

Problemet generell datamaskin och en lösning – D21

Ingenjör Börje Langefors, Linköping

Varje datamaskin av normal typ kan utföra samma bearbetning som varje annan — om man bortser från att bearbetningen kan kräva olika lång tid. Som regel byggs emellertid datamaskinerna så att de blir ekonomiskt effektiva endast för en viss problemklass.

Det finns flera skäl för att man skall söka bygga en verkligt allmänt användbar maskin. En maskin som kan användas ekonomiskt för stora beräkningsuppgifter å ena sidan och för enkla datahanteringsrutiner å den andra kan säljas i större antal. Trots olikheten i företag och personal som sysslar med enbart det ena eller enbart det andra slaget av arbete har alla fördel av att använda samma maskin- och programsystem. Undervisning och service blir enklare och säkrare. Stimulans kan vinnas från olika användningstyper. Många företag har inom sig båda problemtyperna. Det är för dem fördelaktigt att kunna använda en och samma maskin och driftpersonal för båda områdena. Är företaget stort nog för två anläggningar erbjuder fortfarande två anläggningar med identiska centralenheter mycket stor fördel.

Slutligen är det viktigt, ehuru ofta förbiset, att beräkningsuppgifter kräver även omfattande databehandling för att bli automatiserade i full utsträckning. Samtidigt utvecklas kommersiell databehandling snabbt — i progressiva företag — till att omfatta alltmer matematiskt betongade behandlingar. I denna utveckling blir lätt en maskin som synes ha tillräcklig kapacitetsmarginal för kommersiella problem helt otillräcklig. Numer är det också aktuellt för alla uppdragstyper, att man programmerar i en autokod som av maskinen skall översättas till maskinkod. Detta är en ren datahanteringsoperation.

De resonemang som här helt kort har refererats ligger bakom konstruktionen av Saabs allmänna datamaskin¹, D21. Det är intressant att med bakgrund av de erfarenheter som har vunnits vid byggandet och driften av D21 diskutera frågorna: Går det att göra en maskin allmänt användbar och ändå ekonomisk? Hur har man löst detta problem i D21? Svaren på dessa frågor kan man nå genom att studera de special-

betonade drag, som man tidigare ansett nödvändiga hos datamaskiner för speciella problemområden.

Databehandling i vid mening är ett komplicerat område. Detta medför svårigheter för en snabb överblick, vilket har lett till att det finns ett myller av förhastade och ytligt motiverade uppfattningar om datamaskiner och om de olika databehandlingsproblemens särmerken. Vi måste tyvärr visa bristerna hos en del av dessa motiveringar innan vi kommer fram till rimlig bedömning av det invecklade problemkomplexet.

Särdrag hos kommersiell databehandling och vetenskapliga beräkningar

All databehandling synes avse data vilka beskriver egenskaper hos ett antal objekt. Exempel på sådana objekt utgör detaljer av vilka ett företag tillverkar sina produkter. En typisk kommersiell databehandlingsrutin kan avse redovisning av antalet i lager av var och en av dessa detaljer.

Bearbetningen sker här för en detalj i taget varvid en datapost avseende ändring i förrådshållningen, jämte en post ur förrädsregistret, utgör processens ingångsdata. För varje detalj genomgås processen på nytt. Varje detalj är i denna process oberoende av alla andra. Den totala bearbetningen kommer sålunda att utföras såsom en serie enkla bearbetningar, var och en avseende en enda detalj. Om ändringsposterna och förrädsregisterposterna ordnats i samma ordning beträffande detaljnumren kan bearbetningen ske i samband med en enkel genomgång av berörda register.

Vid detta slags bearbetning behöver endast en post från varje register rymmas samtidigt i minnet. Detta kräver endast ett par hundra teckenpositioner i minnet. Emellertid kommer då effektiv bandkommunikation att ställa större krav.

En typisk teknisk databehandlingsuppgift kan avse något konstruktionsproblem som sammanhänger med hopbyggandet av en grupp av detaljer till en slutprodukt. Här påverkar olika

detaljer varandra, varför bearbetningen ej kan ske lika enkelt som i förra fallet. Man måste kunna komma åt olika poster i varierande ordning. Registren kan få svepas över fram och tillbaka ett stort antal gånger. Ett stort minne underlättar avsevärt denna hantering. Vid planering av detaljernas och produkternas tillverkning får man också ett samband mellan olika detaljer, t.ex. tillverkningsordning och tidpunkter. Här blir åter ett mera komplicerat sökande i registret nödvändigt.

Det har sagts — så länge att det nu kommit att på sina håll betraktas som fastställt — att de vetenskapliga beräkningarna kännetecknas av komplicerad aritmetik och enkel datahantering. Den kommersiella bearbetningen säges, på liknande sätt, kännetecknas av enkel aritmetik men komplicerad datahantering.

Vi har redan sett att denna enkla karakterisering är bristfällig. Vi skall visa att den är helt ohållbar. Bland de mest typiska bearbetningarna inom de större vetenskapliga beräkningar som göres i datamaskiner, är matrisoperationer eller matrisliknande operationer. Ett enkelt exempel utgör matrisadditionen $A + B = C$. Denna kan tillgå så att matriserna A och B ligger på var sitt magnetband. C lägges ut på ett tredje band. Operationen kräver en samtidig, enkel genomgång av A -, B - och C -banden. Detta är exakt samma sak som sker i en typisk kommersiell registervårdsoperation (uppdatering).

I programmeringssystem för matrisoperationer sådana som t.ex. Saabs "Samba", som i många år använts vid Sara, programmeras detta i princip så att man helt enkelt skriver

$$A + B = C;$$

Aritmetik och bandhantering klaras sedan helt automatiskt av programsystemet. För en kommersiell registervårdsoperation av i princip samma slag, med exakt samma bandhanteringstyp, fordras ett betydligt längre program även om detta skrivs i Cobol (Common business oriented language) som är ett programmeringssystem helt och hållet gjort för sådan enkel registervård.

Varför blir då registervården mera svårprogrammerad? Jo, den har i allmänhet mera varierande och komplicerad aritmetik och denna kan därför ej återges i standardformler.

Det är alltså den kommersiella behandlingen som kännetecknas av den mera komplicerade aritmetiken. Datahanteringen är exakt lika för båda, i de diskuterade exemplen.

Om vi nu betraktar matrismultiplikation i stället, t.ex. $A \cdot B = C$, så klaras denna formel lika lätt helt automatiskt i matrisprogrammeringssystemet. I detta fall kräver bearbetningen normalt en väsentligt mera komplicerad datahantering än den enkla, kommersiella registervården.

Detta belyses därav att bandregistren för A och B uppdelas i delar och för varje del av A skall hela registret B genomgå så att endera A eller B kommer att behöva svepas över ett

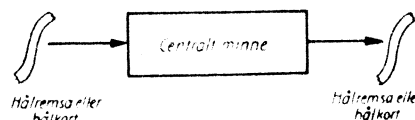


Fig. 1. Principskiss över maskin för automatisk behandling av matematiska problem.

stort antal gånger efter ett visst schema. Vid multiplikationen kommer alltså matrisens element att få ett relativt komplicerat samband med varandra. Detta sammanhänger för övrigt med att matrismultiplikation just är en typ av bearbetning som berör detaljer vilka har sammanhang med varandra.

Slutsatsen blir följande: Vi har funnit att i våra typexempel det matematiska problemet hade enklare aritmetik men betydligt mera komplicerad datahantering än det kommersiella. Detta är tvärt emot vanliga föreställningar och påståenden. I den mån man kan leta upp andra exempel med en annan tendens, så kan detta på sin höjd bevisa att ingen generellt giltig karaktärsskillnad kan konstateras.

Den komplicerade datahanteringen vid vetenskapliga beräkningar kan ibland koncentreras till datamängder av t.ex. 20 000 ord. Sådan bearbetning kan då klaras helt i snabbminnet på en stor maskin men kräva magnetbandshandling vid en mindre. Ett stort snabbminne medför naturligtvis i sådana fall en avsevärt minskad bearbetningstid. Den interna hastigheten får också i sådana fall accentuerad betydelse. För problem av denna typ kan en maskin vara 100 gånger snabbare än en annan även om de har helt likvärda in- och utmatningsorgan. Problem av denna typ kan också förekomma som delproblem i kommersiella rutiner, t.ex. i form av tabellslagning eller vid planeringsuppgifter.

Datamaskiner för olika problemområden

Matematiska maskiner

Vid konstruktionen av maskiner för matematiska problem, fig. 1, de datamaskiner som tillverkades först, lade man tyngdpunkten vid största möjliga minne och snabbhet. Man föredrog att använda binär aritmetik på grund av att maskinen då blev enkel och snabb. Minnena var emellertid från början små av tekniska skäl.

Man har i detta område en strävan till utbyggnad med större minne, stor ordlängd, "flytande räkning" samt index-aritmetik, fig. 2.

Det är för förståelsen av den fortsatta dis-

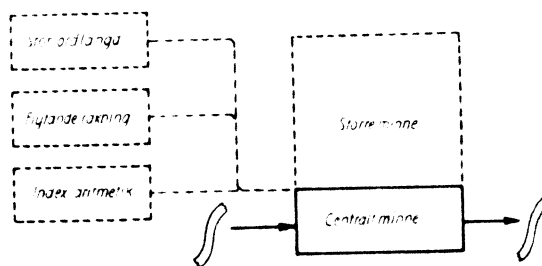
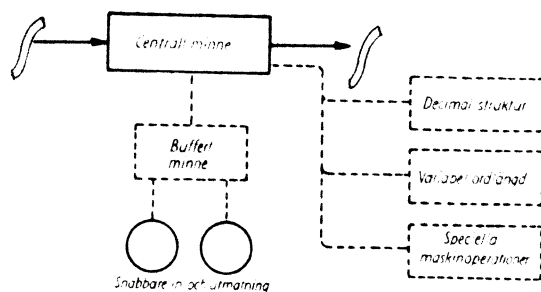


Fig. 2. En beräkningsmaskin utbyggd med specialfunktioner.

Fig. 3. En beräkningsmaskin utbyggd för kommersiella rutiner.



kussionen ej väsentligt att förstå innebörden av de olika angivna specialfunktionerna. Det räcker med att inse att realisering av dessa funktioner med hjälp av speciell utrustning kostar pengar. T.ex. innebär större ordlängd en dyrare aritmetisk enhet.

Låt oss emellertid i förbigående konstatera att dessa specialfunktioner syftar till att göra datamaskinen mera lämplig för beräkningsarbete genom ökade prestanda och större precision samt genom att underlätta programmeringen. Låt oss också konstatera att när det ibland framhålls att en maskin för matematiska beräkningar måste ha minst 48 bitars ordlängd så är detta åter ett exempel på förhastade konstateranden. Användningen kan kräva en viss precision. Denna kan nås med stor ordlängd men också med andra medel.

Kommersiella maskiner

När maskiner började konstrueras för kommersiella rutiner lades vikt vid andra specialfunktioner: decimal struktur, senare alfanumerisk; variabel ordlängd; snabbare in- och utmatning, magnetband, snabbskrivare; buffertminnen för samtidig operation av yttre organ samt speciella maskinoperationer (instruktioner), fig. 3. Man ville t.ex. ge maskinen decimal struktur för att den skulle bli lättare att programmera och för att spara tiden för översättning av inmatade decimala data till binär form.

Variabel ordlängd skulle underlätta programmeringen av kommersiella dataposter med variabelt långa termer och möjliggöra kompakt lagring på magnetband vilka därigenom kan överföra informationen på kortare tid.

Här ligger återigen det väsentliga i att man inser att anordningarna i fig. 3 kostar pengar och införs för att ge ökade prestanda och enklare användning, vid kommersiella problem.

Det kan vara av intresse att här åter peka på en missuppfattning som fått ganska allmän spridning. Det gäller decimala maskiners fördelar ur kommersiell synpunkt. I verkligheten torde en decimal maskin ha vissa fördelar i samband med beräkningsarbete. För kommersiella användningar har decimalmaskinen två nackdelar. Alfabetiska data blir klumpigt behandlade — vilket ej är fallet i binära maskiner — och dessutom finns, särskilt i kommersiella problem, gott om storheter som endast kan ha två eller tre värden och där alltså binär lagring är påtagligt fördelaktig. Ett enkelt exempel: Att ange om en person är man eller

kvinna fordrar endast en binär position. En decimal position motsvarar fyra binära och är i sådana sammanhang slöseri.

Lättkodning

De berörda specialanordningarna inbyggda i olika datamaskiner utgör i viss utsträckning resultatet av en strävan att bygga maskinerna så att de kan instrueras med ett kodningsspråk som blir allt bekvämare för användaren.

Så småningom har utvecklats en teknik som innebär att man lägger in specialfunktionerna med ord i maskinens minne i stället för med tråd i maskinens styrorgan. Det är klart att detta innebär att maskinen därvid får genomlöpa ett specialprogram, basprogram, för att utföra en funktion som i inbyggt skick skulle fordrat endast en instruktion. Detta kräver då minnesutrymme och tid. Därav kommer att större snabbhet ger bättre möjlighet för denna teknik.

Det är dock ett misstag att tro att sådana basprogram skulle behöva läggas in på varje punkt i programmet där de används. Sådan dålig teknik förekommer, men det naturliga är att man lägger ut varje basprogram en gång i minnet och lägger enkla hoppinstruktioner i huvudprogrammet som kopplar in basprogrammen vid behov. Resultatet blir ett mycket begränsat minnesbehov för basprogrammen. Detta blir också oberoende av programmets storlek.

Specialfunktionerna som realiserats med ord (basprogram) kan bättre anpassas till användningen. Slutligen har utvecklingen fört fram till lättkodningssystem som gör det möjligt för användaren att instruera maskinen i ett språk som anpassats mera till hans problem än till maskinens egenart. Särskilda översättningsprogram användes för att översätta det skrivna programmet till maskinens språk. Sådana översättare eller kompilatorer måste avstå från att använda ett flertal i maskinen inbyggda specialfunktioner. De avancerade lättkodningssystemen leder av detta skäl till märkbar minskning av kapaciteten hos maskiner som är beroende av inbyggda specialfunktioner eller speciella egenskaper.

Detta har t.ex. medfört att ett antal maskiner vilka från början konstruerats med tanke på kommersiell databehandling måste avstå från att i nämnvärd omfattning utnyttja sådana värdefulla hjälpmedel som t.ex. Cobol trots att detta lättkodningssystem just utvecklats för kommersiell databehandling. Cobol gör det möjligt att skriva program för databehandling i ett språk som nära ansluter sig till engelska. Fördelen härav är dock mycket skenbar. Viktig är däremot Cobols automatiska handläggning av in- och utmatning samt hantering av varierande dataformat och dess funktioner för bekväm programadministrering.

Den sist nämnda fördelen erbjuder i än högre grad Algol (Tekn. T. 1960 s. 1087), som dock från början utformats för automatisk programmering av formler.

längd är ett — som vi skall se ej ens det bästa. Små minnen å andra sidan (vanliga i maskiner för kommersiell bearbetning) motverkar kompakt lagring genom att tvinga till korta datablock på banden. Det finns maskiner med variabel ordlängd, vilka har så litet minne att packningen på band blir så liten, på grund av korta block, att många gånger mera förloras härigenom än som vinnes genom variabla ordlängden.

En annan vanlig uppfattning, som det finns anledning vara på sin vakt mot, är att problemet kompakt lagring på magnetband skulle vara specifikt för kommersiell bearbetning. I verkligheten är detta problem specifikt för användningen av magnetband.

Eftersom dessa fördomar om nödvändigheten av variabel ordlängd — realiserad genom användning av minnen med direkt adressbara positioner för alfanumeriska tecken (teckenmaskiner) — kommit att bli djupt rotade på många håll torde det vara nödvändigt att något ingående studera detta problem, dvs. problemet om kompakt lagring på band.

Problemets bakgrund är att olika slag av data kräver olika utrymme. Data har alltså själva variabel längd. Det finns härvid två olika slag av variation. Ett ligger i att olika dataslag har olika utrymmesbehov vid ett och samma tillfälle (statisk längdvariation eller längdolikhet). Dessutom kan samma dataslag kräva olika längd vid olika tillfällen (dynamisk längdvariation). Exempel på statisk längdolikhet är: Uppgift om år kräver fyra decimala tecken, uppgift om namn t.ex. 30 alfabetiska tecken. Exempel på dynamisk längdvariation är: "Bo Ek" kräver fem alfanumeriska tecken. Nästa förekommande namn kan vara "Börje Langefors" som kräver 15 tecken, i båda fallen inklusive tecknet för mellanslag.

När data med olika längd blandas om varandra kräver naturligtvis datamaskinen information om de olika datas lägen, för att kunna bearbeta dem. Härvid sammanföres alla data som ingår i en bearbetning till en post. I ett förrädsregister t.ex. sammanföres alla data om en och samma detalj till en post, vilken identifieras med detaljens nummer. Om en sådan post inläses från ett magnetband till maskinens minne så att postens första tecken kommer i en förutbestämd minnesposition räcker en lista över postens olika data, i den ordning de förekommer, samt information om respektive datalängder till för att datamaskinprogrammet skall kunna lokalisera varje enskild data-term och hämta den för bearbetning.

Om data har dynamiskt variabel längd måste givetvis varje post själv medföra information om längderna. Vid statisk datalängd kan informationen alternativt vara inlagd i programmet. Av detta följer att dynamiskt variabla data ej medför fördelar vid korta termer.

Längdinformationen kan ges antingen med hjälp av ett särskilt tecken i varje term vilket markerar termens slut "word mark" eller genom separata längdinformationstermer. Maski-

ner med variabel ordlängd konstrueras så att varje minnesadress åsyftar en grupp om t.ex. sex binära minnespositioner (t.ex. magnetkärnor). En sådan grupp kan ange 64 olika kodkombinationer som var och en kan motsvara ett tecken dvs. en siffra, bokstav eller en typografisk symbol (t.ex. "?").

I sådana teckenmaskiner användes ofta en särskild kodkombination (om sex positioner) såsom slutmärke. Därvid förlänges varje term med detta tecken, alltså med 6--7 (binära) positioner och man får alltså ett extra minnesbehov, ett "gränspålägg". En teckenmaskin kan i stället konstrueras så att i varje tecken tillägges en extra binär position för gränsmarkeringen. Man får i detta fall alltså ett gränspålägg av en position på sex. I denna konstruktion blir det emellertid möjligt att låta gränsmärkena och de extra positionerna därför, endast förekomma i minnet och ej på bandet. Bandlagringen befrias då från gränspålägget.

Maskiner för kommersiell databehandling, vilka ej har inbyggd variabel ordlängd, brukar förses med särskilda operationer som förmår att ur ett adresserat ord hämta en angiven del — varvid alltså ett särskilt utrymme i instruktionen fordras för att, inom ordet, adressera första och sista positionen av den önskade ord delen. I sådana maskiner kan flera termer packas i samma ord. Det är dock därvid ej lämpligt att låta någon term skära ordgränsen. Programmeraren får alltså vid dessa maskiner lägga pussel med termerna för att fylla orden så bra som möjligt. Alla ord blir i regel ej fyllda varför man här åter får en form av gränspålägg.

I D21 användes ingen av de här beskrivna anordningarna. Data lagras trots detta fullkomligt kompakt på bandet. Ordgränserna nonchaleras därvid. För statiska data uppstår på bandet inget gränspålägg. Dynamiskt variabla data åtföljs av längdinformation. Informationen för lokalisering av de olika termerna inmatas till programmet som medelst en särskild "uppackningsdel" hämtar termerna, en efter en, från den minnesarea dit de matats från bandet, och lägger dem i en serie separata ord.

D21's orderstruktur underlättar denna operation, dess snabbhet gör att den endast tar ringa tid i förhållande till bandtiden. Medan denna metod ej kräver gränspålägg på bandet, vilket är det väsentliga, motsvarar den erforderliga längdinformationen i programmet ett gränspålägg i minnet. Ett block av data på magnetband innehåller normalt ett stort antal poster. För varje band reserveras normalt två inmatningsareor i minnet vilket fördubblar antalet postutrymmen i minnet med två. För flertalet av poster krävs normalt ett och samma uppackningsprogram, varför dettas längdinformation försörjer flera poster. På detta sätt reduceras gränspålägget per post i minnet. I "teckenmaskiner" erhålles för övrigt också en motsvarighet till detta gränspålägg vid programmet (utöver gränstecknet) därigenom att adressering per tecken i stället för per ord om flera

Vid Saab har utvecklats ett generellt in-utmatningssystem, förkortat Genius, vilket utformats så att det erbjuder automatiserad in-utmatning och datapackning, såsom i Cobol, till varje annat programmeringsspråk. Sålunda blir Genius plus Algol därigenom ett effektivt och bekvämt system både för kommersiell och teknisk databehandling. Vid Saab har dessutom utvecklats en grundläggande autokod DAC, vilken ger tillgång till alla maskinens interna instruktioner och därför gör maskinkod onödig.

Som nämnts förekommer tekniska problem som behandlas med matrisymbolik. Vid dessa blir programmering helt onödig eftersom man har utvecklat programsystem som direkt kan bearbeta matrisformlerna. Det kan väntas att liknande matematisk teknik kommer att utvecklas för administrativa system, varvid även där programmering kan bli onödig. Jag har för övrigt sedan 1953 studerat möjligheter att formulera tekniska strukturproblem i en mera generell kategori. Algebraisk topologi kom därvid till användning och innebar en mera allmän användbarhet av utvecklade matrismetoder. Härvid var just användning i administrationsstrukturer ett av målen. Senare har på flera håll en liknande teknik i en enkel form kommit till användning för vissa planeringsuppgifter (Pert, Critical Path; Tekn. T. 1962 s. 737).

En generellt användbar datamaskin

Om man vill göra en datamaskin för alla användningsområden och därvid skulle bygga in alla de specialfunktioner, som behövs för det ena och det andra slaget, skulle lätt maskinen bli oekonomisk. Vid användning i ett problemområde blir anordningarna som avsetts för det andra outnyttjade. Saabs datamaskin D21 bygger på att alla specialfunktionerna kan undvaras eller, rättare sagt, kan realiseras programmeringsvägen om maskinen göres snabb och ej har för litet minne. En sådan maskin blir enkel och ekonomisk för alla användningsklasser. Maskinen omställs från en problemtyp till en annan genom att man matar in andra standardprogram. Specialfunktionerna hade till syfte att öka prestanda. Det torde vara uppenbart att ökad hastighet och större minne ger samma möjlighet härvidlag.

Specialfunktionerna skulle också förenkla programmeringen. Detta kan emellertid, som vi sett, också ske genom att man profiterar på snabbhet och minnesutrymme samt på vissa enkla och generellt användbara konstruktionsdrag. D21 har indirekt adressering med och utan adress-stegning, indirekt beordring ("utför ordern"), hopp med lagring av returadress, varvräkning som exempel på effektiva funktioner för lättkodanvändning, vilka kräver obetydlig extra utrustning. Man kan uppnå ej blott samma bekvämlighet för programmeraren utan större. Detta gäller därför att maskinens allmäninriktning också gör den effektiv vid användning av de moderna automatiska program-

meringssystemen, vilka har visat sig svåra att utnyttja i mera specialinriktade system.

Vid användningen av bekväma kodningshjälpmedel, vilka t.ex. ingår som viktiga element i avancerade lättkoder, blir ofta snabb aritmetik av stort värde. Denna behöver därvid ej arbeta med stor ordlängd. Ett enkelt exempel härpå är användningen av tabelldata. Låt oss t.ex. anta att man har en pristabell i vilken pris anges för ett antal kvaliteter med en tabellrad för varje och ett antal dimensioner, med en kolumn för varje. Priset för kvalitet 7 och dimension 13 skulle bekvämt kunna skrivas Pris (7,13). Detta tilläts t.ex. i Cobol. Datamaskinen räknar fram adressen för den aktuella uppgiften genom att till adressen för tabellens första element lägga sex gånger antalet minnesord i en tabellrad plus tolv gånger antalet ord i varje tabellelement. Kompilatorn kommer alltså att lägga ut en adressräkning som omfattar multiplikation och addition. Det är klart att vid en maskin, som är mycket långsam i multiplikering, vilket är regel vid teckenmaskiner, blir denna bekväma teknik ej gynnsam. D21 är idealisk i sådana sammanhang. Sådana svårigheter är orsak till att Cobol ej rekommenderas för vissa maskiner.

Möjligheten att verkligen kunna använda de moderna avancerade programmeringssystemen och ej behöva avstå därifrån av prestandaskäl torde komma att visa sig vara en av de verkligt stora fördelarna med en datamaskin med allmän inriktning som D21. Det blir alltmera programmeringsarbetet som blir den dominerande svårigheten vid införande av databehandling. Att dras med obekväma programmeringsteknik för att hjälpa maskinen blir alltmera oekonomiskt. Efterhand som mera avancerade programmeringssystem kommer fram blir denna synpunkt allt viktigare.

Packning av data

Möjligheten att packa data så att kompakt lagring ernås har kommit att framstå som ett av den kommersiella databehandlingens särpräglade krav på utrustningen. Detta kommer till uttryck i att de flesta datamaskiner med specialinriktning på kommersiella problem under de senare åren försetts med variabel ordlängd. Det händer t.o.m. ibland att variabel ordlängd uppställs såsom en av de egenskaper man måste kräva av en datamaskin vid kommersiell bearbetning.

Dock borde det vara uppenbart för envar att egenskaper som att vara försedd med variabel ordlängd ej kan ha någon betydelse i sig själv för det företag som skall använda en datamaskin. Här är åter ett exempel på förhastade omdömen inom ADB-området. Vad som åsyftas är snabb och ekonomisk kommunikation mellan datamaskinen och dess magnetbandsenheter. Ett medel, men ej det enda, att nå detta mål är kompakt datalagring på banden. Kompakt lagring kan i sin tur erhållas på olika sätt varav användning av maskin med variabel ord-

tecken fordrar flera positioner i instruktionernas adressdelar. I maskiner med delordsoperationer leder delordsadresseringen till motsvarande pålägg.

Som nämnts omfattar ett tecken i en maskin med "variabel ordlängd" oftast sex binära positioner. Således kan i verkligheten ordlängden ej minskas under sex binära positioner. I en decimal maskin utgör på liknande sätt fyra positioner en undre gräns.

Binära maskiner tillåter bekväm åtkomst till varanda binär position och är i detta avseende de enda maskinerna med fullkomligt variabel ordlängd.

Som exempel på olika lagringsformer kan betraktas en datagrupp "Dato", omfattande termerna År, Mån, Dag vilkas maximala värden är t.ex. 2000, 12, 31.

Vi betraktar motsvarande del av en post där dessa terms värden anges av 1962, 09, 21 och studerar hur detta exempel lagras i olika system, fig. 4. I en teckenmaskin med tecken om sex positioner fordras sammanlagt elva tecken, dvs. 66 binära positioner på bandet och i maskinen, fig. 4 a. En teckenmaskin med tecken om sex positioner och med extra position för gränsmarkering rymmer motsvarande datum på åtta tecken eller 48 binära positioner på band och 56 i minnet, fig. 4 b. I en decimal maskin med 10 siffror per ord och varje ord omfattande 10 tecken krävs 40 positioner såväl på bandet som i maskinen, fig. 4 c. I Saabs D21 räcker 20 positioner på bandet och i minnet för lagring av angivna datum, fig. 4 d. Elva positioner räcker för att lagra talvärden upp till 2 048 i binär form. Fyra positioner rymmer 15 talvärden och fem räcker upp till 32.

Det använda exemplet är något mer än genomsnittligt gynnsamt för D21, som här kräver endast 2,5 binära positioner per decimal siffra. Medelvärde ligger under tre binära positioner per siffra.

För alfabetisk information blir naturligtvis teckenmaskinerna mindre ogynnsamma, men decimala maskiner mera ogynnsamma.

Utläggning av data i packad och eventuellt binärlagrad form klaras automatiskt av avancerade programmeringssystem, sådana som Cobol eller Genius, vilka tillverkas för D21

Det har redan nämnts att även längden av bandblocken har stor betydelse för den erhållna packningsgraden. Detta sammanhänger med att data lagras i begränsade längder på banddatablock som ej får vara större än att ett av dem rymmer i maskinens minne. Mellan blocken måste då finnas ett mellanrum som ej bär information. Detta rymmer t.ex. 600 tecken. Om en post omfattar 200 tecken skulle på en längd av bandet som motsvarar 800 tecken endast 200 tecken vara lagrade. Packningsgraden skulle sålunda redan av detta skäl nedgå till 25 %.

Endast om maskinens minne medger att blockets längd är mycket större än blockmellanrummet erhålles en packningsgrad som närmar sig 100 %. 2 000—6 000 tecken fordras därför.

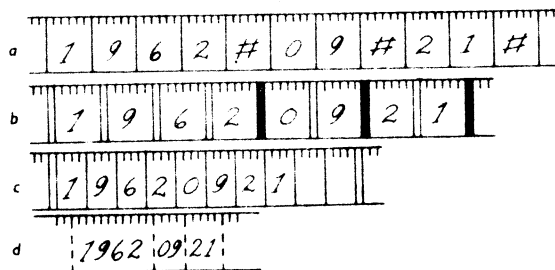


Fig. 4. Utrymmesbehov för lagring av datum i fyra olika typer av maskiner: a teckenmaskin med tecken om sex positioner (gränsmärke), b som a men med extra position för gränsmarkering, c decimal maskin, d Saabs D21.

Hög packning används för att ge hög hastighet på överföringen mellan banden och maskinen. Om maskinen kan skriva och läsa band samtidigt fördubblar detta bandeffekten vid alla registerarbeten där nytt register bildas. D21 kan arbeta på detta sätt. Det kräver två blockutrymmen i minnet för bandläsning och två för skrivning, för samma register, alltså 8 000—24 000 tecken.

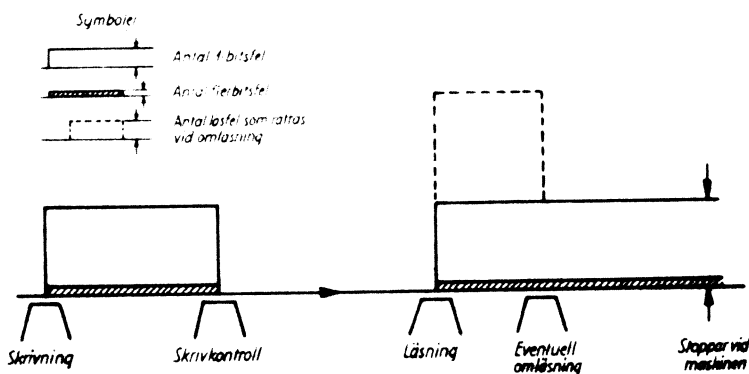
Allt detta gör att en maskin med litet minne och som ej kan skriva och läsa band samtidigt kan ha en verklig bandhastighet som är t.ex. fyra gånger så låg som i D21 även om den nominella hastigheten är densamma. (Som nominell hastighet anges som regel den som svarar mot oändligt långa block.)

Fördelar med stort minne

D21 är så konstruerad att minnet kan varieras i storlek från 4 000 till 65 000 ord. Därmed kan minnet anpassas till behoven. Det kan vara värt att notera att även i fall då minnet ej behöver överstiga en viss storlek med hänsyn till kapacitetskravet så kan det vara god ekonomi att öka minnet för att underlätta programmeringen. Denna är oftast så kostsam att minskat arbete med att pussla med program och data i trånga minnen väl räcker för att betala ett större minne.

En särskild teknik för att utnyttja ett relativt stort kärnminne tillsammans med magnetband i problem som annars skulle kräva slumpvist

Fig. 5. Feleliminering i ett system med skrivkontroll och läskontroll, upptill beteckningar.



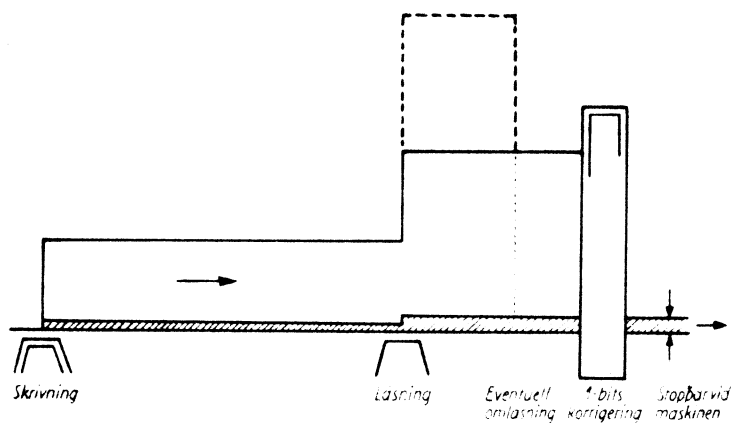


Fig. 6. Feleliminering i ett system med 1-bitskorrektion och flerbiterkontroll. beteckningar som i fig. 5.

arbetande accessminne eller sortering har utvecklats och getts namnet "grupp-access".

Felkontroll

När stora mängder av data skall behandlas automatiskt är det givetvis viktigt att man också ser till att det i erforderlig utsträckning automatiskt kontrolleras att fel ej inkommer eller uppkommer. Problemet felkontroll är komplicerat och kräver en ingående matematisk-statistisk analys. Detta kan naturligtvis ej ske här men några grundproblem skall beröras. Fel kan inkomma med ingångsdata, uppstå vid överföringar mellan maskinen och dess kringorgan samt inuti maskinen under bearbetning.

Den helt dominerande felkällan är ingångsdata, särskilt i de vanliga fall då dessa stansas för hand efter handskrivna listor eller märks av för hand på annat sätt. Detta gäller även under den förutsättningen att stansningen verifieras i särskild operation och omfattande inmatningsredigering vidtagits. Felfrekvensen är här enligt tillgänglig erfarenhet från skilda områden av storleksordningen ett fel på 10 000 inmatade uppgifter. Detta innebär att 1–10 felaktiga data per timme inmatas i maskinen.

En mellanställning intar kringorganen medan den interna maskinen i dag ofta har en till-

förlitlighet som är avsevärt bättre än den i ingångsdata och utgör 1–10 fel per år.

Av dessa fakta kan man inse att bearbetningens tillförlitlighet är helt beroende av att man gör sig det omakett att i systemet lägga in väl genomtänkta kontroller.

Kontrollerna kan vara av typen kontrollsiffror fogade till identifieringsnummer, en effektiv teknik som i alltför stor utsträckning försummas, samt olika slag av kontrollsummor, tvärsummor och avstämningar. Inom bokföringstekniken finns sedan gammalt metoder för felkontroll. Dessutom kan flera av de kontrollprinciper som används vid numeriska beräkningar också tillämpas vid administrativa system.

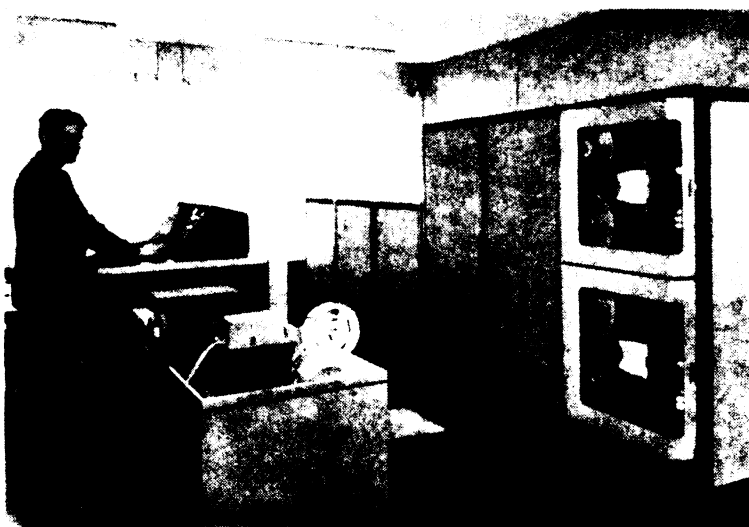
Sådana systemkontroller har den fördelen att de kan konstrueras så att de uppfångar fel begångna såväl av datalämnare som av kringorgan och i maskinen. Däremot kan i maskinen inbyggda kontroller ej upptäcka den majoritet av felen som uppkommer utanför maskinen. Av detta skäl är det olyckligt när system läggs upp där man litar på maskinkontroller och slarvar med systemkontroller. Detta kan vara frestande eftersom de senare ställer stora krav på systemets organisation. Slutligen bör noteras att i varje system med manuellt inslag finns ett stort antal felkällor av sådan karaktär att de ej kan avslöjas av några kontroller. Av hänsyn till bl.a. de uppräknade förhållandena samt de använda transistor-kretsarnas stora tillförlitlighet, har D21's centralenhet konstruerats utan inbyggda kontroller, ett förhållande som gör maskinen enklare och i och för sig är ägnat att minska frekvensen av interna fel.

Magnetbandsystemet, som ur felfrekvenssynpunkt intar en mellanställning, har vid D21 försetts med ett synnerligen omsorgsfullt kontroll- och korrektionssystem.

Fel vid läsningen av magnetband, vilka ger upphov till förlust av information, kan ha sin orsak i fysiska fel vid skrivningen och vid läsningen samt skador som bandet åsamkas av manuell hantering. Vanliga felkällor är elektriska störningar, föroreningar på läshuvuden, töjningar av banden samt skador på magnetbeläggningen. Dessa ger med mycket låg frekvens (ca 1 på 10^5 – 10^6) fel som innebär att en binär position i en tvärrad på bandet blir felaktig (1-bitsfel). Rader som innehåller flera än ett fel är ännu mycket sällsyntare (förekommer vid Sara 10 000–100 000 gånger så sällan som 1-bitsfelen).

Detta har lett till att magnetbandsystemen i allmänhet förses med en kontroll som i princip endast avslöjar 1-bitsfel i en tvärrad på bandet, s.k. paritetskontroll. I verkligheten upptäcker en sådan också flerbiterfel eftersom dessa alltid drar med sig 1-bitsfel i sin omgivning. Ibland kompletteras sådana paritetssystem med en läskontroll efter skrivning vilken då eliminerar fel som uppstår vid skrivning. Den halvpårt av felen vilken uppstår vid läsning av banden eller mellanliggande hantering avhjälpes givetvis inte av skrivkontrollen. Paritetskontroller

Fig. 7. D21-anläggning med två magnetbandsaggregat.



i bandets längdled har mycket liten effekt vid moderna bandsystem med hög pulståthet, eftersom ett fel i en rad alltid drar med sig fel i angränsande rader, på grund av den täta packningen. Litteraturen i detta sammanhang försummar denna effekt varför resultaten blir helt meningslösa.

Vid Sara utvecklades redan 1956—58, som ett av de första i världen, ett system som automatiskt korrigerar alla 1-bitsfel och ger felsignal för flerbitsfel¹. Därmed reduceras frekvensen av störande fel till mindre än en tiotusendel, dvs. de blir praktiskt taget eliminerade. Vid Sara förekommer flerbitsfel ett par gånger om året vid 2-skiftskörning. Dessa leder till felsignal under läsning av bandet och kräver manuella ingripanden. Den stora framgången med korrektionssystemet vid Saras magnetband har motiverat dess införande även vid D21. Korrektionssystem har på senare tid blivit allt vanligare varför Saab här kommit att framstå som föregångare.

Man kan grafiskt åskådliggöra antalet fel som "flyter" med magnetbandet från skrivningen på bandet över läsning från bandet (vid ett senare tillfälle) och fram till inmatningen i maskinen, vid vilket tillfälle fel hindras intränga i maskinen och medför felsignal och slutligen stopp, fig. 5 och 6.

Den streckade strimlan av felflödet avser, som framgår av beteckningarna, fig. 8 upptill, flerbitsfel. Proportionen mellan dessas antal och antalet 1-bitsfel är mycket mindre än fig. 8 visar och flerbitsfelen skulle ej synas i figuren om rätt relation användes. Enär skrivningen i motsats till läsningen kan utföras med stor säkerhetsmarginal genom att skrivströmmen göres mångdubbelt starkare än nödvändigt, blir totala antalet fel vid skrivningen väsentligt mindre än vid läsning. Vid läsning ger i stället möjligheten till omläsning en motsvarande felreducering.

Vid system med paritets- och skrivkontroll elimineras flödet av skrivfel medan läsfelen når fram till inmatningen och där ger stopp (i den mån de ej eliminerats vid omläsningen). Skrivkontrollen eliminerar som syns ca 50 % av alla felen, fig. 5.

I system med 1-bitskorrektion och flerbitskontroll utgör antalet fel före korrigeringen av summan av läs- och skrivfel medan efter korrigeringen endast flerbitsfelen kvarstår, utgörande mindre än 1/1 000 av alla läs- och skrivfelen, fig. 6.

Litteratur

1. Wentzel, V. & Jewertz, B. *Datamaskinerna Saab D2 och Saab D21*. Tekn. T. 91 (1961) s. 997.
2. Widin, K. *Magnetbandsminne för datamaskinen Sara*. Tekn. T. 88 (1958) s. 1175—1179.

63 m³ cisternvagnar för eldningsolja byggs för SJ. De blir fyraxliga och förses med värmeslinga och isolering. Cisterndiametern är 2,6 m och boggi-vagnens egenvikt 24,5 t.

Tekniker skriver bättre! En trött redaktör för en brittisk vetenskaplig tidskrift har sagt sin mening om vetenskapsmäns förmåga att uttrycka sig i vad han tvekar att kalla vetenskaplig litteratur. Han anser sig ha observerat en väsentlig skillnad mellan dagens vetenskapliga uppsatser och klassiska litterära verk.

Vetenskapliga uppsatser är dåligt skrivna. Infinitiver klyvs med glad iver och substantiv används som adjektiv. Meningar är för långa och jargong florerar. Pompöst framförda banaliteter används för att ge auktoritet även åt triviala uppsatser. Sällan bryts regeln att en uppsats skall inledas med någon löjväckande truism. Många vetenskapsmän synes ej erkänna önskvärheten av att göra sig förstådda av omvärlden eller framtiden. Liksom småpojkar, som väntar sig uppskattning trots sina otvättade ansikten, yxar vetenskapsmännen till sina uppsatser i den tron att sanningen skall bli uppenbar även i den minst ordnade ordföljd. Den högsta sanning väntas stråla fram ur den största oklarhet. Därmed förlänas oklarhet sken av en dygd.

Sällan besjålas vetenskapsmän av en önskan att dela med sig av sin visdom till de många. Ofta vänder man sig till en handfull likasinnade (som man ändå träffar tre gånger om året på konferenser). Endast undantagsvis tänker man på alla dem som prenumererat med goda pengar för att få tidskrifterna. Många gånger söker forskaren inmuta sin andel i någon upptäckt genom att publicera vad som synes vara utdrag ur laboratoriejournalen ("Geminus" i New Scientist 1 november 1962 s. 261).

GAH

Maskinexporten i världen. Den totala maskinexporten från världens (utom östblockets) länder fördubblades från 1954, då den var ca 5 400 M\$, till 1961, då den uppgick till 11 700 M\$ (tabell 1). Vid bedömning av exporten från de olika länderna bör även de olika prisstegringarna beaktas. Från 1954 till 1961 uppges maskinpriserna ha stigit med mer än 20 % i Frankrike, Storbritannien, USA och Västtyskland, med 10—20 % i Belgien, Japan, Norge och Sverige, och med mindre än 10 % i Danmark, Italien och Nederländerna.

Om man går efter andelen av västländernas totala maskinexport, behåller USA sin förstaplacering, medan Tyskland gått förbi Storbritannien och nu intar andra plats. Anmärkningsvärt stor ökning av maskinexporten visar Italien, som på sju år gått fram från 140 till 930 M\$ (Nachrichten der Hannover-Messe okt. 1962 s. 120).

WS

Tabell 1. Den västliga världens maskinexport i M\$ samt de olika ländernas andel av hela västvärldens maskinexport 1954 och 1961

	1954		1961	
	Maskin-export M\$	Andel av världsexporten %	Maskin-export M\$	Andel av världsexporten %
USA	2 000	37,1	3 600	30,7
Tyskland	1 020	19,0	2 770	23,7
Storbritannien	1 070	19,8	1 920	16,4
Italien	140	2,6	930	5,6
Frankrike	250	4,6	610	5,2
Schweiz	240	4,4	470	4,0
Sverige	170	3,1	450	3,8
Japan	100	1,9	290	2,5
Övriga västländer	410	7,5	930	8,1
Hela världen (utom östblocket)	5 400	100	11 700	100