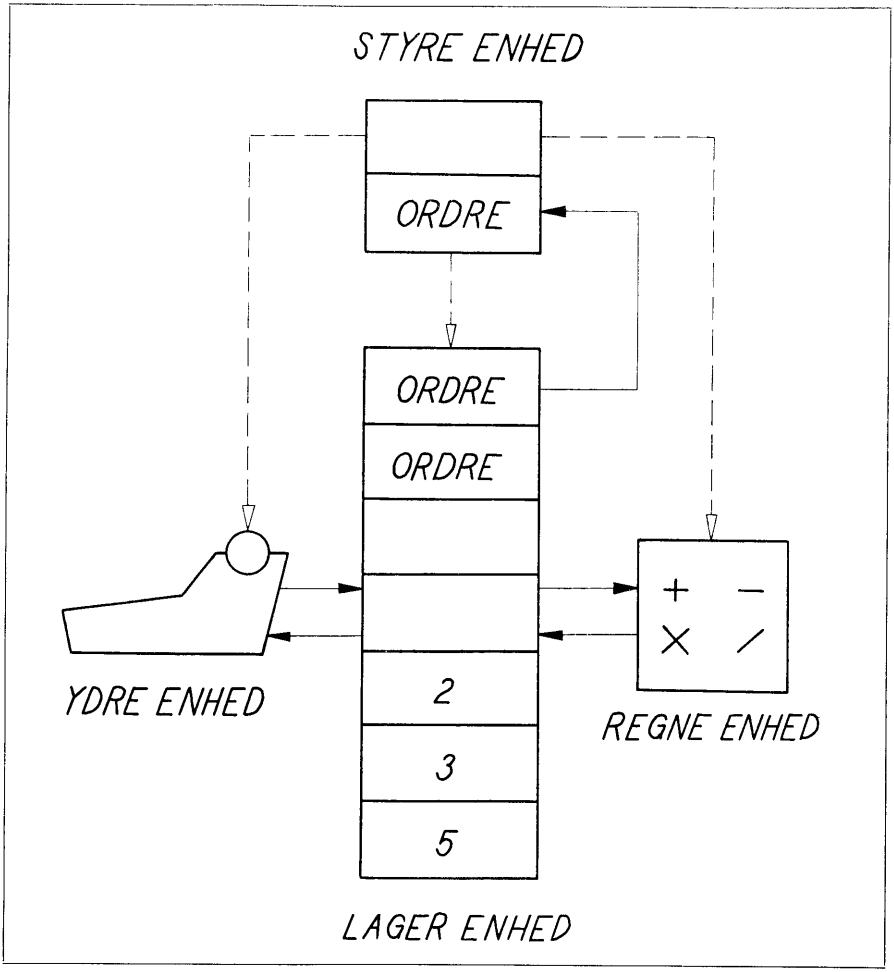
Den enorme elektroniske hastighed er noget, de fleste af os accepterer. Mere forbløffende virker det, at en regnemaskine kan arbejde *uden* menneskelig indgriben. Den styrer faktisk sig selv, mens den regner! Lad os kigge nærmere på en meget simpel regnemaskine for at få en fornemmelse af, hvordan maskinen egentlig arbejder, og hvordan den er bygget.

Hvordan skal en maskine da være opbygget, for at den kan regne? Ja, der skal for det første være en måde at få tallene ind og ud af maskinen. Den regnemaskine, De ser på næste side, kommer i kontakt med omverdenen ved hjælp af en elektrisk *skrivemaskine*. Her kan operatøren taste de tal ind, som maskinen skal arbejde på. Og når en beregning er slut, trykker skrivemaskinen selv det færdige resultat ud.

Regnemaskinen må også have et sted, hvor den kan gemme de tal, den skal regne på. Denne enhed kaldes for *lageret*. Lageret er delt i en række 'skuffer' eller *celler*. Og i hver celle kan der gemmes ét tal.

Endelig er der selve *regneenheden*. Her foregår de egentlige beregninger. Regneenheden kan lægge to tal sammen eller trække dem fra hinanden. Den kan også multiplicere og dividere.

Vi vil i detailler se, hvordan maskinen løser den simple opgave at lægge to tal sammen, for eksempel 2 + 3. Operatøren taster først de to tal ind på skrivemaskinen. Skrivemaskinen virker som en almindelig fjernskriver. Den sender en serie elektriske impulser ind i regnemaskinen, og disse bevirker, at to-tallet og tre-tallet blir anbragt i hver sin lagercelle. Fra lageret sender maskinen selv tallene videre til regneenheden, hvor de lægges sammen ad elektrisk vej. Resultatet, der blir et fem-tal, sendes fra regneenheden tilbage til lageret, hvor det gemmes i en tredie celle. Til



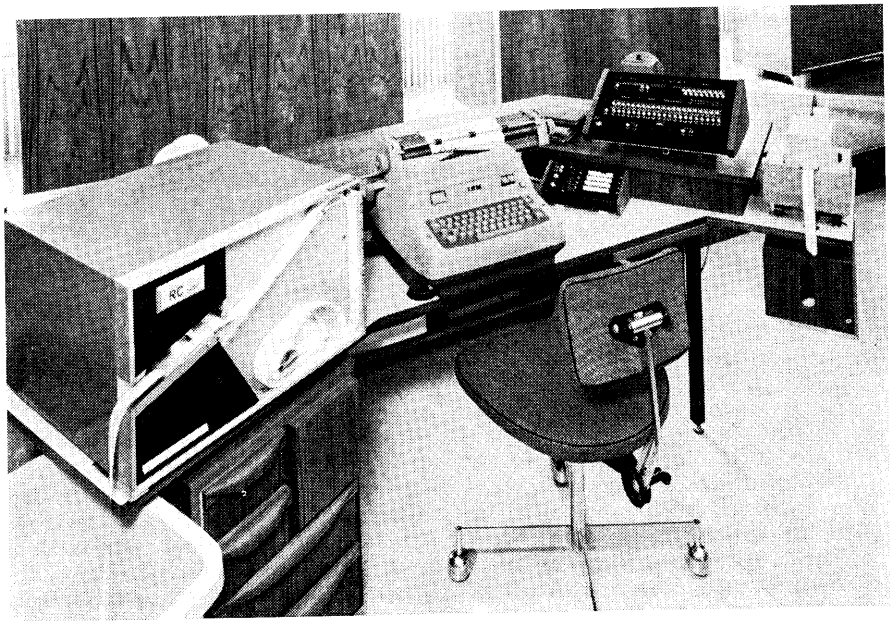
Regnemaskinens opbygning: tallene kommer ind og ud af maskinen gennem en elektrisk skrivemaskine til venstre. I midten er lagerenheden, hvori ordrer og talresultater gemmes. De egentlige beregninger sker i regneenheden til højre. Styreenheden foroven modtager ordrerne fra lageret og omsætter dem til elektriske signaler, der styrer de tre andre enheder.

sidst sendes resultatet fra lageret ud til skrivemaskinen som en serie elektriske impulser, der får denne til at trykke et fem-tal. Dette er i korte træk, hvad der foregår set på afstand.

Regnemaskinen består altså af en skrivemaskine, et lager og en regneenhed. Men endnu adskiller den sig ikke stort fra en almindelig kontorregnemaskine. Man kan taste tal ind i en kontorregnemaskine; den kan lægge dem sammen, og den gemmer

undervejs det foreløbige resultat. Men så længe man arbejder med en kontorregnemaskine, er det hele tiden *operatøren*, der trin for trin bestemmer rækkefølgen af de enkelte regneoperationer.Og her ligger den afgørende forskel. Når man først har startet en elektronregnemaskine, arbejder den videre på egen hånd — uden operatørens indgreb! Det er simpelt hen en nødvendighed for at kunne udnytte maskinens enorme hastighed. Tænk for eksempel på en elektronregnemaskine, der kan udføre 100 000 regneoperationer på ét sekund. Hvis operatøren her skulle styre rækkefølgen af de enkelte operationer ved at trykke på forskellige knapper, ville maskinen hele tiden stå og vente på operatørens næste ordre. På den måde ville den elektroniske hastighed blive illusorisk.

Så det springende punkt er: hvordan får man maskinen til at arbejde på egen hånd? Det sørger den fjerde enhed, *styreenheden*, for. Styreenheden kan, ligesom regneenheden, modtage et tal fra en lagercelle. Men styreenheden opfatter ikke et tal som noget, der skal regnes på. Den opfatter derimod et tal som en *kode*, der fortæller, hvilken operation maskinen nu skal udføre. For eksempel vil styreenheden opfatte et et-tal som en ordre, der siger, at maskinen nu skal modtage et tal fra skrivemaskinen og anbringe det i en bestemt lagercelle. Eller et fem-tal kan angive, at et tal skal hentes fra en bestemt lagercelle og sendes over til



En mindre elektronregnemaskine: skabet i baggrunden rummer maskinens centrale dele, lageret, regneenheden og styreenheden. På bordet står en skrivemaskine for ind- og udlæsning af tal. Til højre ses en kontrolpult og til venstre et læseapparat for tal, der er hullet på papirstrimmel.

regneenheden, for der at blive lagt til et andet tal.

Disse tal, som maskinen opfatter som ordrer, har operatøren i forvejen placeret i lageret. Styreenheden læser disse ordrer én for én og omsætter dem til elektriske signaler, der styrer de tre andre enheder. At styreenheden kan opfatte et tal som en ordre og omsætte den til handling, skyldes ikke sort magi. Nøjagtigt det samme gør enhver automatisk telefoncentral. Det tal, De drejer på nummerskiven, er en ordre til centralen om at finde en bestemt abonnent og sende et signal til vedkommendes telefon.

Styreenheden er regnemaskinens central. Den kan modtage et tal fra lageret og opfatte det som en ordre om at vælge en bestemt enhed og sende et elektrisk styresignal til denne enhed — enten til skrivemaskinen, til en bestemt lagercelle eller til regneenheden.

Lageret indeholder altså to slags tal. Der er de normale tal, som regneenheden lægger sammen, trækker fra hinanden o. s. v., og så er der de tal, som styreenheden modtager og opfatter som ordrer. Den følgende tegning viser, hvad lageret faktisk indeholder, når maskinen skal lægge to tal sammen. På dette nærbillede af lageret er de enkelte lagerceller nummereret fra 0 til 15. I cellerne 0 til 6 står de ordrer, maskinen skal udføre, mens cellerne 13, 14 og 15 er reserveret til de tal, maskinen skal regne på.

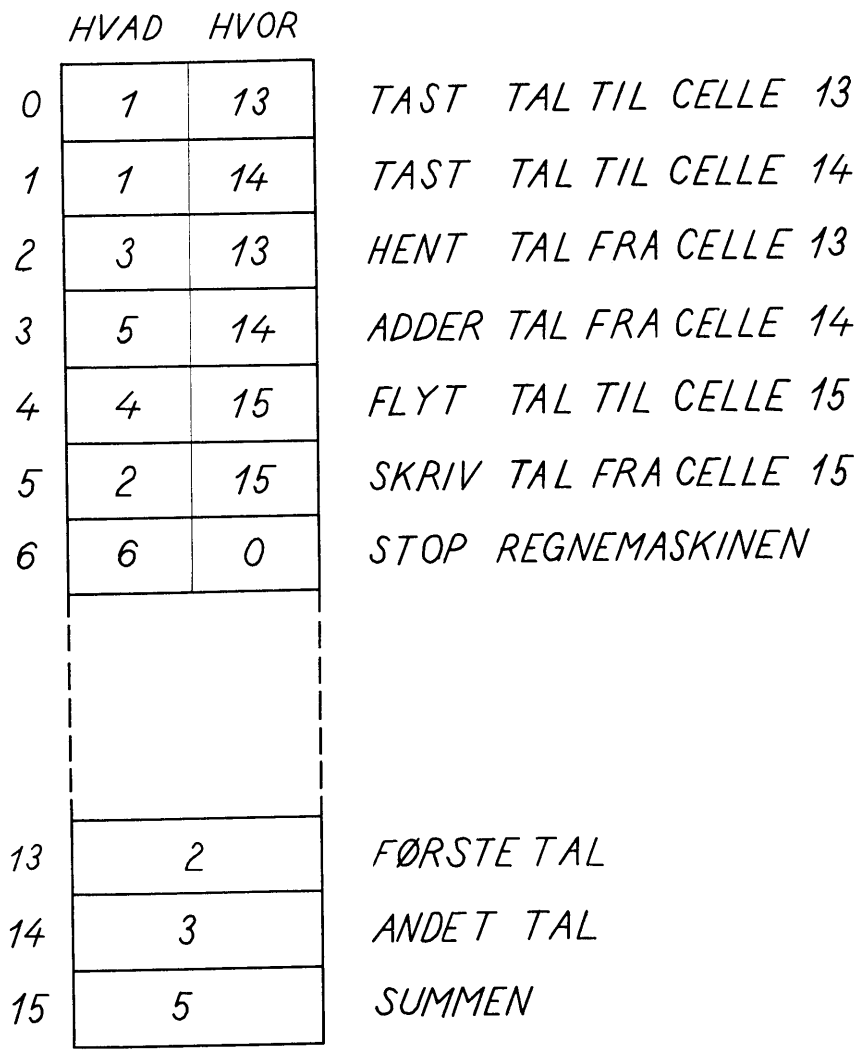
Foroven i lageret står altså en række ordrer, som styreenheden skal udføre. I hver lagercelle står der én ordre. Vi kan som eksempel se på den ordre, der står i celle 2:

3 13

Ordren består af to dele. Venstre halvdel er en talkode, der fortæller styreenheden, *hvad* maskinen skal gøre. Tre-tallet her betyder for eksempel, at styreenheden skal finde et bestemt tal i lageret og derefter sende dette tal over i regneenheden. Højre halvdel af ordren fortæller styreenheden, *hvor* i lageret det ønskede tal findes. Denne del af ordren er nemlig simpelt hen nummeret på en bestemt lagercelle. Den fuldstændige ordre angiver altså i dette tilfælde, at styreenheden skal *hente* det tal, der er gemt i *celle 13*, og anbringe det i regneenheden.

Det er operatøren, der har forberedt denne samling ordrer, og han har ved hjælp af skrivemaskinen fået dem anbragt i lageret. Vi skal nu se, hvordan de enkelte ordrer udføres. På det tidspunkt, da den egentlige beregning startes, har vi endnu ikke anbragt noget tal i de tre nederste celler. Operatøren trykker først på en *startknap*. Det bevirker automatisk, at den ordre, der står i celle 0, sendes op til styreenheden. Denne første ordre angiver, at maskinen nu skal være klar til at modtage et tal fra skrivemaskinen og gemme dette i celle 13. Styreenheden sender derfor en startimpuls til skrivemaskinen, og operatøren kan nu taste det første tal ind, for eksempel et to-tal. Skrivemaskinen sender to-tallet ind i maskinen som en serie elektriske impulser. Samtidig sender styreenheden et signal til lageret, så to-tallet havner i celle 13.

Styreenheden er nu færdig med den første ordre. Den henter så en ny ordre fra den næste celle, d. v. s. fra celle 1. Det er en ordre om at modtage endnu et tal fra skrivemaskinen og gemme det i celle 14. Lad os sige, at operatøren her taster et tre-tal. Det ryger ind i celle 14. Nu har regnemaskinen fået de to tal, den



Nærbillede af lageret: de ordrer, som styreenheden skal udføre, står i cellerne 0 til 6. Cellerne 13 til 15 indeholder de tal, der skal regnes på. Meningen med de enkelte ordrer er angivet til højre for lageret.

skal lægge sammen, og den arbejder nu videre uden operatørens medvirken.

Den læser den følgende ordre fra celle 2. Denne ordre fortæller, at styreenheden skal hente det tal, der står i celle 13, og sende det over i regneenheden. Her sender styreenheden først signal til lageret, så to-tallet hentes ud af celle 13. Herfra sendes tallet over til regneenheden som en serie elektriske impulser. I regneenheden gemmes to-tallet midlertidigt i en lokal lagercelle.

Den næste ordre står i celle 3. Den siger, at maskinen skal addere det tal, der står i celle 14, til det tal, der lige er sendt til regneenheden. Altså sender styreenheden tre-tallet fra celle 14 over i regneenheden, hvor det ad elektrisk vej adderes til to-tallet. Resultatet blir et fem-tal, der foreløbig blir stående i regneenheden. Men allerede den næste ordre, der hentes fra celle 4, bevirker at resultatet flyttes fra regneenheden over i celle 15.

Herefter kommer en ordre, der siger, at indholdet af celle 15 skal skrives på skrivemaskinen. Styreenheden sender derfor fem-tallet fra celle 15 ud til skrivemaskinen som en serie elektriske impulser. Disse impulser får nu skrivemaskinen til at trykke det fem-tal, der er resultatet af vores beregning. Den sidste ordre er en *stop-ordre,* der fortæller styreenheden, at beregningen er slut.

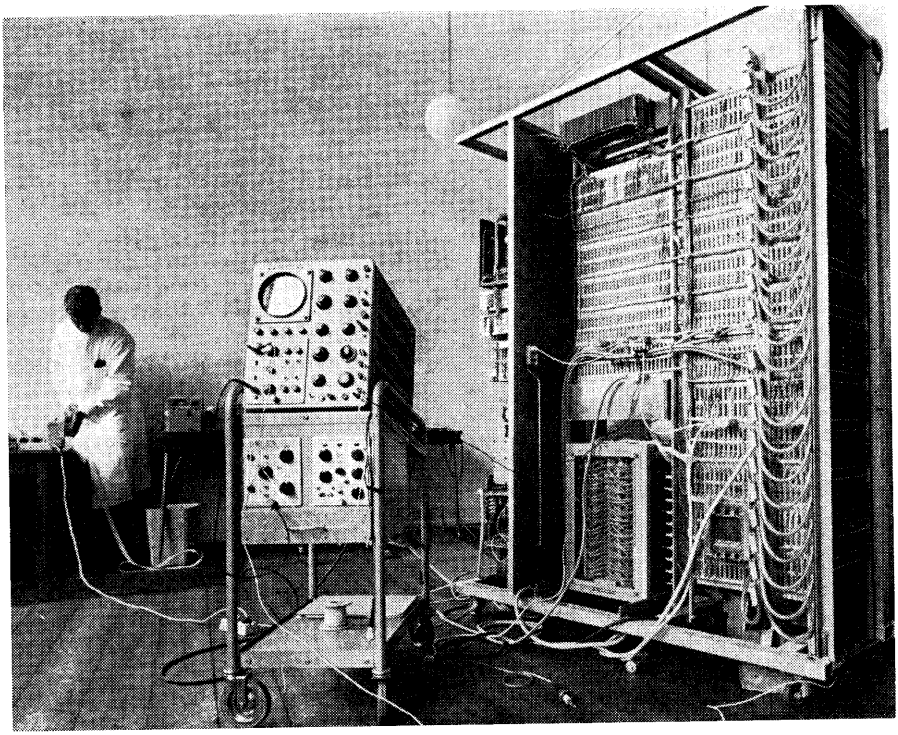
Det lille eksempel viser, at det er styreenheden, der gør elektronregnemaskinen til noget enestående. Styreenheden kan modtage en række tal fra lageret og opfatte dem som simple ordrer, der siger: *hent* et tal, *adder* et andet tal, *flyt* resultatet o. s. v.

Den samling ordrer, der står i lageret, når en beregning startes, kaldes et *program.* For at få maskinen til at løse en bestemt opgave må man først sætte sig ned og skrive et sådant program, d. v. s man må skrive en passende serie af de primitive ordrer, som maskinen er i stand til at udføre. Denne forberedelse kaldes *programmering.* Når det er sket, må man ved hjælp af skrivemaskinen anbringe den færdige serie ordrer i lageret. Herefter kan man starte selve beregningen ved at trykke på startknappen.

Programmet må indtil mindste detaille fortælle styreenheden, hvad den skal gøre. Styreenheden selv er nemlig kun i stand til slavisk at læse ordrerne én for én og udføre dem. Men den kan på ingen måde afgøre, om der er nogen fornuftig mening med programmet som helhed. Det har brugeren af maskinen alene ansvaret for. At forestille sig, at maskinen kan 'tænke' selv, er altså meningsløst. Det kan den ikke! Alle dens handlinger er fuldstændig planlagt på forhånd af det menneske, der har skrevet programmet.

John von Neumann's geniale ide, at lade regnemaskinen styre 'sig selv' ved hjælp af et *lagret program,* gør den til et af de mest universelle værktøjer, mennesket har opfundet. Denne egenskab indebærer nemlig, at maskinen ikke på forhånd er bygget og fast indstillet til at løse én bestemt opgave. Ved at anbringe forskellige programmer i maskinens lager kan man få den til at udføre så vidt forskellige opgaver som at udskrive telefonregninger, foretage pladsreservationer på et fly eller styre et helt stålvalseværk.

Efter dette indtryk af, hvordan elektronregnemaskinen *arbejder* skal vi i en følgende artikel se på, hvorledes dens enkelte enheder er *opbygget.*



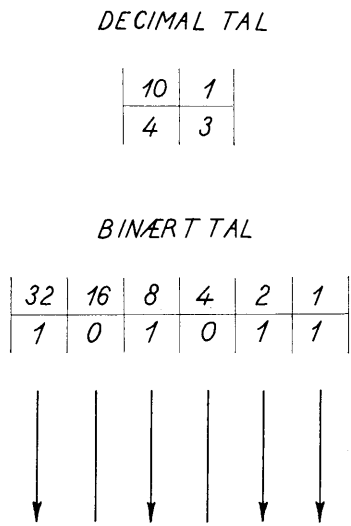
Tendensen til at tillægge elektronregnemaskiner næsten menneskelige egenskaber har givet sig de mest barokke udslag. En repræsentant for et stort regnemaskinefirma blev af en kunde bedt om at forklare forskellen mellem to elektronregnemaskiner. »Lad os sige det på den måde,« sagde repræsentanten, »740-maskinen *synes*, at 690'eren er en sinke!«

Efter at have set, *hvor* primitive de enkelte operationer i en regnemaskine faktisk er, vil vi med sindsro betragte *alle* regnemaskiner som sinker. Set på afstand virker det nok forbløffende at se maskinerne udføre de mest forskelligartede og indviklede beregninger uden operatørens manuelle indgriben. Men man må ikke glemme, at enhver kørsel på maskinen er resultatet af en nøje planlægning, hvorunder brugeren har splittet sin komplicerede beregning op i en række uhyre simple operationer, som maskinen kan udføre på egen hånd.

Efter denne planlægning blir det færdige program nedskrevet som en serie talkoder, der anbringes i maskinens lager. Når maskinen så startes, tager styreenheden disse talkoder én for én og omsætter dem til elektriske signaler, der styrer simple operationer i maskinens øvrige enheder: skrivemaskinen, lageret og regneenheden. Det er i korte træk den måde, elektronregnemaskiner arbejder på. Disse maskiner 'tænker' ikke — de 'adlyder' simpelt hen slavisk de ordrer, som brugeren har sammensat sit program af.

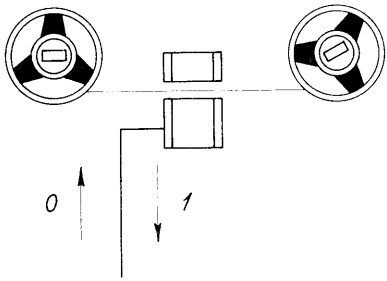
Vi skal nu se, hvordan en sådan programstyret regnemaskine i praksis kan opbygges. For at starte med *tallene* selv — hvordan ser de ud inden i maskinen? Ja, her er igen en ting, der viser, hvor primitiv regnemaskinen er. Den kender nemlig kun tallene *nul* og *ét*. Det betyder, at hvis vi vil taste et tal, lad os sige 43, ind på skrivemaskinen, så må vi først omskrive tallet til en serie et-taller og nuller, som maskinen vil acceptere.

Ser vi først på de almindelige decimaltal, så ved vi, at første ciffer i et tal udtrykker antallet af enere, og næste ciffer gir antallet af tiere o. s. v. Tallet 43 betyder med andre ord 4× 10 + 3×1. I regnemaskinen har man givet cifrene nye betydninger. Her udtrykker det første ciffer antallet af enere, mens det næste ciffer er antallet af toere. Tredie ciffer er antallet af firere og det fjerde antallet af ottere. Man fortsætter hele tiden med at gange op med 2 for hvert ciffer — i stedet for at gange med 10, som vi er vant til. I dette *binære talsystem* skrives tallet 43 som 101011. Det betyder 1×32 + 0×16 + 1×8 + 0×4 + 1×2 + 1×1.



Decimale og binære tal: inden i regnemaskinen er alle decimale tal omskrevet til en serie et-taller og nuller. Disse binære tal svarer fysisk til en række elektriske strømme.

Disse binære tal, der består af en række nuller og et-taller, ser ved første øjekast lange og indviklede ud, men for regnemaskinen er de langt simplere at arbejde med end de normale decimale tal. De binære tal repræsenteres inden i maskinen ved *elektriske strømme*. Hvis tallet 43 skal sendes fra lageret over til regneenheden, sker dette gennem seks ledninger, én for hvert ciffer i det binære tal. Hvis der her går strøm i en ledning, svarer det til et binært et-tal, mens ingen-strøm svarer til et nul. Allerede her ser De en af de ting, der gør det umagen værd at bygge en maskine, der arbejder med binære tal: cifrene *ét* og *nul* kan repræsenteres ved to simple fysiske tilstande — enten går der *strøm* i en ledning, eller også går der *ingen strøm*.

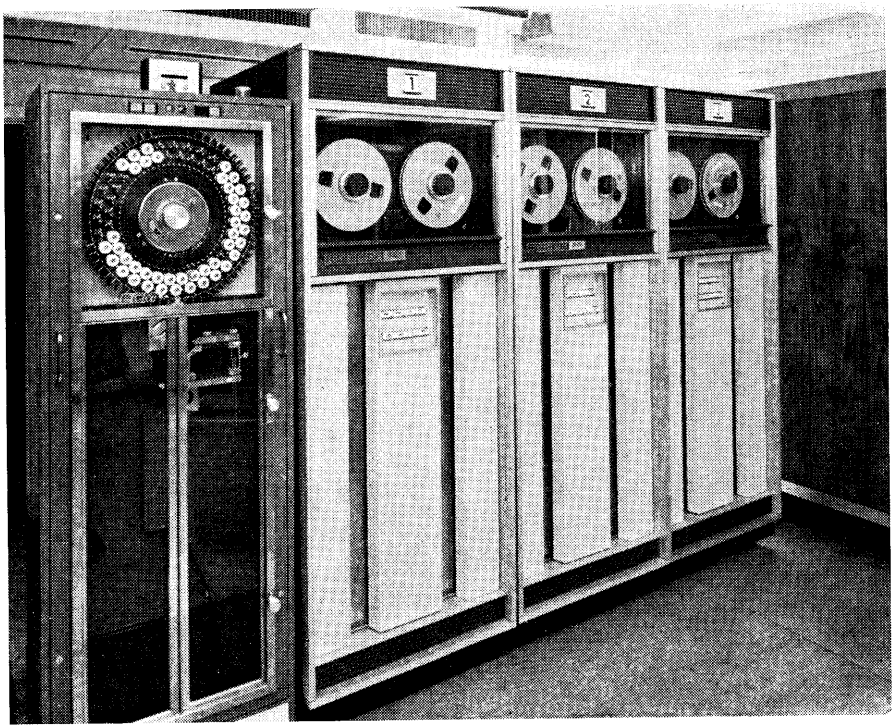


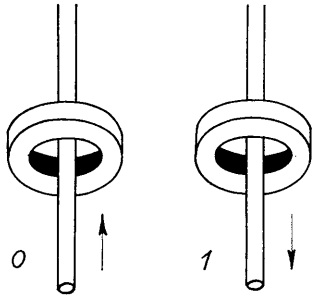
Magnetbåndslageret: regnemaskinen kan gemme tal på samme måde, som musik og tale gemmes på en båndoptager. Det sker ved at sende de binære tal som elektriske strømme ind i en såkaldt magnetbåndsstation.

Et tal kan altså *sendes* fra en del af maskinen til en anden, for eksempel fra skrivemaskinen til lageret eller fra lageret til regnemaskinen, som en række strømimpulser. Det næste spørgsmål er så, hvordan et tal kan *gemmes* i lageret.

Det er en nærliggende tanke at gemme tallene på samme måde, som man gemmer musik på en båndoptager. Idéen i en båndoptager er som bekendt den, at det, der skal gemmes, sendes ind i båndoptageren som en serie elektriske signaler. Den elektriske strøm vil her magnetisere et magnetbånd. Når båndet igen spoles tilbage og afspilles, blir magnetismen

En af de store magnetbåndsstationer, der anvendes som ekstralager, når regnemaskinen skal udføre beregninger med store talmængder. Magnetbåndet, der er 1 km langt, kan gemme 10 årgange af VOR VIDEN.





Ferritlageret: regnemaskinens hovedlager består af titusindvis af bittesmå magnetiske ringe. Hver af disse ferritkerner kan gemme enten et et-tal eller et nul. Lagringen sker ved at sende en elektrisk strøm enten den ene eller den anden vej gennem hullet i midten af kernen.

omsat til de oprindelige elektriske signaler. En båndoptager virker med andre ord som et lager.

Inden i regnemaskinen er de binære tal repræsenteret ved elektriske strømme, så vi kan også her bruge båndoptagerens princip til at lave et

DECIMAL ADDITION

MENTE	1
	58
	+ 27
SUM	85

BINÆR ADDITION

MENTE	1
	010
	+ 011
SUM	101

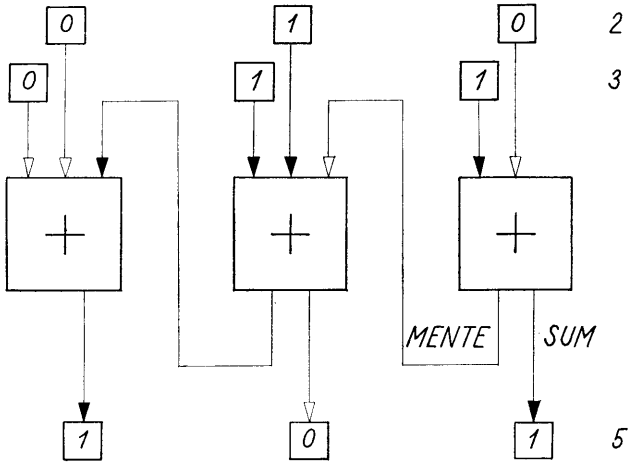
Med papir og blyant adderes to decimale tal ved at tage ét ciffer ad gangen og danne en sum og en mente. Regneenheden anvender samme princip, når den adderer to binære tal.

magnetbåndslager. Tallene kommer ind i båndoptageren på en lidt anden måde, end De før så det, idet et nul svarer til, at der går strøm den ene vej i ledningen, mens et et-tal svarer til, at strømmen går den anden vej.

Man bruger meget tit sådanne magnetbåndsstationer som lagre, når regnemaskinen skal udføre beregninger, der kræver et meget stort lager. På et enkelt magnetbånd kan man gemme op til 100 millioner nuller og et-taller. Desværre har magnetbåndene én alvorlig ulempe: På grund af den *mekaniske* båndspoling er de alt for langsomme sammenlignet med maskinens *elektroniske* enheder, regneenheden og styreenheden. For at finde ét bestemt tal på magnetbåndet kan man risikere at skulle spole helt til enden af båndet. Det tager flere minutter, og imens må styreenheden stå og spilde sin kostbare tid.

I praksis bruger man derfor også en anden type lager, *ferritlageret*, der er langt hurtigere. Ferritlageret består af en mængde små ringe, der er lavet af et magnetisk materiale. Disse *ferritkerner* er så små, at der kan ligge over hundrede af dem i et fingerbøl. I lageret sidder der titusinder af disse kerner. Gennem hver kerne går der elektrisk ledning. Hvis man sender en strøm den ene vej i ledningen blir kernen magnetiseret på én måde, svarende til at den nu gemmer et nul. På samme måde vil en strøm, der går den modsatte vej, magnetisere kernen på en anden måde, så den gemmer et et-tal. Hver lille ferritkerne kan altså gemme ét binært ciffer, der enten har værdien nul eller ét.

Den store fordel ved ferritlageret er, at styreenheden direkte kan få fat i ét bestemt tal ved at udvælge netop den ledning, der går gennem den ker-



Regneenhedens opbygning: de to rækker ferritkerner foroven indeholder de binære tal, der skal lægges sammen. Magnetismen i disse kerner omdannes til elektriske strømme, der sendes ind i en række ens transistorkredsløb. Hvert kredsløb lægger ad elektrisk vej to binære cifre sammen og danner en sum-strøm og en mente-strøm. Efter additionen står resultatet i den nederste række ferritkerner.

ne, hvori tallet er gemt. Man behøver ikke som ved magnetbåndet at lede gennem en masse tal for at finde de bestemte tal, man søger. Desuden arbejder ferritlageret rent elektrisk uden mekaniske bevægelser. Derfor er det langt hurtigere at arbejde med end magnetbåndene. Vi skal senere se nærmere på ferritlageret. Men først lidt om *regneenheden.*

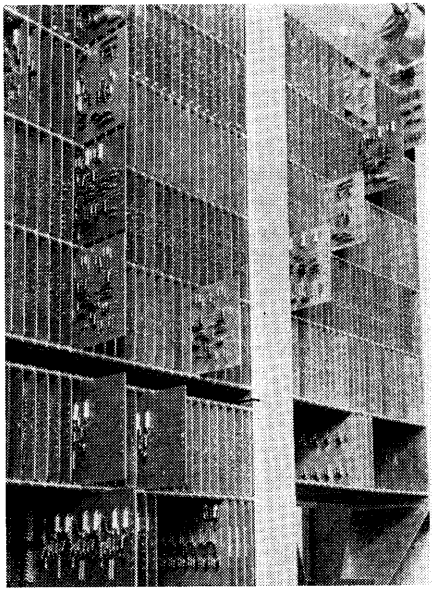
Når man lægger tal sammen med papir og blyant, sker det ved at tage ét ciffer ad gangen og danne en *sum* og en *mente.* Hvis vi for eksempel skal udregne 58 + 27, så starter vi med 8 + 7. Det gir 15, altså summen 5 plus 1 i mente. Efter det lægges de to næste cifre sammen med menten: 1 + 5 + 2 gir 8. Så har vi fundet den samlede sum 85.

Nøjagtigt det samme sker i elektronregnemaskinen. Her står tallene ganske vist i binær form, men det er også den eneste forskel. Lad os sige, at de tal, der her skal lægges sammen, er 2 + 3. De hedder på binær form henholdsvis 010 og 011, idet 0×4 + 1×2 + 0×1 = 2, mens 0×4 + 1×2 + 1×1 = 3.

Regneenheden begynder med at danne en sum og en mente af de to bageste cifre: 0+1 gir summen 1 og 0 i mente. Så tager den de to næste cifre 1+1. For de binære tal gælder den regel, at 1+1 gir summen 0 og 1 i mente*). Endelig adderes menten til de to forreste cifre: 1+0+0 gir 1. Det færdige resultat 101 er et binært femtal, idet 1×4 + 0×2 + 1×1 = 5, så resultatet stemmer.

Den følgende tegning viser, hvordan regneenheden er bygget op. Den

*) Det kan vises ved at betragte den decimale addition: 1 + 1 = 2. Når disse tre tal omskrives til binær form, står der direkte at læse at 1 + 1 = 10.



Regnemaskinens centrale dele, styreenheden og regneenheden, er sammensat af elektroniske byggeblokke. På hver af disse små plasticplader er der monteret et standardiseret transistorkredsløb. Optræder der en fejl i en af komponenterne, smides hele blokken væk og erstattes af en tvilling.

indeholder tre lokale lagerceller. I to celler foroven opbevares de to tal, der skal lægges sammen. I dette tilfælde gemmes to-tallet i den øverste celle som et binært tal 010, mens tretallet gemmes lige nedenunder som 011. Resultatet, der her blir et femtal, står efter additionen i den nederste celle som 101. De binære tal, der her skal lægges sammen, har hver tre cifre, så en lagercelle består i virkeligheden af tre små ferritkerner, der hver kan gemme et nul eller et et-tal.

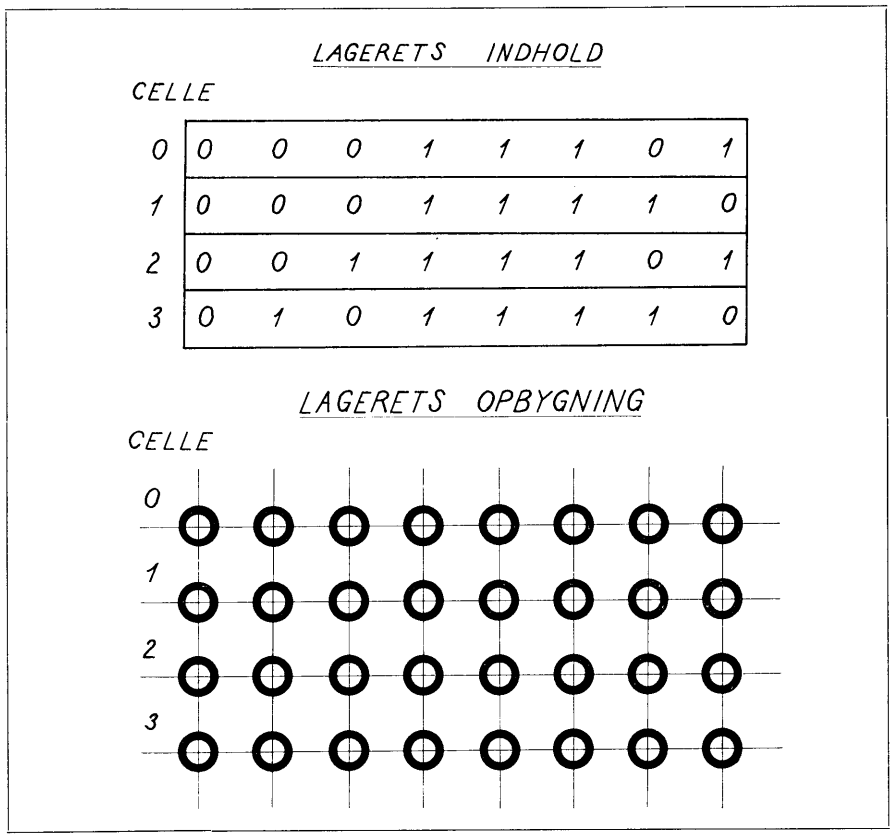
Selve additionen foregår i tre elektriske kredsløb. Hvert kredsløb kan lægge to cifre sammen og danne en sum og en mente. Disse elektriske kredsløb er meget simple og fuldstændig ens kredsløb, der arbejder med transistorer.

Når regneenheden har modtaget et startsignal fra styreenheden, begynder additionen. Først tager regneenheden de to bageste cifre. Der sker det, at magnetismen i de to bageste ferritkerner blir omdannet til elektriske signaler, der løber ind i den første additionsenhed. Ud af additionsenheden kommer som sum et elektrisk signal, der magnetiserer ferritkernen nedenunder. Der kommer også et mentesignal, der sendes videre til det næste kredsløb. Her adderes menten til indholdet af de to næste ferritkerner. Der dannes så igen en sum, der gemmes i en ferritkerne, og en mente, der løber videre til den forreste additionsenhed.

Når alle tre cifre foroven er blevet lagt sammen, står den samlede sum i de tre nederste ferritkerner. Fra denne lokale lagercelle kan resultatet så sendes tilbage til det egentlige lager og gemmes der.

Det egentlige lager, ferritlageret, består af et stort antal celler, der hver kan gemme ét tal. Nogle af disse tal opfatter maskinen som ordrer, andre opfatter den som tal, der skal regnes på. Men fælles for alle tal i lageret er det, at de består af en samling nuller og et'ere, for det er den eneste slags tal, maskinen kan arbejde med.

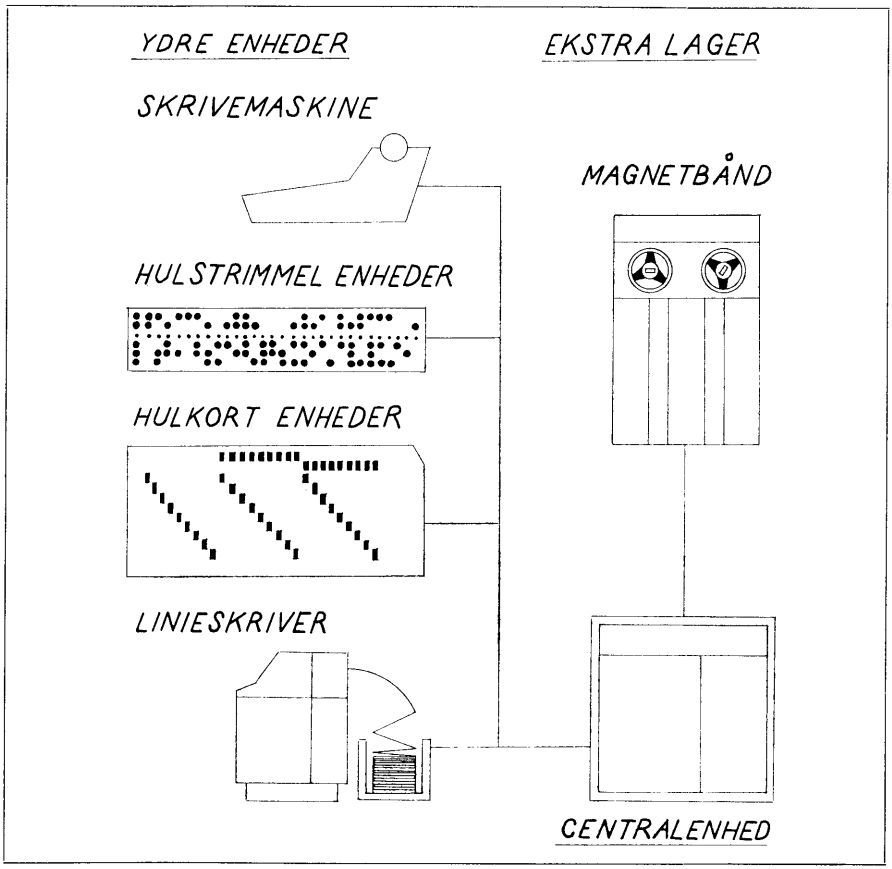
Tegningen af ferritlageret viser foroven, hvad lageret faktisk indeholder under en beregning — en forvirrende samling nuller og et-tallet. Vi har kun vist de første fire celler i lageret. I virkeligheden består lageret af flere tusinde celler. I dette eksempel kan hver celle gemme et binært tal med otte cifre. Nedenunder ser man, hvordan lageret fysisk er bygget op. Det består af en mængde ferritkerner, der sidder på et netværk af



Ferritlageret er opdelt i en række nummererede celler, der hver kan indeholde ét binært tal. Lageret består fysisk af en mængde ferritkerner, der sidder på et netværk af elektriske tråde. Hver celle består her af otte kerner, nemlig én for hvert nul eller et-tal, der kan gemmes i cellen.

elektriske tråde. Hver lagercelle består af otte ferritkerner, nemlig én kerne for hvert binært ciffer, der kan være i cellen. Ved at sende strømme gennem ledningerne kan man ændre kernernes magnetisme og på den måde *gemme* et nyt tal i lagercellen. Man har også mulighed for igen at omsætte kernernes magnetisme til de oprindelige elektriske strømme og således *læse* indholdet af en bestemt celle.

Den forenklede regnemaskine, vi her har betragtet, kan kun komme i forbindelse med operatøren gennem en elektrisk skrivemaskine. Alle tal, der skal ind i lageret, må operatøren taste på skrivemaskinen. Det gælder både de tal, maskinen opfatter som ordrer, altså selve *programmet*, og de tal, maskinen skal regne på, *datamaterialet*. Efter en beregning trykker skrivemaskinen selv resultater-



Til store regnemaskiner er skrivemaskinen alene for langsom som ydre enhed. Indlæsning af tal sker derfor fra hulstrimmel eller fra hulkort. Og de færdige resultater trykkes på en linieskriver med 1000 linier i minuttet. Endnu hurtigere er det at læse tal ind og ud fra magnetbånd.

ne ud. Det sker med en hastighed af 10 tegn i sekundet. Det lyder af meget, men er i virkeligheden alt alt for langsomt, når man sammenligner med maskinens egen elektroniske hastighed. Man har jo ikke meget glæde af, at regneenheden kan udføre 10 000 additioner på ét sekund, når det derefter vil tage 3 timer at trykke resultaterne!

I praksis er store regnemaskiner derfor udstyret med andre *ydre enheder*, som er langt hurtigere end skrivemaskinen. Og skrivemaskinen selv bruges kun, når man skal trykke nogle få resultater efter en beregning.

Strimmellæseren er en af disse ydre enheder. Det er et mekanisk apparat, der kan aflæse en hullet papirstrimmel. Hver hulrække på tværs af strim-

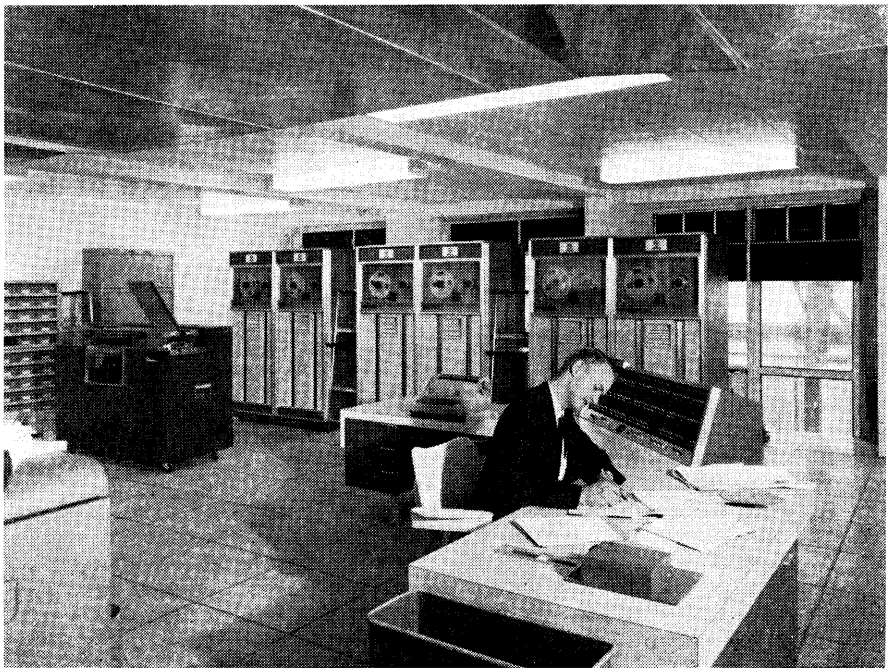
len svarer til ét tegn. Operatøren har i forvejen lavet hulstrimlen på en speciel skrivemaskine. Strimmellæseren er i stand til at læse hver hulkombination ved at sende lysstråler gennem hullerne. På bagsiden opfanges lysstrålerne af en række fotoceller, der laver dem om til elektriske signaler, som går ind i regnemaskinens lager. Strimmellæseren er 100 gange så hurtig som skrivemaskinen.

Regnemaskinen kan også læse tal, der er hullet på *hulkort*. Læsehastigheden er, ligesom for hulstrimler, på omkring 1000 tegn i sekundet.

Efter en beregning kan man vælge at lade regnemaskinen aflevere de færdige resultater på en hulstrimmel eller på hulkort, hvis man har lyst til det. Man kan dog også direkte få trykt resultaterne på en speciel skrivemaskine, der trykker en hel linie ad gangen. En sådan *linieskriver* trykker 1000 linier på ét minut.

Normalt kan en regnemaskine også udstyres med en eller flere *magnetbåndsstationer*. De bruges som ekstralagre ved meget omfattende beregninger, hvor selve ferritlageret ikke er stort nok. Tallene går ud og ind af en magnetbåndsstation med en hastighed af 80 000 tegn i sekundet. Med denne hastighed kan en hel årgang af *Vor Viden* læses på 25 sekunder!

Nu har vi fået maskinen forsynet med passende ydre enheder, men der er stadig en ting, der gør maskinen



En stor regnemaskine; bagved kontrolbordet med skrivemaskinen står til venstre en hulkortlæser og til højre seks magnetbåndsstationer.

MASKINKODE	SYMBOLSK KODE
00011101	TAST A
00011110	TAST B
00111101	TAG A
01011110	ADD B
01001111	GEM SUM
00101111	TRYK SUM
01100000	STOP

Maskinkode og symbolsk kode: et færdigt regnemaskine program består af en serie binære talordrer. For at lette program-forberedelsen indføres bogstavforkortelser for de enkelte ordrer.

uhyre besværlig at arbejde med, nemlig det, at den kun kender tallene nul og ét. Den opgave, vi i starten stillede maskinen: at lægge to tal sammen, krævede, at vi først sammensatte et program af de ordrer, som maskinen er i stand til at udføre. De ordrer, maskinen kan forstå, har form af en samling talkoder, der oven i købet skal skrives i binær form for at blive accepteret af maskinen.

Tilsyneladende tvinger maskinen os altså til at skrive vores program i en kode, der består af en uoverskuelig samling nuller og et-tallet. I det aktuelle tilfælde skal programmet se ud, som man ser det til venstre på den næste tegning. Det er den form, det skal have i lageret, for at maskinen kan udføre det. Når man skal skrive store programmer med mange hundrede ordrer, blir denne rene *maskinkode* fuldstændig uoverskuelig. Man kommer uundgåeligt til at lave kodefejl i sit program, og disse fejl er meget svære at lokalisere og rette.

Det ville være langt hurtigere og sikrere at skrive et program, hvis vi kunne benytte os af et mere menneskeligt kodesprog. Man benytter derfor den fremgangsmåde først at skrive ordrerne på en letlæselig måde, idet der indføres bogstavforkortelser for maskinens enkelte operationer. I denne *symbolske kode* ser vores additionsprogram ud, som vist til højre på figuren.

Den symbolske kode lider nu af én alvorlig mangel: Selv om den er bekvem for brugeren, forstår maskinen stadig kun sine nuller og et-taller. Før maskinen kan udføre programmet, må det derfor på en eller anden måde ordre for ordre oversættes fra den symbolske bogstavform til maskinens egen binære form. Denne slaviske oversættelse er et uhyre trivielt arbejde, så trivielt, at man uvilkårligt spørger sig selv: kan maskinen ikke på en eller anden måde hjælpe os med dette? Denne problemstilling har givet stødet til udviklingen af det mest epokegørende fremskridt inden for programmeringsteknikken — de såkaldte *oversætter-programmer.*

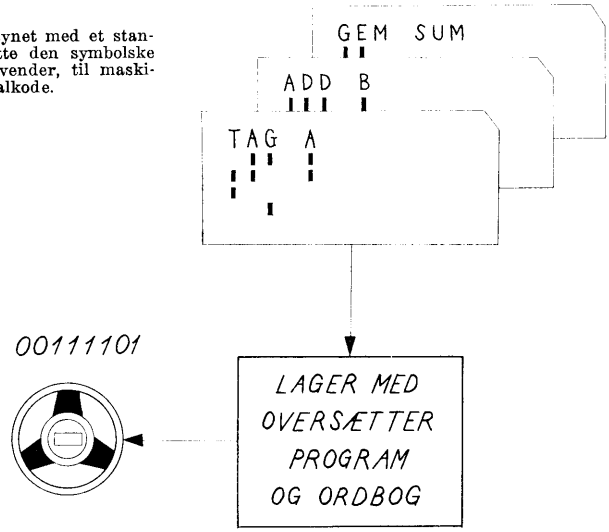
Oversætter-programmet er et standard-program, der fra fabrikantens side leveres sammen med regnemaskinen. Dette standardprogram, der altid ligger klar i maskinens lager, er i stand til at oversætte den symbolske kode til maskinens eget sprog. Lad os se, hvordan dette oversætterprogram arbejder. Vi går nu hen til maskinen med vores eget lille additionsprogram. Programmet har vi skrevet på hulkort, således at der på hvert hulkort står én symbolsk ordre, f.eks. TAG A, ADD B, GEM SUM o. s. v. Så starter vi regnemaskinens oversætterprogram. Under oversættelsen læser maskinen ét hulkort ad gangen. Hulkombinationen på kortet skal nu oversættes til en maskinordre, der består af nuller og et'ere. I lageret ligger der en »ordbog«, der bestemmer, hvilken samling nuller og et-taller hver enkelt symbolske ordre skal oversættes til.

Maskinen læser for eksempel et hulkort, hvorpå der står TAG A, og så leder oversætter-programmet i ordbogen og finder ud af, at denne symbolske ordre skal oversættes til maskinordren 0111101. Den oversatte kode gemmes på et magnetbånd.

Når maskinen har læst alle hulkortene, er den egentlige oversættelse slut. Vi har nu på magnetbåndet stående en samling nuller og et-taller, som maskinen kan anerkende som et rigtigt program. Men vi har endnu ikke fået *udført* dette program; vi har kun fået det *oversat*. Umiddelbart efter oversættelsen spoler maskinen derfor magnetbåndet tilbage og læser derefter det oversatte program ind i ferritlageret. Når programmet først er anbragt dèr, begynder maskinen at udføre ordrerne én for én.

Da man først havde fundet ud af at lave programmer, der kunne oversætte symbolske bogstavordrer til maskinens egen talkode, gik man et skridt videre, idet man sagde: hvorfor i grunden tvinge brugeren til at formulere selv den simpleste addition som en serie primitive *ordrer*, TAG A, ADD B, GEM SUM? Det ville føles langt mere naturligt at beskrive dette med en simpel matematisk *formel*, altså: SUM = A+B.

Denne idé til mere *avancerede programmeringssprog* resulterede i starten i et sandt Babelstårn af nye regnemaskine-sprog. Hver fabrikant fulgte nemlig sit eget hoved, og der fremkom ustandselig nye programmeringssprog med kryptiske navne som FACT, FLOWMATIC, INTERCOM, IT, LOGLAN og MAD. Endnu i dag er det ikke lykkedes de store regnemaskinefirmaer at nå til blot nogenlunde enighed om et standardsprog for regnemaskiner. En af årsagerne hertil er de store økonomiske interesser, der ligger bag de eksisterende sprog. Det er kostbart for fabrikanten at udvikle oversætterprogrammer til nye sprog, og det er heller ikke gratis for brugerne. Disse skal først oplæres i an-



Moderne regnemaskiner er forsynet med et standardprogram, der kan oversætte den symbolske bogstavkode, som brugeren anvender, til maskinens egen binære talkode.

Et FORTRAN Program

```
C PROGRAM TO SOLVE N SIMULTANEOUS EQUATIONS BY
C SEIDEL ITERATIONS.  N MUST NOT EXCEED 40.
. . . . .
C START AN ITERATION
   34 ERROR = 0.0
      DO 35 I = 1, N
      SUM = 0.0
      DO 40 J = 1, N
   40 SUM = SUM + A(I, J) * X(J)
   39 TEMPX =(B(I) - SUM + A(I,I) * X(I))/A(I,I)
      ERROR = ERROR + ABSF(X(I) - TEMPX)
   35 X(I) = TEMPX
. . . . .
```

Eksempler på programmeringssprogene Fortran, Algol og Cobol. Når disse programmer skrives på hulkort, kan de puttes direkte i elektronregnemaskinen, som selv oversætter dem til en binær maskinkode.

Et ALGOL Program

```
procedure sturms sequence;
begin pl:=0; ql:=1; al:=0;
for i:=1 step 1 until n do
    begin y:=(c[i] -lambda)×ql-p[i]×pl; pl:=ql; ql:=y;
    if pl≥0=ql≥ then al:=al+1;
    comment counting of aggreements in sign;
    end i;
if ql=0 then
    begin if pl>0 then al:=al-1 else
          if pl=0 then
          begin tridibisection:=false; go to EXIT; end
    end
end of sturms sequence;
```

Et COBOL Program

```
PROCEDURE DIVISION.
START. OPEN INPUT TRANSACTION-FILE, MASTER-FILE,
       OUTPUT REPORT-FILE.
TRANSACTION-READING. READ TRANSACTION-FILE. IF MODEL-NUMBER
       IN TRANSACTION EQUALS 99999 GO TO WRAPUP.
MASTER-READING. READ MASTER-FILE. IF MODEL-NUMBER
       IN TRANSACTION UNEQUAL TO MODEL-NUMBER IN MASTER
       GO TO MASTER-READING. MULTIPLY QUANTITIY IN
       TRANSACTION BY UNIT-PRICE IN MASTER GIVING
       TOTAL-PRICE. IF TOTAL-PRICE IS GREATER THAN 100.00
       PERFORM DISCOUNT-CALCULATION. MOVE NAME IN
       TRANSACTION TO NAME IN REPORT. . . . .
```

vendelsen af det nye sprog, og må så i tilgift indstille sig på at skulle skrive alle deres tidligere programmer om.

Selv om egentlige standardsprog altså ikke eksisterer, er der dog i øjeblikket tre programmeringssprog, der har en større udbredelse end de øvrige. Et af de første avancerede sprog var FORTRAN *(Formula Translation)*, udviklet af det amerikanske gigantfirma IBM til beskrivelse af matematiske og videnskabelige beregninger. En videreudvikling heraf er ALGOL *(Algorithmic Language)*, der er udviklet af europæiske og amerikanske forskere i fællesskab. ALGOL er blevet noget nær et internationalt programmeringssprog. For eksempel publicerer en række tidsskrifter i Skandinavien, Tyskland og USA programmer skrevet i ALGOL.

Af amerikansk oprindelse er også sproget COBOL *(Common Business Oriented Language)*. Det er skabt af en komite nedsat af det amerikanske forsvarsministerium. COBOL er i øjeblikket det eneste standardsprog til administrativ databehandling. Desværre har det i sin nuværende form iøjnefaldende svagheder. Det er dårligt defineret og temmelig indviklet at lære.

Disse programmeringssprogs enestående betydning er den, at de gør programmeringen af elektronregnemaskiner til en dagligdags ting. I regnemaskinens første år blev det med god grund sagt, at maskinen ikke alene var temmelig *indskrænket*, men også direkte *utilnærmelig*. Mennesket måtte enten snakke med maskinen på dens eget binære sprog — eller helt lade være. I dag er programmering ikke mere et håndværk for specialister. Enhver kan lære at kode i FORTRAN eller ALGOL på nogle få dage.

Læsere, der her har savnet oplysning om regnemaskinens praktiske *anvendelse* bør opspore Carl Erik Fröberg's bog om »Datamaskiner« i skriftserien TEMA, Stockholm 1962.