

RAMSPECIFIKATION

TankeNyckeln

VÄSTERÅS
1987-05-07

Roland Orre

H J Ä R N A N S H Ä V A R M

INNEHÅLL

1	INLEDNING	2
1.1	Omfattning och syfte	3
1.2	Målgrupp	4
1.3	Uppläggnig	4
1.4	Biografisk bakgrund	5
1.5	Idémässig bakgrund	6
2	DATORN, MÄNNISKANS TJÄNARE	8
2.1	Vem behöver en dator?	9
2.2	Kommer datorn att stjäla jobben?	12
2.3	Programmering av datorer	12
2.4	Datorn som personligt verktyg	13
2.5	Datorn i hemmet	16
2.6	Uppvärmning av ägg	18
2.7	Datorn som hjälpmedel	22
3	RAMSPECIFIKATION AV TankeNyckeln	23
3.1	Målsättning	24
3.2	Motiv och filosofi	25
3.3	Fysiska egenskaper	26
3.4	Fysisk disposition	27
3.5	Presentationsenhet	28
3.6	Inmatningsmetoder översikt.	30
3.7	Inmatningsmetod.	32
3.8	Filsystem och internt minne	33
3.9	Massminnesanslutning	34
3.10	Övrig in/ut-matning	35
3.11	Buss	40
3.12	Kommunikations-protokoll	41
3.13	Centralenhet	45
3.14	Operativsystem	46
3.15	Språk	47
4	VISIONER	60
4.1	Inledning	61
4.2	Teknisk utveckling	61
4.3	Psykosocial utveckling	63
4.4	Transport	63
4.5	Produktion	64
4.6	Datalogins utveckling	64
4.7	Avslutning	65
5	REFERENSER	66

1 INLEDNING

INNEHÅLL

1.1	Omfattning och syfte	3
1.2	Målgrupp	4
1.3	Uppläggning	4
1.4	Biografisk bakgrund	5
1.5	Idémässig bakgrund	6

1 INLEDNING

1.1 Omfattning och syfte

Denna rapport utgör den personliga delen av det teoretiska arbetet i kursen "Konstruktion av automatiserings-system", kurskod 29.1 . Kursen läses under det fjärde året vid civilingenjörsutbildningens data-variant i teknisk fysik vid Teknikum i Uppsala.

Målet med kursen (29.1) är att teoretiskt och praktiskt få chans att angripa några av de problem som uppstår vid konstruktion av datoriserade automatiserings-system.

De kunskaper som skall tillämpas faller i huvudsak inom områdena: Systemteori, Reglerteknik, dator teknik, dator kommunikation samt arbetsmetodik.

Kursen genomfördes i huvudsak som ett grupparbete men skulle även resultera i en personlig rapport i ämnet. Någon koppling mellan grupparbete och rapport var ej nödvändig.

Denna rapport utgör samtidigt underlag för examination i informationsteknik (20.22).

Denna rapport avser att utgöra idémässigt konstruktionsunderlag till ett system vars mål är att lyfta mänskligheten in i en ny fas av sin utveckling mot tänkande individer.

Systemet bygger på två grundidéer:

- En metod för snabb datakommunikation mellan rörliga enheter.
- Att integrera kommunikation, snabb bearbetningsteknik och metoder för smidig interaktion med maskinella hjälpmedel till ett verktyg som många hoppats på, få räknat med men som alla behöver; **TankeNyckeln.**

1.2 Målgrupp

Dokumentet vänder sig till examinatorer i Teknisk Fysik och till alla som är intresserade av datorer och människor i samspel samt de som är intresserade av hur den tekniska utvecklingen skall fortskrida för att vara till mesta möjliga gagn för människans andliga och sociala utveckling.

Om idén med **TankeNyckeln**, som denna rapport bygger på, inte kan förverkligas inom överskådlig tid pga att kortsiktiga intressen och besvärliga regler bromsar det hela, hoppas författaren att detta dokument åtminstone kan ge ett incitament till dem som läser det att fundera på hur vi egentligen vill ha det.

- Hur skall vi gå tillväga för att skapa en bättre människa? En människa som tillvaratar sina egna och jordens resurser på ett optimalt sätt utan risk för krig och andra konflikter.

- Utvecklas tekniken åt rätt håll idag? Kan människan verkligen dra nytta av denna utveckling?

- Vad skulle vi egentligen vilja uppnå med den tekniska utvecklingen?

1.3 Uppläggning

Dokumentet är upplagt enligt följande:

- Innehållsförteckning.
- Inledning.
- Datorn, människans tjänare.
- Ramspecifikation av **TankeNyckeln**.
- Visioner.
- Litteraturreferenser.

Inledning:

Här behandlas rapportens bakgrund och syfte.

Datorn, människans tjänare:

En diskussion om varför vi behöver datorer och hur vi vill arbeta med dem och vad vi förväntar oss.

Ramspecifikation av **TankeNyckeln**:

Denna rapport bygger på en idé som getts arbetsnamnet **TankeNyckeln**. Denna idé skall kunna realiseras i ett verkligt system. För att detta skall vara möjligt krävs riktlinjer från vilka funktion, design och utveckling

skall utgå. I denna specifikation ställs dessa riktlinjer upp till en sådan nivå att designer av programvara och maskinvara har frihet att utnyttja den modernaste tekniken vid detaljspecifikation av **TankeNyckeln**.

Visioner:

Här diskuteras något om möjliga vägar för utvecklingen av symbiosen dator-människa under de närmaste 50 åren. Detta med speciell tonvikt lagd på att idén med **TankeNyckeln** har realiserats.

1.4

Biografisk bakgrund

Jag har alltsedan barnsben varit oerhört intresserad av fysik och teknik. Vid unga år drömde jag om att bli astronom, uppfinnare resp fysiker i nämnd ordning. Jag har alltid känt en stor glädje i att skapa. Vid 11 år byggde jag ett spegelteleskop, vid 13 ett oscilloskop. sedan en mängd förstärkare, radioapparater o dyl under de närmaste åren. Vid 18 byggde jag en synthesizer som specialarbete i gymnasiet.

Mitt val av studieinriktning var helt självklart för mig. Det fanns ingen annan linje där man behandlade så många av de ämnen jag var intresserad av som just inom teknisk fysik. Förmodligen hade jag även något i bakhuvudet som min far sagt om civilingenjörer i mina unga år.

Under min studietid insåg jag att matematiken var en underbar konstform och ett fantastiskt verktyg med vars hjälp det går att beskriva de flesta fenomen i naturen om man utgår från en lämplig modell av detsamma.

Liksom matematiken är ett kraftfullt verktyg att modellera, abstrahera och formulera problem i, så upptäckte jag också att datorn var ett likafullt utmärkt verktyg att organisera och lösa problem med. Jag kom att intressera mig mycket för datorer. Datorn var ett verktyg som fascinerade mig mycket, en tankens skiftnyckel, som, tycktes det, kunde lösa alla problem bara den programmerades för detta.

Datorvetenskapen är dock fortfarande på babystadiet, duktiga datorer är än så länge ofta komplicerade att använda och det är ännu långt ifrån en självklarhet för alla att se datorn som en naturlig förlängning av hjärnan på samma sätt som en skiftnyckel är en naturlig

förlängning av handen. Av denna anledning har jag börjat intressera mig mycket för ergonomi under de senaste åren, speciellt informationsergonomi.

1.5

Idémässig bakgrund

Denna rapport utgör det som saknas sedan januari 1982, då jag tog min sista tentamen för att jag skall kunna ta ut min Civilingenjörsexamen.

Många kanske anser det vara slöhet att inte ta ut sin examen på en gång. Som försvar till detta vill jag säga följande:

- Jag är en person som är både envis och tolerant, jag ger aldrig upp och resignerar.
- Jag har inte dragit några som helst nackdelar av att vara "student" några år extra.
- Jag har kunnat åka billigare på tåg.
- Jag har kunnat köpa böcker billigare.
- Jag har betalat en mycket låg fackföreningsavgift.
- Mitt medlemskap i IEEE har varit reducerat till halva beloppet.

När jag nu skriver detta anser jag det dock vara hög tid att ta ut examen, bla av följande anledningar:

- Jag ser Civilingenjörsexamen som första steget i min karriär.
- Jag vill inte försämra statistiken för utexaminerade.
- Jag betalar nu full avgift till IEEE och fackförbund.

Uppgiften för rapporten något vagt formulerad. Då jag i många avseenden är perfektionist ville jag inte heller slarva ihop någonting som hade som enda syfte att jag skulle få ut min examen. Jag såg istället denna rapport som en utmaning och en möjlighet.

Det första problemet var att bestämma vad rapporten skulle handla om. Nästa problem var tiden. För det mesta kom rapporten att utgöra ett ständigt dåligt samvete som

låg i bakgrunden vad man än sysslade med. Jag kan därför inte rekommendera någon annan att låta uppgifter bli släpande efter.

De främsta orsakerna till fördröjningen har varit:

- Att jag har arbetat för mycket på min arbetsplats.
- Att jag har väntat på den rätta idén.

Jag tänkte först skriva om hur man kan göra sin villa intelligent på allehanda sätt. Jag tyckte att det var ett skojigt projekt, men insåg också att de flesta människor förmodligen inte vill bo på det sätt som jag började skissa på, även om det var intressant ur teknisk synvinkel.

Villaprojektet var egentligen inte så revolutionerande i sig, de flesta av de idéer jag hade var tillämpningar av teknik och metoder som används inom industrin.

En villa byggd på det sätt jag skissade på skulle i dagens läge bli så dyrbar att ingen normal inkomstagare skulle kunna finansiera det.

Grundsyftet med mitt villaprojekt var att jag ville avlasta människan med oväsentligheter för att kunna syssla med det intressanta. Jag har sedermera insett att det var fel väg att gå.

Under sommaren 1986 fick jag äntligen idén, vilken om den genomförs kommer att kunna betyda mer för människans utveckling än vad böcker, tidningar, telefoner och datorer gör idag ; **TankeNyckeln !**

2DATORN, MÄNNISKANS TJÄNARE

INNEHÅLL	
2.1	Vem behöver en dator? 9
2.2	Kommer datorn att stjäla jobben? 12
2.3	Programmering av datorer 12
2.4	Datorn som personligt verktyg 13
2.4.1	Övertro på datorer 13
2.4.2	Utvecklingen av persondatorn 14
2.4.3	Dagens persondatorer 15
2.5	Datorn i hemmet 16
2.6	Uppvärmning av ägg 18
2.6.1	Figur 1, Ägg i vatten 18
2.6.2	Figur 2, Värme flöde genom volymselement 19
2.7	Datorn som hjälpmedel 22

2

DATORN, MÄNNISKANS TJÄNARE

2.1

Vem behöver en dator?

Vem behöver en dator ?

Denna fråga har många ställt och få svarat på. Innan det retoriska svaret kommer, ges här några argument.

Människan är en varelse som utrustats med en hjärna som i volym och abstraktionsförmåga klart överträffar de flesta av planetens innevånare, hur det ligger till med detaljerna här vet man dock ännu inte med säkerhet.

Med tanke på människans fysiska egenskaper skulle vi med stor sannolikhet dött ut mycket snabbt om vi inte haft mer hjärna än en mus eller dylikt. Människans hjärna verkar ha vissa egenskaper som driver på kunskap och utveckling.

En mycket värdefull egenskap hos människan är hennes nyfikenhet, vilken hela tiden strävar att utvidga hennes kunskapsbank.

Människan äger en fantastisk förmåga till abstrakt tänkande. Med hjälp av denna förmåga kan hon bygga modeller av verkligheten och dra slutsatser med mycket liten mängd samtidigt påtagliga fakta tillgängliga. Detta slutsatsdragande brukar ofta benämnas intuition eller sunt förnuft, beroende på situation eller typ av slutsatsdragning.

Människan besitter vidare en förmåga att extrapolera sina modeller och ansätta lösningar som ibland från vissa aspekter strider mot det skunda förnuftet. Denna förmåga benämns fantasi, kreativitet eller omdömeslöshet beroende på sammanhanget och omgivningen.

Dessa egenskaper utgör tre viktiga drivfjädrar för människornas karriär och visioner.

Vi människor har dock en en egenskap som är beklaglig i sammanhanget, det är vårt korta liv. Detta hindrar oss från att hejdlöst odla alla våra intressen. För att fördjupa oss i några intressen måste vi försaka de övriga och specialisera oss. Under en livstid hinner vi helt enkelt inte med så mycket som vi skulle önska.

För att råda bot på vårt korta liv bör vi göra allt som står i vår makt för att förlänga den del av livet där vi kan odla frukterna av våra underbara egenskaper.

En förlängning av livet kan ske på två sätt :

Vi kan förkorta vår arbetstid så att större del av vår tid kan användas till att odla andra intressen än dem vi odlar på vårt arbete.

Alternativt kan vi förskjuta vårt åldrande så vi blir tex 1000 år i stället för 100.

Den senare metoden har under långeliga tider varit en av människans stora drömmar, den om ett evigt liv. Det påstås ofta numera att vi människor är programmerade till en max-ålder på ca 100 år. Om detta är fallet borde det förstas vara möjligt att ändra på det programmet

Det kan ju dock finnas en mening med en sådan programmerad ålderbegränsning. Om vi tänker oss att människan är stadd i ständig utveckling enligt evolutionsteorin skulle utvecklingsnivån alltså vara beroende av antalet generationer. Om vi skulle kunna öka ålderstakets höjd med en faktor tio skulle den hastighet varmed människans utveckling sker förmodligen sjunka till en tiondel under förutsättning att folkmängden är konstant.

En programmerad åldersbegränsning skulle även kunna tolkas så att det är något som vi själva kan ta bort när vi uppnått en viss mognad och då äger förmåga att inse hur programmet skall ändras till önskad livstid.

Det först nämnda alternativet, att förkorta arbetstiden har också sina sidor. Dels finns det en stor grupp människor som i huvudsak tillfredsställer sina intressen under sitt arbete, dels finns en stor skara som inte använder sin fritid till odling av sina andliga intressen utan tillfredsställer andra behov istället. Det senare är en livskvalitetsfråga som inte kommer att behandlas vidare i denna skrift.

Rätt verktyg

Om någon ställs inför problemet att såga av en plank och endast har en nagelfil tillgänglig börjar denne knappast med arbetet direkt. Personen undersöker förmodligen först möjligheterna att skaffa en såg.

De flesta människor som idag är vana att använda ordbehandlare och behöver skriva ett dokument el dyl ser nog till att göra detta på en ordbehandlare istället för att utnyttja en skrivmaskin. Anledningen är här att man

känner större frihet vid ordbehandlaren varvid kreativiteten ökar. Då man på ordbehandlaren arbetar med ett abstrakt papper kan godtyckliga ändringar göras utan större besvär tills man är nöjd. På en skrivmaskin kan man lätt känna tvånget att skriva rätt på en gång vilket kan verka hämmande på själva skapandet.

För allt arbete gäller :

Om man skall erhålla ett gott resultat och känna sig inspirerad av arbetsuppgiften bör man använda lämpligast möjliga verktyg för att kunna arbeta på den för problemet bäst anpassade abstraktionsnivån.

Att använda fel sorts verktyg är ofta detsamma som att slå sönder tid och intresse i onödan.

Människans hjärna är förutom nyfikenhet, abstraktion och fantasi eller kretivitet utrustad med många fler egenskaper som hos vissa individer är mycket utmärkande och hos andra är nästan försumbara.

I regel är hjärnan svag på absolut precision men stark på relativa bedömningar. Tex numeriska beräkningar gentemot snabba överslag eller att jämföra toner eller färger med varandra.

En räknemaskin behöver förmodligen längre tid på sig att konstatera att $1/\pi$ är ungefär $1/3$ än att komma till resultatet 0.318310...

På samma sätt som handen behöver en skiftnyckel för att dra åt en bult eller en hammare för att slå i en spik behöver människans hjärna en hävarm för att ta bort de bromsklotsar som verkar hämmande på nyfikenhet och kreativitet. Ett verktyg som förmår frigöra fantasin och fylla igen kunskapsvacuum.

Frågan :

Vem behöver en dator ?

besvaras förmodligen sanningsenligast med : **ALLA !**

2.2

Kommer datorn att stjäla jobben?

Kommer datorn att stjäla våra jobb ?

Om 100 år är det säkert lätt att tveklöst svara: Nej!

Idag är det lättare att konstatera att tekniska framsteg oundvikligen medför omstrukturering på arbetsmarknaden.

2.3

Programmering av datorer

För att få en dator att göra vad man önskar måste den ges instruktioner om vad den skall göra. Denna instruktionsgivning benämns vanligen programmering.

Programmering kan utföras på en mängd olika vis men gemensamt för de flesta är att någon form av språk representerande en lämplig abstraktionsnivå av maskinen används för att beskriva hur ett problem skall lösas.

De första egentliga datorerna var analogmaskiner. 1930 utvecklades vid MIT en stor analogmaskin som tex kunde beräkna granatbanor.

I en analogmaskin löses problem genom att låta elektriska kretsar få samma egenskaper som den matematiska modellen, vilken skall efterlikna verkligheten.

Programmeringen av analogmaskinen består av att koppla in integratorer, deriveringssteg etc samt ställa in tidskonstanter, förstärkningskoefficienter o dyl. På detta sätt kan mycket svåra analytiska problem bestående av tex kopplade differentialekvationer lösas. Det är däremot betydligt svårare att få en sådan maskin att översätta kinesiska till engelska.

Senare kom de numeriska datorerna där informationen representerades av tal, oftast i 2-systemet varför siffrorna 0 och 1 var tillräckliga. Till en början fick informationen kodas till sådana binära tal, dvs abstraktionsnivån var låg men å andra sidan var friheten istället nästan obegränsad. Detta tack vare att dessa maskiner kunde hänföras till klassen "Touring-maskiner", vilket innebar att de kunde fås att simulera vilken annan maskin som helst.

Genom att utarbeta sk programmeringsspråk som var avpassade till den typ av problem man ville lösa tog datorutvecklingen ordentlig fart.

Det finns idag många tusen olika programmeringsspråk. Många av dessa är dock mycket lika, varför det är bättre att prata om språktyper istället.

Det finns många starkt problemanpassade språk, som tex Cobol, Simula och Lisp, medan andra räknas till de "generella" språken som tex Pascal och PL1. Det visade sig dock tidigt att både Lisp och Simula hade enorma fördelar gentemot många andra språk varför de idag betraktas som varande i högsta grad generella.

En av de första programmeringsspråken för små datorer var BASIC (Beginners All purpose Sequential Instruction Code), vilket var ett litet språk utan stora resurskrav. De små tidiga datorerna hade ofta för litet minne för att klara något annat språk effektivt.

Tyvärr är det ofta så att lösningar baserade på tekniska begränsningar konfirmeras på ett tidigt stadium i en produkts livscykel och senare betraktas som som kutym och antas som standard på området. Sådana standards benämns ofta de facto-standards, dvs beroende av sakernas förhållande.

Exempel på sådana standards är förutom BASIC på programsidan, QWERTY för tangentbord och RS232 för kommunikation.

Det talas ofta om generationer när det gäller programmeringsspråk men detta är ett något diffust begrepp. En högre generation brukar betyda en modernare och mer avancerad produkt men i fallet programmeringsspråk förknippas det snarare med den abstraktionsnivå man tillåts arbeta på mha ~~av~~ språket.

2.4

Datorn som personligt verktyg

2.4.1

Övertro på datorer

Många människor har under 60 och 70-talen grundlagt en övertro på datorer, mycket tack vare de genomgripande administrativa rationaliseringarna som var vanliga men ibland misslyckade pga att systemen ofta hade stora begränsningar och brister som visade sig efter hand.

Det blev vanligt förekommande att datorerna fick skulden för bristerna i de administrativa systemen. Övertron på datorer gick ofta över i misstro.

Från mitten av 70-talet började de sk persondatorerna komma ut på marknaden. De som inte helt låtit misstron mot datorer ta överhanden kunde köpa dessa i tron att de kunde lösa en mängd problem med dem, tex receptlagring.

2.4.2

Utvecklingen av persondatorn

Persondatorerna har gradvis utvecklats från mitten på 1970-talet till dags dato (87) från att vara klumpiga effektkrävande apparater för entusiaster till smäckra bärbara enheter för affärsmän mfl.

Persondatorerna bygger givetvis på mikroprocessorn, denna lilla programmerbara tingest som uppfanns av Intel och Texas nästan parallellt år 1971.

Intels första mikroprocessor, 4004, en liten fyra-bitars dator på ett chip, var den som blev mest känd.

Processorn 4004 har sedan av Intel utvecklats till en rad nya skepnader som 8008 ,8080 (8-bitars), 8086/8088,80286(16-bitars) samt 80386(32-bitars).

4004 hann aldrig börja utnyttjas till mycket annat än räknemaskiner. Motsvarigheter har dock använts långt in på 80-talet i fick-räknedosor, av vilka de mer avancerade idag innehåller ha en 8-bitars processor.

Efter 8008, vilken snabbt avlöstes av 8080 började "persondatorerna" dyka upp som tex IMSAI-8080. Detta vidunder hade tom en frontpanel som de gamla datorerna för att kunna editera programmet samt för att kunna knappa in en "BOOT-loader" dvs det program som skall starta datorn.

8080 fick snart konkurrens från andra företag, ex Motorola kom med 6800, Rockwell med 6502, Cosmac med 1802 samt Zilog med Z80, vilken var en utökning av 8080 genom tex fler adresseringssätt, dubbla register- uppsättningar och fler instruktioner.

Gemensamt för dessa mikroprocessorer är tex adresseringsrymden på 65536 (64 K) byte (8-bitars grupp) och arkitekturen med ett fåtal specialicerade register.

Dessa processorer resulterade i en formlig explosion av sk persondatorer på marknaden, vi har tex Apple med Apple I-III (6502), Commodore med Pet och VIC (6502), Radio Shack med TRS-80(Z80) och Colour Computer (6809). Det fanns givetvis många fler fabriker och processorer.

De flesta av dessa tidiga persondatorer var i många avseenden att betrakta som leksaker och köptes också av många med lek och spel som syfte. Många maskiner köptes av entusiaster som väl kände maskinernas begränsningar och kunde utnyttja dessa maskiner maximalt. Maskinerna slog dåligt på den professionella marknaden.

2.4.3

Dagens persondatorer

Dagens persondatorer vilka flitigt har börjat användas för professionella ändamål är vanligtvis utrustade med processorer hörande till kategorin 16-bitars. Dessa maskiner har fått hyfsade prestanda både vad gäller snabbhet och minnesvolym. De är också ofta utrustade med snabba massminnen varför de kan lagra och behandla stora mängder information.

Till de processorer som är mest vanliga i dagens maskiner hör framförallt MC68000 från Motorola samt 80186/80286 från Intel.

Maskiner med 32-bitars processorer hör ännu vanligtvis till kategorin arbetsstationer. Det finns dock ingen absolut skillnad mellan en persondator och en arbetsstation. Det är snarare en gradskillnad mellan dessa baserad på pris och prestanda. Arbetsstationer brukar ofta vara optimerade för någon speciell uppgift som tex ordbehandling, konstruktion av expertsystem och datorstödd design och konstruktion (CAE/CAD)

De generella 32 bitars processorer som kan förväntas dominera marknaden inom de närmaste åren är Motorolas 68030, Intels 80386 och Nationals 30532. Dessa har alla full 4 GByte (giga byte) adressrymd och inbyggd logik för fullständig hantering av virtuellt minne.

Det finns många fler 32 bitars processorer av vilka vissa, exempelvis den av Inmos utvecklade Transputer, säkerligen kommer att bli mycket populära.

2.5

Datorn i hemmet

Datorer är generella verktyg som kan användas till i stort sett vad som helst, kan man tycka. Hur förhåller det sig med den saken egentligen?

Kan man använda datorer för att städa, laga mat, klä på ungarna, diska och stryka?

För några decennier sedan var det en allmän uppfattning att datorerna skulle kunna klara den typen av uppgifter inom en inte alltför avlägsen framtid.

På senare år har mindre och mindre av dessa visioner diskuterats. En orsak till det kan nog vara att man insett att det faktiskt är svårt att få datorer att göra saker som vi ser som högst vardagliga.

Vissa människor som har följt den tekniska utvecklingen kan ha omvärderat sin syn på datorer i motsatt riktning och tycker att det börjar verka skrämmande realistiskt med intelligenta maskiner som gör våra sysslor.

Andra människor kanske inte anser att det är något fantastiskt med en hushållsrobot längre. Det finns maskinella tillämpningar som för många säkert verkar minst lika avancerade som att sköta lite städning, matlagning och disk.

Vad gäller dagens automatiska spisar etc har de inte blivit speciellt väl mottagna. Enlig leverantörerna används spisarnas automatik inte i någon högre grad.

Varför, är svårt att uttala sig om, det kan ju vara så att automatiken helt enkelt är för dålig. Den kräver i många fall onödigt mycket pysslande och den klarar inte av några svårare uppgifter.

Innan man kan sätta en dator på att tex laga mat, måste kommunikationsproblemet först lösas. En spis el dyl måste kunna kommunicera för att en dator skall kunna styra den. Spisen måste innehålla sensorer för temperaturer, överkok. Alla effekter måste kunna styras från datorn. Detta är ju inga konstiga saker, det görs i stor skala vid styrning av industriella processer. Men sådana funktioner kostar än så länge mycket mer än de konventionella. Utrustning för privat bruk är mycket priskänslig.

Den tekniska utvecklingen pekar dock mot att det blir billigare efter hand att tillverka elektroniskt styrd utrustning än utrustning styrd med mekaniska brytare.

Att hoppas på här är att en smidig billig standard för styrning av elektrisk utrustning växer fram.

En sådan standard bör vara baserad på någon form av buss för att möjliggöra dubbelriktad kommunikation mellan styrenhet och anslutningsenheter av typ kokplattor, ugnar, motorer etc. Exempel på sådan buss är I²C-bussen utvecklad av Philips-Signetics. Denna är en tvåtråds buss med en master. Bussen har tagits fram för användning inom området konsumtionselektronik.

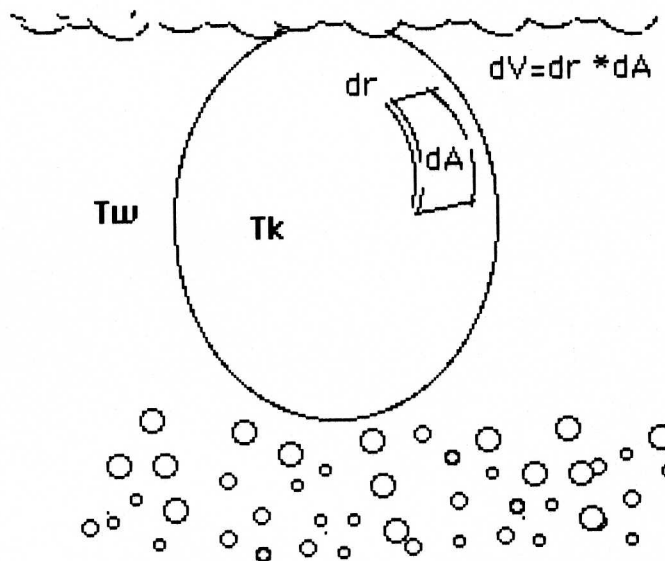
2.6

Uppvärmning av ägg

Som ett exempel på vilka beräkningar som måste göras i ett kök även för de enklaste maträtter ges här ett exempel gällande kokning av ägg.

2.6.1

Figur 1, Ägg i vatten

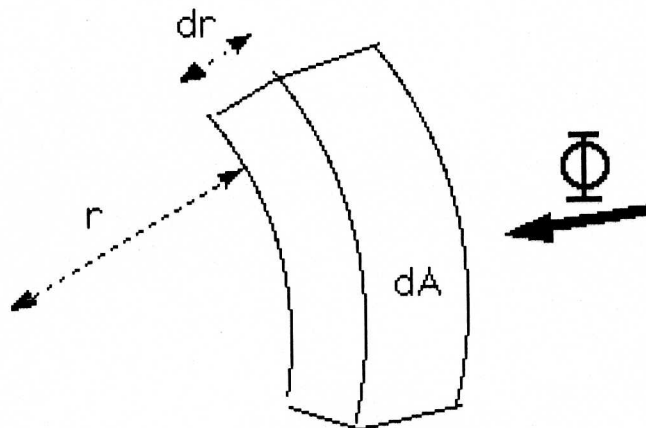


Värmeflödet genom volymselementet dV sättes till Φ .
 Detta flöde går genom ytan dA som har temperaturen T .

Flödet genom denna yta: $\Phi(r) = \lambda dA \text{ grad } T$

2.6.2

Figur 2, Värme flöde genom volymselement



(Anm: grad är förkortning av Gradienten vilken är en vektorvärd funktion som anger ett flödes storlek och riktning i varje punkt där de partiella derivatorna av flödet map de ingående koordinaterna är definierade. För ortonormerat system ges gradienten av f som

$$\text{grad } f = \nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right)$$

Denna operator betecknas ofta med symbolen ∇ vilken brukar utläsas "nabla" .)

Genom ytan dA' (den inre ytan motsv dA) strömmar flödet:

$$\Phi(r') = \lambda dA' \text{ grad } T'$$

Skillnaden mellan dessa flöden värmer upp volymen dV och ges av:

$$\Phi(r) - \Phi(r') = \lambda dA dr (\text{grad } T - \text{grad } T')$$

Men $\text{grad } T - \text{grad } T'$ på sträckan dr är ju detsamma som divergensen för $\text{grad } T$, vilken även betecknas som ΔT . (Laplace T)

Vi får differentialekvationen: $d\Phi = \lambda dV \Delta T$

Uppvärmningen per tidsenhet (dT/dt) av massan m förhåller sig till materialets värmekapacitivet (C) som :

$$dT/dt = d\Phi/(dm C)$$

(massan dm är densitet ρ volymen, vi kallar densitet för ρ och volym för dV)

$$\text{mao : } d\Phi = \rho C dV dT/dt$$

Nu erhålles ekvationen för värmeledningen i ägget enl:

$$\rho C dV dT/dt = \lambda dV \Delta T$$

eller

$$\Delta T = (\rho C)/\lambda dT/dt$$

Vår approximation av ägget till en sfär ger av symmetriskäl att;
 $T = T(r,t)$ dvs temperaturen beror endast av r och t , för enkelhets skull förutsättes även att ägget är homogent och har likformig värmeledningsförmåga och värmekapacitivitet.

För en sfär är polära koordinater att föredra.

$$\Delta T \text{ kan då skrivas } \frac{1}{r} \frac{d^2}{dr^2} (rT)$$

$$\text{Ekvation : } \Delta T = (\rho C)/\lambda dT/dt$$

$$\text{Begynnelsevillkor: } T(r,0) = T_k \text{ (kylskåpskallt i ägget)}$$

$$\begin{aligned} \text{Randvillkor: } T(a,t) &= T_w \text{ (vattnets temperatur vid skalet)} \\ T(0,t) &= T_w \text{ då tiden går mot oändligheten} \end{aligned}$$

Förenklingar: Vattnets temperatur T_w är konstant, uppvärmningen har gått snabbt eller äggen har lagts i kokande vatten.

Då kan vi substituera T enl: $u(r,t) = T(r,t) - T_w$
vilket ger

$$\text{Begynnelsevillkor: } u(r,0) = T_k - T_w$$

$$\begin{aligned} \text{Randvillkor: } u(a,t) &= 0 \\ u(0,t) &= 0 \text{ då tiden går mot oändligheten} \end{aligned}$$

En ny substituering där: $v(r,t) = r u(r,t)$
ger

$$\text{Begynnelsevillkor: } v(r,0) = r (T_k - T_w)$$

Randvillkor: $v(a, t) = 0$
 $v(0, t) = 0$ Naturligt randvillkor ($0 \cdot u = 0$)

Denna ekvation kan lösas mha variabelseparation:

$$v(r, t) = R(r) Q(t)$$

$$\Rightarrow R(0) = R(a) = 0$$

$$R''(r) Q(t) = (\rho C) / \lambda R(r) Q'(t) \quad (\text{ekvationen})$$

$$R''(r) / R(r) = (\rho C) / \lambda Q'(t) / Q(t) = -\tau \quad (\text{en konstant})$$

$$R''(r) + \tau R(r) = 0$$

$$Q'(t) + (\tau \lambda) / (\rho C) Q(t) = 0$$

Dessa är egenvärdesproblem.

Det första ger att $\tau = n^2$; n tillhör de positiva heltalen.

Följande lösningar erhålles:

$$R(r) = \sin(n \pi r/a) \quad (\pi = 3.14159265\dots)$$

$$Q(t) = b(n) \exp\left(\frac{-(n^2 \pi^2 \lambda)}{a^2 \rho C} t\right)$$

(exp är exponentialfunktionen med basen $e = 2.7182818284\dots$)

Begynnelsevillkoret $v(r, 0) = r (T_k - T_w)$ ger:

$$b(n) = 2/a \int_0^a r (T_k - T_w) \sin(n \pi r/a) dr$$

$$= (-1)^{(n+1)} 2 a \frac{(T_k - T_w)}{(n \pi)}$$

Då $u(r, t) = 1/r v(r, t)$ och $T(r, t) = u(r, t) + T_w$

erhålles lösningen:

$$T(r, t) = \frac{T_w + 2a(T_k - T_w)}{(n \pi r)} \sum_{n=1}^{\infty} \frac{-1(n+1)}{n} \sin\left(\frac{n \pi r}{a}\right) \exp\left(\frac{-(n^2 \pi^2 \lambda)}{(a^2 \rho C)} t\right)$$

(Anm: $n=1.. \infty$ står för intervallet av hela tal mellan 1 och oändligheten)

I detta fall var endast temperaturen intressant vid $r=0$ (centrum).

$$\lim_{r \rightarrow 0} \sin\left(\frac{n \pi r/a}{n r}\right) \rightarrow \pi/a \Rightarrow$$

$$T(t) = T_{w+2} (T_k - T_w) \sum_{n=1}^{\infty} -1^{n(n+1)} \exp\left(\frac{-(n^2 \pi^2 \lambda)}{(a^2 \rho C)} t\right)$$

2.7

Datorn som hjälpmedel

Det är förmodligen inte många människor som drömmer om att göra avancerade serieutvecklingar på sitt frukostägg. De flesta av oss är inte alls så snabba och säkra på den typen av arbete att vi skulle hinna räkna klart innan ägget blivit stenhårt. Det skulle också krävas en enorm entusiasm för att göra det varje morgon, alternativt skulle vi så småningom sluta äta ägg. Vi människor tröttnar lätt på stereotypt arbete.

Datorer är annorlunda, stereotypt arbete eller icke. Det spelar ingen roll, en dator upprepar vilket arbete som helst godtyckligt många gånger utan att tröttna.

Datorer har gett designers, konstnärer, konstruktörer, organisatörer mfl helt nya frihetsgrader i sina arbeten. På sikt kan datorerna i positiv riktning påverka hela mänsklighetens sätt att vara och att tänka.

3RAMSPECIFIKATION AV TankeNyckeln

INNEHÅLL	
3.1	Målsättning 24
3.2	Motiv och filosofi 25
3.3	Fysiska egenskaper 26
3.4	Fysisk disposition 27
3.4.1	Figur 3, Möjligt utseende på TankeNyckeln 28
3.5	Presentationsenhet 28
3.6	Inmatningsmetoder översikt. 30
3.7	Inmatningsmetod. 32
3.8	Filsystem och internt minne 33
3.9	Massminnesanslutning 34
3.10	Övrig in/ut-matning 35
3.10.1	Anslutning för digitaliseringspenna 35
3.10.2	Ingång för ljud 35
3.10.3	Utgång för ljud 36
3.10.4	Ljussensor 37
3.10.5	Kameraingång 37
3.10.6	Radiotransciever 39
3.10.7	IEEE-488 (GPIB) 39
3.10.8	Andra snitt 40
3.11	Buss 40
3.12	Kommunikations-protokoll 41
3.12.1	Målsättning 41
3.12.2	Grundkrav 41
3.12.3	Ansats länknivå-protokoll 42
3.12.4	Positionsfält 43
3.12.5	Ytidentitet 43
3.12.6	Nodidentitet 44
3.13	Centralenhet 45
3.14	Operativsystem 46
3.15	Språk 47
3.15.1	BASIC 47
3.15.2	LISP 48
3.15.3	PROLOG 50
3.15.4	LOGO 51
3.15.5	MODULA-2 och PASCAL 52
3.15.6	SIMULA 53
3.15.7	ADA 54
3.15.8	FORTH 55
3.15.9	APL 55
3.15.10	C 56
3.15.11	OCCAM 57
3.15.12	Andra språk 58
3.15.13	Vilket språk? 58

3

RAMSPECIFIKATION AV TankeNyckeln

3.1

Målsättning

Föreliggande specifikation utgör en ramspecifikation för ett personligt verktyg baserat på de senaste landvinningarna inom elektronisk komponentteknik och datorvetenskap.

Denna specifikation har också som målsättning att stadfästa ett förslag till en standard inom området datornät-kommunikation där nätet utgörs av ett stort antal fria rörliga noder och ett litet antal fasta noder.

- Verktöget skall höja den enskilde människans produktivitet, kreativitet och abstraktionsförmåga.
- Arbetsnamnet på detta verktyg är **TankeNyckeln**.

Motivet till namnet är idén att hjärnan behöver en hävarm för att kunna frigöra sig från sådana faktorer som verkar hämmande på dess abstraktionsförmåga, tex stereotypa handlingar av algoritmkaraktär. Verktöget av hävarmstyp benämns ibland nycklar, ex skiftnycklar, momentnycklar.

- **TankeNyckeln** skall genom avancerad radiokommunikationsteknik kunna kommunicera med andra TankeNycklar resp med alla allmänna databaser i världen.
- **TankeNyckeln** skall vara en nyckel till all världens kunskap genom att öppna alla tillgängliga kunskapsdatabaser.

Vad gäller förslaget till kommunikationsstandard kan det förmodligen av globalpolitiska och tekniska skäl inte anses möjligt att förverkliga helt inom den närmaste tiden. En delmängd av denna standard skall ändå byggas in i den första versionen av **TankeNyckeln** för att på ett effektivt sätt möjliggöra kommunikation med centrala databaser och andra TankeNycklar. Kommunikationsmetoden utgör en mycket väsentlig del av hela filosofin bakom **TankeNyckeln**.

TankeNyckelns fysiska utformning kan mycket väl ändras oerhört alltefter de förutsättningar som den tekniska utvecklingen ger. Om 20-30 år kan den kanske opereras in under huden och ta sin energi direkt från kroppen och in/ut-matning skötas medelst synaptiska nervförbindelser, men idén med kommunikationen består!

3.2

Motiv och filosofi

Genom att dra fördel av nya energisnåla, högpresterande elektroniska kretsar och avancerad kommunikationsteknik kan beprövade hjälpmedel som utvecklats inom områdena CAD/CAM och AI, vilka till dags dato endast varit möjliga att utnyttja på förhållandevis dyra arbetsstationer, spridas till en rad nya användargrupper. På lång sikt till större delen av mänskligheten.

Hjälpmiddel av denna kategori är ofta både användarvänliga och effektiva då de baserar sig på forskning inom informationsergonomi och artificiell intelligens. De goda frukterna av avancerad datorteknik bör därför kunna spridas, mottas och accepteras på ett sätt som ej tidigare varit möjligt.

Datorer är fantastiska maskiner, rätt använda kan de lyfta människan in i en ny fas av sin utveckling.

Dagens datorer är visserligen duktiga och kan göra saker som få människor kunde drömma om för några decennier sedan. Men samtidigt finns många drömmar som inte gått i uppfyllelse. Datorer vet inte allt. De är inte mänskliga och det går inte att föra speciellt meningsfulla diskussioner med dem.

Det finns forskare inom artificiell intelligens som ifrågasätter om vi någonsin kommer att få "intelligenta" datorer. Många menar med intelligens att maskinerna då måste ha ett medvetande, ett ego, andra menar att en intelligent dator kan tolkas som en maskin vilken intellektuellt skulle kunna simulera en människa utan att bli avslöjad. (Tourings test).

Denna specifikation gör inga som helst anspråk på att försöka definiera begreppen intelligens och medvetande.

Filosofin är snarare att med den teknik som står till buds idag definiera ett system som rätt utnyttjat ger varje människa ett personligt orakel som alltid finns till hands och som låter människor föra diskussioner mellan varandra och med maskiner på ett sätt som är oberoende av om vi lyckas göra intelligenta maskiner en gång. Givetvis skall systemet tillåtas bli "intelligentare" med åren allt efter som utvecklingen skrider framåt.

Många som läser detta kanske retar sig på begreppet "artificiell intelligens", så missbrukat som det blivit. Den mesta av forskningen som kallas AI har rört vad vi i

dag kallar "Kunskapsbaserade system" (KBS) eller "expert-system". För att undvika bryderi kommer begreppet AI inte att nämnas mer i detta dokument.

Vi får inte glömma att "användarvänlighet" inte är ett entydigt begrepp, lika lite som "intelligens" är det.

Vad som upplevs som användarvänligt av en användare behöver inte nödvändigtvis upplevas så av andra.

En person kan uppleva ett system som går fort att lära sig med få enkla kommandon som användarvänligt.

Samma system kan av en annan person upplevas som en tvångströja och orsaka total frustration varje gång användaren kör huvudet i väggen på grund av systemets osmidighet och inbyggda begränsningar.

Det går inte att göra ett perfekt system som passar alla! Därför måste systemet vara så flexibelt att det går lätt att bygga ut och ändra för att passa varje individuell användare som en handske.

Ett flexibelt system får inte fungera som en bromskloss utan tillåta att individen blir duktigare efter hand.

Avsikten med **TankeNyckeln** är att varje människa skall ges möjlighet att utveckla sig under hela sin livstid och att mänskligheten skall kunna utvecklas hand i hand med den enskilda människan.

Människor skall anse det vara självklart att använda en dator vid lösandet av mycket abstrakta problem, lika väl som de idag tar en skruvmejsel till hjälp för att vrida en skruv eller utnyttjar räknedosor alternativt kalkylprogram för att lösa aritmetiska problem.

3.3

Fysiska egenskaper

Kravet på det fysiska utförandet grundar sig på den teknik som (nästan) finns tillgänglig idag.

Följande utgör fysiska grundkrav :

- **TankeNyckeln** skall vara fullkomligt portabel, i bemärkelsen användbar överallt oberoende av nätström.
- Den skall gå att använda av en stående person genom att den med god balans skall gå att hålla i ena handen.

- Den skall ha en överskådlig presentationsenhet med hög detaljrikedom.
- Det skall gå att använda **TankeNyckeln** kontinuerligt under så lång tid att intervallen mellan batteribyten eller laddning inte upplevs som irriterande korta. Kortast acceptabla intervall ca ett dygn. Motivering : Människor kan ibland arbeta upp till ett dygn i sträck men sällan längre, varför laddning kan ske under sömnen.

Med hänsyn taget till ovanstående krav samt till dagens tekniknivå respektive teknikacceptans och standarder inom områdena strömförsörjning och dokumentationsteknik betyder detta en storlek på ca 310x220x20 mm.

Detta mått är baserat bla på det europeiska dokumentationsformatet A4 (210x296 mm) respektive de elektriska element som finns under benämningen "Size AA", LR6 eller MN1500. Hänsyn bör tas till även andra världsdelaers standardiserade dokumentationsformat.

Presentationsenhetens storlek bör vara runt 214x300 mm. Då kan fullt A4-format presenteras tillsammans med några streck och pilar som kan ligga utanför "papperet".

Några ytterligare millimeter åtgår förmodligen till anslutningskontakter för presentationsenheten. Höljet bör utföras i en slagttålig plast som tex Lexan. Lådan bör dessutom vara skärmad för att låta påverkan från elektriska och magnetiska fält bli så liten som möjligt. Då får antennen också ett jordplan.

För att skydda presentationsenheten då den inte används bör finnas ett skyddande uppfällbart/avtagbart lock.

- Locket skall vara överdraget med solcell av amorft kisel för laddning/drift av verktyget där belysningen så tillåter. Sådana kiselceller kommer sannolikt att finnas tillgängliga med 13 % effektivitet 1989 enl U.S. Department of Energy's Solar Energy Research Institute.

3.4

Fysisk disposition

I huvudsak följande skall få plats i lådan :

256 - 512 minneskapslar av typ "flat pack" fördelade på ett eller två kort. Målsättningen här är att i första versionen få in 16-64 Mbyte statiskt minne.

Motsvarigheten till ca 32 st AA-celler (laddningsbara).

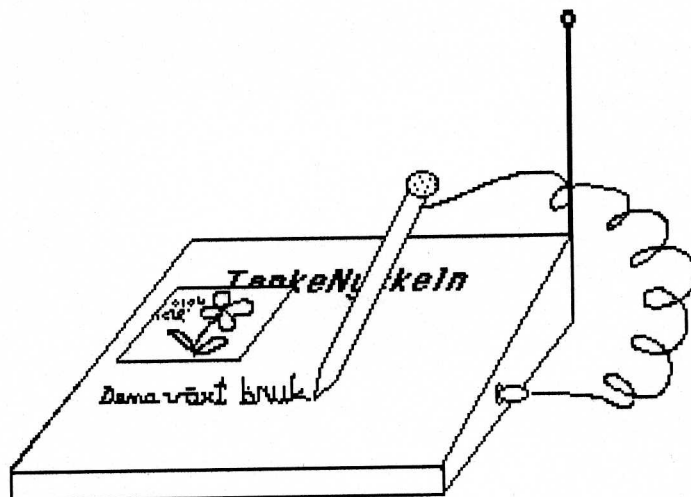
Logik bestående av CPU, klock-logik, in-ut kretsar, drivkretsar för presentationsenhet, adressavkodning, talsyntes och röstigenkänning samt radiodel bestående av modulator-demodulator och en transceiver för ett fn okänt frekvensområde (förmodligen 0.5 - 2 GHz).

Presentationsenhet utgjord av en LCD-skärm hoppyggt med en digitaliseringsenhet "digitizer". Storlek ungefär som A4-format ex 215x300 mm .

För att erhålla så låg kostnad som möjligt för de första prototyperna resp de första serierna användes enbart komponenter ur accepterad industristandard. På lång sikt måste man givetvis vara öppen för att det utvecklas speciella kretsar för att reducera storleken, tyngden och strömförbrukningen. Vidare kan helt ny teknik komma ifråga för presentation och inmatning på sikt.

3.4.1

Figur 3, Möjligt utseende på TankeNyckeln



3.5

Presentationsenhet

Presentationsenhet är den del av TankeNyckeln som kräver specifik forskning och utveckling för att kunna tillverkas. Med detta sagt att någon lämplig presentationsenhet inte finns vid tiden för denna specifikations framtagande.

- Upplösningen på skärmen bör vara så god att det går att presentera text och grafik av god kvalitet. En riktlinje bör vara punkter som är mindre än 0.1 mm. Till att börja med får kravet på upplösning hållas ner till ca 0.2 mm.
- Skärmen storlek skall vara som A4 ca 210x300 mm, vilket till en början innebär 1000x1500 bildpunkter för att när tekniken förfinats öka till ca 2000x3000 bildpunkter.
- Skärmen är integrerad med ett digitaliseringsbord, dvs en enhet som med hög hastighet och precision skall sampla en pennas absoluta position på skärmen, Denna enhet (en sk "digitizer") används istället för den nu vanligt förekommande kombinationen mus och tangentbord.

För att erhålla en rimlig strömförbrukning är den enda tänkbara skärmen idag av typ flytande kristall (LCD).

Kraven på presentationsenheten är höga. De flesta LCD-skärmar som hittills sett dagens ljus lider av mycket dåliga kontrastegenskaper, samtidigt som upplösningen inte brukar vara den bästa. Orsaken till att skärmarna inte har kontrast motsvarande den som återfinns på tex räknedosor och klockor är att de stora skärmarna hittills varit multiplexade, dvs varje punkt på skärmen är endast spänningssatt under en liten del av tiden. Detta för att komma ner till ett mindre orimligt antal tilledere. I en armbandsklocka eller ett mätinstrument är varje element kopplat till en egen kontakt vilket knappast har varit tänkbart för tex skärmar på 640x400 punkter. För att erhålla en bättre kontrast ökas spänningen i varje impuls men LCD är känsliga varför man snabbt kommer upp i gränsen för vad kristallerna tål.

Vid en jämförelse av tex lysdiodmatriser, vilka alltid multiplexas, med LCD-skärmar kan man säga att en lysdiod är mindre känslig för toppvärdet på inmatad effekt, den bryr sig mer om tidsmedelvärdet av effekten. Kristallen i en LCD följer helt andra principer. Här är det fråga om en polarisering av molekyler under påläggande av ett elektriskt fält. Det finns inget direkt samband mellan intensitet och pålagd effekt. Strömförbrukningen åtgår till största delen för att vrida molekylerna. Detta tar dock en viss tid och hinner man inte lägga på fältet under tillräckligt lång tid hinner inte heller molekylen vrida sig helt. När sedan fältet släpper kommer molekylen att vrida sig tillbaka, vilket också tar en viss tid.

Hittillsvarande LCD-skärmar av större format är resultat av en besvärlig kompromiss mellan många egenskaper.

Den kompromisslösa LCD-skärmen går dock att göra, i denna läggs informationen ut punkt för punkt och behöver endast läggas ut på nytt när den skall ändras. Denna typ av skärm har sk bistabila kristaller. Det har talats om passiva bistabila kristaller under några år, men det är i dagens läge osäkert när dessa blir användbara.

Det finns dock en typ av LCD-skärm under utveckling där varje punkt innehåller en transistor och en kondensator, alltså samma teknik som i dynamiska minnen. Man utgår alltså från ett tunt amorft kiselsubstrat av samma storlek som skärmen och tillverkar en slags minneskapsel med mycket låg densitet, där varje minnescell svarar för att hålla en liten yta polariserad. Denna typ av skärm är inte bistabil utan måste uppdateras med viss cyklicitet. Logiken för att hålla informationen bör kunna byggas in i skärmen på samma sätt som i vissa minnen, de sk semi-statiska, vilka är dynamiska men beter sig som statiska.

3.6

Inmatningsmetoder översikt.

Då verktyget **TankeNyckeln** skall vara mycket lätt att använda bör dess normala inmatningsmetod vara snabb, lätt att lära samt mycket flexibel. Den bör inte heller ta stor plats i anspråk.

Alfanumeriska tangentbord har länge varit den gängse inmatningsmetoden för datorer. Dessa är dock olämpliga då bra tangentbord är otympliga, de kräver bägge händerna samt är bara lämpade för text och siffror.

För att peka i menyer och för hantering av grafik har mus (eng mouse) länge varit ett vanligt hjälpmedel. Musen är lätt att hantera och har därför blivit populär, principen för musen är att relativa rörelser av denna mot en yta översätts till markörförflyttningar. En mus kan även utgöras av givaren till ett digitaliseringsbord, den överför i så fall absoluta koordinater och dess placering är analog med markörens position på skärmen.

Den relativa musen brukar bestå av en boll som rullar mot bordet på vilken hjul för x- resp y-led känner rörelsen. Det finns även sk optisk mus som med fotoceller känner hur många linjer i x- resp y-led som passeras. En optisk mus kräver alltså en speciell yta men på senare tid har en variant som inte kräver någon speciell yta tagits fram. En sådan mus bygger på principen att en slät yta

alltid har vissa ojämnheter. När musen rörs över ytan uppstår brus från de optiska givarna som kan korreleras för x- resp y-led varigenom förflyttningen registreras.

Den absoluta musen som även kan vara utförd som en penna använder vanligtvis kapacitiv eller induktiv princip. I den kapacitiva består givaren av ena delen av en kondensator, den andra delen av kondensatorn utgörs av ett stort antal smala ledande remsor efter de två koordinataxlarna. Genom att omväxlande ladda upp och sedan ladda ur den kondensator som består av summan av de kondensatorer som utgörs av givarplattan och remsorna och sedan mäta fasläget på den resulterande plattpotentialen kan positionen bestämmas med stor precision, typiskt 25 mikrometer. Den induktiva fungerar nästan på samma sätt men här har remsorna ersatts av spolar, även givaren (musen) innehåller en spole. Precisionen hos den induktiva är inte riktigt lika stor som hos den kapacitiva, typiskt ca 0.1 mm upplösning och en noggrannhet på ca 0.5 mm. Den är dock billigare att tillverka och är den vanligaste.

Det lär också ha gjorts en tröghetsmus som mha accelerometrar känner hastighetsförändringar vilka sedan kan integreras för att ge hastighet eller position. Någon information om denna har dock inte gått att få tag på.

Musen har dock följande nackdelar:

- Den tar extra plats, det krävs en ledig yta med lämplig friktion för rullbollsmusen resp en specialyta för de enklare optiska. För avancerade optiska resp tröghetsmus krävs ingen specialyta.
- Musen är endast lämpad för grafisk inmatning, om text skall kunna matas in krävs dessutom ett tangentbord.
- Musen är ej bra för frihandsritning, detta pga att det är våra ganska grova armmuskler som används för förflyttningen. Vår finmekaniska precision sitter i fingrarna vilka ej används vid arbete med mus.

Det finns ett trevligt alternativ till mus som består av ett "headset" som liknar ett par hörlurar men innehåller tre ultraljudsmikrofoner. En ultraljudsändare placeras i närheten av presentationsenheten. Huvudets position blir med denna metod analog med markörens. Metoden lämpar sig dock bäst för pekning. Ett bra handikaphjälpmedel!

Det finns även mätmetoder för att känna ögats rörelser med hög precision. Denna princip förväntas komma till användning hos tex stridsflygare som behöver ha händerna

lediga. Men även för fredligt bruk tex för synskadade, en sådan givare kan styra tex en liten CCD-kamera vilken sedan kan användas för att generera en "bild" på hjärnans yta el dyl. Metoden är intressant och kan komma till användning i **TankeNyckeln** när någon effektiv metod för inmatning av "text" på motsvarande sätt tagits fram.

3.7

Inmatningsmetod.

- För att komma runt problem som tangentbords otymplighet och musens beroende av en yta samt dess begränsningar ifråga om frihandsritning skall principen med digitaliseringsbord användas i **TankeNyckeln**. Som givare användes en penna.
- Digitaliseringsbordet integreras med presentationsenheten, dvs LCD-skärmen.
- Noggrannheten på digitaliseringsenheten bör vara ca 0.1 mm, den får i vilket fall inte vara sämre än upplösningen på den använda LCD-skärmen.

Genom att använda en penna som inmatningsmedel uppnås ett flertal fördelar :

- Tar ingen extra plats. Utrymmet för presentationsenhet behövs ändå, åtminstone till presentationstekniken har förbättrats.
- Handens finmotorik i fingrarna utnyttjas till fullo.
- En penna är mycket lämplig för frihandsritning.
- Vid inmatning av text kan valfria tecken användas. Dessa är ej språkberoende, stenografi kan med fördel användas.
- Det går fortare än kombinationen mus, tangentbord. En driven maskinskrivningskunnig kan visserligen mata in ren text något fortare med ett tangentbord men hantering av text kräver också en mängd ingrepp som fontbyten, storleksändringar, utval av områden, positionering etc. Sådant tar tid pga att handen då måste röras fram och tillbaka tex mellan mus och tangentbord eller till funktionstangenter. Detta talar entydigt till pennans fördel.

- En mycket kort upplärningstid för att lära sig mata in information. Denna tid utgörs till största delen av **TankeNyckeln**s adaptation till sin användare. **TankeNyckeln** är ett personligt verktyg och skall därför läras att förstå sin användares handstil om denne inte vill använda någon standardiserad handstil ex ANSI X3.45 "Character set for Hand-Printing" som **TankeNyckeln** förstår initialt.

3.8

Filsystem och internt minne

Filsystemet är i första hand primärminnesbaserat för att vara snabbt och robust. Av denna anledning skall primärminnet vara så väl tilltaget att det ledigt räcker för några dagars arbete.

Målsättningen är att få plats med 256 eller 512 minneskapslar av typ LCC (plastic chip carrier) eller typ "flat pack". Med användande av de 128 Kbyte CMOS statiska minneskapslar som räknas komma ut på marknaden i slutet av 1987 el början av 1988 kan en minnesvolym på 64 Mbyte erhållas. Dagens ekonomiska alternativ som är 32 Kbyte, ex Nec μ PD43256C ger 16 Mbyte.

Primärminnet uppdelas efter behov mellan filsystem och primärminne. Eventuellt kan en tvådimensionell arkitektur typ Multics väljas. I sådant fall skiljes ej på minne och filsystem. Denna arkitektur är den bästa från många synpunkter men får bero på den valda centralenhetens arkitektur.

Allt minne i **TankeNyckeln** är kontinuerligt strömförsörjt. För att försäkra sig om att inget data tappas bort skall flera säkerhetssystem för detta finnas. Dels skall en del av strömförsörjningen enbart försörja minnena, dels skall ström kunna tas från andra enheter om minnenas egen försörjning falnar. Som sista säkerhet skall finnas en reserv av tex Lithium-typ som skall räcka för några dagars eller veckors minnesuppbäckning. Den normala strömförsörjningen skall klara några månader utan uppladdning för minnesuppbäckningen.

Det är med dagens teknik nödvändigt att använda CMOS statiska minneskapslar för att få tillräckligt låg strömförbrukning. CMOS dynamiska skulle vara ett alternativ men de drar tyvärr mer ström, de drar ca 10 ggr mer i väntelägesförbrukning (standby), men informationen i dynamiska

minnen måste uppdateras ("refresh") med jämna mellanrum, under "refresh" tar minnet lika mycket ström som under drift.

De statiska minnena kan dessutom sättas i något som kallas "data retention mode" genom att dra ner matningsspänningen till 3 volt. De upprätthåller då informationen enbart och är inte tillgängliga men strömförbrukningen blir mycket låg. Som exempel drar Toshiba TC55257 (32 Kbyte CMOS) max 50 μ A i "data retention mode".

Filsystem skall även kunna residera på yttre massminnen anslutna via härför avsedd kontakt.

3.9

Massminnesanslutning

Anslutning av massminne sker via en SCSI-port, ANSI X3T9.2 (Small Computer Systems Interface). SCSI-protokollet är specificerat för hastigheter upp till 4 MByte/s.

Eftersom SCSI är ett generellt gränssnitt behövs även ett mjukvaruprotokoll för att hantera direkt accessbara enheter som tex magnetiska och optiska skivminnen. Det protokoll som tidigare gällt som ad hoc standard har sedermera standardiserats och kallas "Common Commands Set" (CCS). CCS definieras som ett addendum till SCSI.

SCSI-portens fysiska utformning är en 25-polig honkontakt av D-typ. Detta är samma typ av kontakt som används för låghastighetsprotokollet RS232, varför denna möjligen skulle kunna kombineras med detta.

SCSI kan hantera många typer av enheter ex skrivare. Protokollet klarar upp till 8 enheter vilka samtidigt kan kommunicera med upp till 8 processer via DMA (Direct Memory Access), den totala bandbredden kan givetvis inte överstiga 4 MByte/s.

Strömmen till SCSI-kontrollkretsen skall slås av automatiskt för att spara ström då SCSI-porten inte används för kommunikation.

För hantering av SCSI finns ett antal kretsar på marknaden, den som väljs bör klara det fullständiga protokollet med full bandbredd, vilket alla kretsar ännu inte gör. Strömförbrukningen hos den valda kretsen bör även vara låg och antalet kringkretsar skall vara så få som möjligt.

Exempel på en strömsnål krets (CMOS-teknik) som har implementerat SCSI-protokollet fullständigt enligt specifikationen är MB87030 från Fujitsu.

3.10

Övrig in/ut-matning

3.10.1

Anslutning för digitaliseringspenna

Under överskådlig tid utgör digitaliserings-pennan det huvusakliga verktyget för att kommunicera med **TankeNyckeln**. För anslutning av digitaliserings-pennan behövs än så länge en sladd, eftersom de hittills använda principerna för digitalisering bygger på detta.

Det är förstas önskvärt att undvara sladden, så småningom kanske tom att undvara hela **TankeNyckeln** fysiskt. Med dagens teknik är det möjligt att tex använda infrarött ljus för kommunikationen. Pennan skulle kunna laddas upp varje gång man ställer den ifrån sig. Problem som då uppstår är tex att man glömmer ställa ifrån sig pennan och man sedan måste vänta ett tag till den laddar upp igen. Det finns dock snabb-laddningsbara ackumulatorer att tillgå.

Utvecklingen av digitaliseringsbord bör dock kunna medföra andra möjligheter i framtiden. Det borde tex gå att göra plattan tryckkänslig varför ett godtyckligt föremål skulle kunna utgöra penna.

Kontakttyp för digitaliseringspennan väljes vid konstruktionen av **TankeNyckeln**.

Kontakt för digitaliseringspenna bör finnas på både vänster och höger sida av presentations-enheten för att möjliggöra alternativ placering för vänster och högerhänta.

3.10.2

Ingång för ljud

Ljud skall kunna behandlas i **TankeNyckeln**. Dels för kommandogivning som komplement till handskrift, dels för att tala in meddelanden som skall sändas till annan persons TankeNyckel. Vid goda förhållanden skall dubbelriktad talkommunikation motsvarande telefoni kunna

upprätthållas. För detta krävs ett dataflöde med ett tidsmedelvärde på ca 64 kBit per sekund. En sådan kanal kan också innefatta en hyfsad bild om deltamodulering används.

För kommandogivning och kommunikation användes en åtta bitars analog-digital-omvandlare med logaritmisk kompression, detta ger en god dynamik trots det kompakta formatet. Ett system som finns tillgängligt med dessa egenskaper är det av Sony introducerade PCM-ljudet för videoinspelning på 8 mm kassetband. Huruvida detta system förväntas bli standard på området går ej sia om fn.

Sony använder samplingsfrekvensen 31,5 kHz och har en dynamik på 88 dB. Systemet klarar att återge ljud mellan ca 20-15000 Hz.

För talkommunikation kan samplingsfrekvensen reduceras.

Mikrofon för kommandogivning etc placeras i pennan.

För övrig ljudbehandling användes 16 bitars linjär omvandling vilket ansluter till den nu gällande standarden för skivor och band, med 44,1 resp 48 kHz samplingsfrekvens. Analog-digital-omvandlare för detta får i början placeras externt.

För behandling av ljud från digitala ljudsystem skall en ingång för detta finnas, för detta användes 1-pol skärmd audiokontakt.

3.10.3 Utgång för ljud

Som ljudkälla finns en liten inbyggd högtalare, ljud kan även tas ut via ett 3.5 mm stereo jack.

För avtappning av digitalt kodat ljud finns en 1-pol skärmd audioutgång.

För generering av ljud användes 16 bitars digital-analog-omvandlare i stereo. Sådana finns idag att få med rimlig ström- förbrukning och god precision från ett flertal tillverkare tack vare utvecklingen av CD-systemet.

16 bitars D/A-omvandlare användes även för att presentera ljud som genererats med 8 bitars olinjär omvandlare.

Högtalarutgång och stereojack har även en programstyrd volymkontroll. Hur denna utförs beror på den valda omvandlarkretsen. De kretsar som tas fram för CD-bruk kommer snart att innehålla även detta.

3.10.4 Ljussensor

På presentationsenhetens framsida skall finnas två ljussensorer. Dessa skall ha förmågan att registrera ljusgradienter från tex Xenon-blixtar resp halogenlampor.

Med hjälp av ljussensorerna kan varje normal kopieringsmaskin användas som snabb skönskrivare. Detta bör vara en värdefull egenskap för affärsmän, författare, kompositörer, föredragshållare etc som snabbt vill kunna skriva ut tex rapporter, manuskript eller stordiaunderlag på den lokala platsen utan att behöva förlita sig på att det finns en anslutningsbar skrivare i närheten. Kopieringsmaskiner finns överallt, kravet på dessa är dock att de skall ha en plan yta för originalet. Vidare måste originalet registreras mha av en ljusgradient som kan fås från tex en blixtlampa eller en avsökande lampa.

Ljussensorn möjliggör för animatörer, konstnärer och andra bildbehandlare som arbetar i färg på **TankeNyckeln** att färgbilder med godtyckligt antal färger kan framställas utan tillgång till andra hjälpmedel än en fotoblixt och tre färgfilter, (rött, grönt, blått), samt en "instant"-kamera. Detta förutsätter dock att blixtens styrka eller kamerans bländare kan varieras.

Ytterligare förutsättningar för att kunna framställa fotostatkopior och färgbilder är dels att bilden på presentationsenheten är statisk, dvs den ej är multiplexad utan uppbyggd av ex bistabila kristaller, dels att den har god kontrast. Antalet möjliga färger alternativt gråskalenivåer är direkt beroende av denna kontrast.

3.10.5 Kameraingång

En av grundidéerna med **TankeNyckeln** är att den skall vara en personlig kuskapsorganisatör. För att kunna uppfylla detta måste det vara enkelt att mata in information i den.

Genom att ge **TankeNyckeln** en ingång för vidokamera eller liknande ökas möjligheterna att smidigt mata in stora mängder information väsentligt.

Då man normalt har sin **TankeNyckel** med sig kan man alltid hämta in intressant information som dyker upp och lägga in i sin kunskapsdatabas efter lämplig filtrering.

Tillämpningarna är givetvis oändliga men som exempel kan nämnas :

I föreläsningssalen ; Föreläsaren skriver hela tiden i ganska rask takt på tavlorna samtidigt som han berättar om det hela. Här får många elever välja mellan att anteckna allt eller förstå.

Med **TankeNyckeln**s hjälp är det enklare. Anteckningarna med lärarens spretiga handstil skannas automatiskt in och förvandlas till ett behagligt typsnitt samtidigt som eleven hela tiden kan göra kompletteringar till lärarens figurer och anteckningar.

TankeNyckeln kan även registrera lärarens tal vilket senare kan återges i samband med att man går igenom anteckningarna.

Vid tidningsläsning o dyl ; När man läser tidningar och tidskrifter ser man ofta intressanta artiklar, ritningar, algoritmer och liknande som är intressanta och verkar användbara. Det normala är då att man springer och kopierar det intressanta ur tidningen om det inte är ens egen. Man har då egentligen bara flyttat problemet. Kopieringen hamnar i en hög eller i bästa fall i en pärm under lämplig rubrik. Tyvärr är den i många avseenden fortfarande oanvändbar. Informationen är inte sökbar, ritningen man hittade är inte direkt tillgänglig i ens design, algoritmen går ej heller direkt att använda i ens program.

Med **TankeNyckeln**s hjälp kan man närsomhelst lägga in godtycklig tryckt kunskap på ett organiserat sätt så den senare är tillgänglig när den behövs genom att avsöka den med kameran.

Kameran måste ha fjärrstyrbar zoom, panorering och höjdinställning. På detta sätt kan godtycklig upplösning erhållas genom att **TankeNyckeln** avsöker området.

3.10.6 Radiotransciever

Radiotranscievern i **TankeNyckeln** är nyckeln till kommunikation med andra enheter.

De önskvärda egenskaper denna del skall ha är följande :

- Tillräcklig bandbredd för att kunna hantera ett informationsflöde av 10 Mbit/s med ett rimligt signal/brus-förhållande.
- Rundstrålande antenn för att riktningen till andra näraliggande enheter ej skall behöva vara känd. Detta är ett svårt kriterium då den stora bandbredden kräver en förhållandevis hög bärvågsfrekvens. Går det ej att uppfylla detta får vissa inskränkningar göras på dataflödes hastigheten till en början.
- Bärågsfrekvensen skall kunna styras programvarumässigt inom ett stort område. Detta är nödvändigt tills en standard för kommunikationen har upprättats. En försöksverksamhet med kraftigt reducerad begränsad bandbredd kan startas inom tex amatörradiobanden på 140 MHz. Visar det sig senare att det går att utnyttja några tiotal MHz på tex 300 MHz skall samma transciever utnyttja denna större bandbredd och frekvens.

3.10.7 IEEE-488 (GPIB)

För forskare och andra som gör mätningar mha elektroniska instrument är det önskvärt att kunna koppla upp sig till det 8 bitars parallellbuss-snitt som benämns IEEE-488 eller GPIB, vilket är den vanligast förekommande instrumentbussen. Kretsarna för detta tar dock både en del ström och plats varför IEEE-488 snittet hellre placeras i en extern enhet. En sådan enhet finns som tillhör till MacIntosh för att ansluta till dess SCSI-port. En ytterligare god anledning till detta är den osmidiga kabel som användes mellan IEEE-488-enheter.

TankeNyckel är tänkt som ett redskap som skall vara där användaren befinner sig, inte att användaren skall vara där **TankeNyckeln** är, av denna anledning är det bra att inte kunna binda upp den fysiskt till en plats som fallet skulle vara om den har en IEEE-488 anslutning.

Det finns dock en god anledning att ta med en anslutning för sådant som IEEE-488. Om det går att göra **TankeNyckeln** tillräckligt billig och universiell så att det går att använda ett stort antal TankeNycklar för även fasta uppkopplingar skall givetvis sådana användas istället för att speciella anpassningsenheter skall tas fram. Det minskar med nödvändighet flexibiliteten.

Frågan är då om IEEE-488 är ett sådan generell buss. Tills denna fråga är utredd byggs inte någon IEEE-488 anslutning in.

3.10.8

Andra snitt

Det finns givetvis en rad andra snitt som är önskvärda för att kunna ansluta sig till andra datorer och kommunikationsnät.

Exempel på sådana kommunikationssnitt är GPIL, HDLC/ADCCP, SDLC, BISYNC, DDCMP, Midi, telefon etc ..

Två saker är klara.

- Det går inte att bygga in alla !
- Det finns ingen anledning att bygga in alla!

Med den skog av kommunikationssnitt som finns idag är det bättre att ha ett fåtal snitt i **TankeNyckeln** och bygga in anpassningar till andra snitt i anslutningssladdarna. På sikt bör vi räkna med att många nu existerande gränssnitt försvinner till förmån för få men generella.

Tanken med **TankeNyckeln** är att den skall utgöra en idé för framtiden, då skall den inte behöva släpa med sig all upptänklig kompatibilitet med existerande oegentligheter.

3.11

Buss

Någon buss i vanlig mening (ex VME, NuBus, MultiBus etc) skall inte finnas tillgänglig i **TankeNyckeln**, detta vore att permanenta de idéer som datorer bygger på idag och att bygga in de begränsningar som dessa har.

Internt kommer den första implementationen av **TankeNyckeln** med största sannolikhet kunna anses ha en 32 bitars buss. Detta med anledning av att de mikroprocessorer som pris och prestandamässigt är intressanta idag har just 32 bitars databussbredd.

3.12

Kommunikations-protokoll

3.12.1

Målsättning

Alla innehavare av **TankeNyckeln** skall kunna kommunicera med alla andra innehavare av likadan utrustning. Varje människa skall också med **TankeNyckeln**s hjälp kunna komma åt alla mänsklighetens kunskaper som är samlade i databasen "World Knowledge" (el motsvarande).

Målsättningen är att alla människor som så önskar skall kunna utrusta sig med en **TankeNyckel**.

Den viktigaste grundprincip som **TankeNyckeln** bygger på är förmågan att kommunicera.

3.12.2

Grundkrav

Kraven på kommunikations-protokollet kan sammanfattas:

- Ett stort antal noder med unika identiteter.
- En rimligt stor databandbredd, minst 10 Megabaud.
- Noderna skall vara flyttbara, de skall ständigt kunna lokaliseras om under pågående kommunikation eller under vila.
- Noderna skall fungera över hela jordytan, protokollet skall dock inte vara begränsat till denna.
- Kommunikation över långa avstånd skall ej vara beroende av att det finns fasta transportnoder. Enda kriteriet för att ett meddelande skall komma fram är att det finns aktiva noder efter hela vägen.

- Fasta noder, datakoncentratorer, skall kunna aktiveras efter hand för att effektivisera kommunikationen över stora avstånd.
- Sändardelens effekt skall automatiskt anpassas (programvarustyrt) efter de yttre förhållandena.
- Antalet möjliga nodadresser skall vara så stort att de inte finns någon risk att de tar slut under de närmaste tusentalen, under antagandet att varje människa som föds får en ny unik adress.
- Nätet skall vara paketsändande, paketen skall ha variabel längd.
- Det skall vara möjligt att sända regionala meddelanden, dvs meddelanden som riktar sig till en viss intressegrupp eller till ett visst fysiskt område.
- Alla noder skall ha samma möjligheter att sända meddelanden, dvs ingen prioritet.
- Moduleringsmetod skall väljas efter kriterierna: god störmarginal, god räckvidd.
- Nätet skall ha god resistens mot icke funktionsdugliga enheter, (detta krav måste specificeras före konstruktion för att vara användbart).
- Protokollet bör vara utbyggbart för att först klara av solsystemet och senare galaxen.

3.12.3

Ansats länknivå-protokoll

Länknivåprotokollet innehåller fält för sändarens och mottagarens nodidentiteter.

Protokollet skall vara oberoende av fasta noder, detta får till följd att identiteten måste kompletteras med fält för sändarens och mottagarens position i ett lämpligt valt koordinatsystem.

Funktionsprincipen för detta protokoll är i stora drag att varje nod antas känna sin egen position med tillräckligt stor noggrannhet. Vid avsikt att sända ett meddelande till annan nod skattas dennes aktuella nodadress med så stor precision som är möjligt. När så meddelandet går iväg kommer det område som svarar mot destinations-positionen att sökas igenom enligt den angivna osäkerheten. Om destinations-noden finns inom det

avsedda området skickar denne ett svarsmeddelande med en mer (om möjligt) detaljerad positionsangivelse, för att snabba upp nästa meddelande.

3.12.4 Positionsfält

Antag att polära ytkoordinater används, alltså latitud och longitud, då vi avser en planet.

Ett koordinatpar antas vara bestämt av 48 bitar. Låt en bit betyda att positionen finns på denna yta eller på annan yta. Låt 5 bitar ange hur stor osäkerheten är i antal bitar. Låt latitud resp longitud bestämmas av 20 bitar vardera. Detta medför $2^{20} = 1048576$ möjliga positioner inom varje koordinat.

På ex jordytan kan positionen då bestämmas inom ca 38 m när vid ekvatorn.

(jordens radie, $r = 6378$ km,

=> omkrets = $2\pi r = 40074$ km .

=> precision = (omkrets/positioner) = ca 0.038 km .)

Denna noggrannhet får anses tillräcklig!

Anmärkning

Polära koordinater är inte optimala om man vill uppnå en enhetlig precision över hela ytan. Vid en av polerna kan tex longituden på 19 meters radie från polen anges med så hög noggrannhet som 0.1 mm, detta är dock acceptabelt.

Polära koordinater är väl etablerade inom vårt samhälle, de har använts inom sjöfarten i många hundra år. Det första kända precisions-instrumentet för att bestämma dessa koordinater vid navigering, spegeloktanten, konstruerades 1731 av optikern John Hadley efter en idé av Isaac Newton.

3.12.5 Ytidentitet

Ansats: Låt en härför avsedd bit koordinatfält betyda att adressen inte härstammar från aktuell kropp (ex jordens) vad gäller avsändarposition resp har en annan kropp som mål om det gäller destinationsfält.

Låt kropps identitet anges med 48 bitars heltal.

Vi antar kropparna, åtminstone till en början, numreras allt efter att de börjar utnyttjas.

Ex: Låt Jorden vara kropp 0, månen kropp 1 osv. En mycket vag uppskattning är att vi efter några hundra år kan ha bemannat några tusen kroppar inom detta solsystem. Med hjälp av 48 bitars kroppsidentitet dvs ca $2.8 \cdot 10^{14}$ identiteter går det att adressera ca 2000 kroppar runt varje stjärna i hela galaxen, detta får anses tillräckligt under överskådlig framtid.

Skulle antalet adresser någon gång ta slut får någon adress reserveras för att betyda utökad adress.

Anmärkning: För att detta protokoll skall fungera i all framtid krävs att vi är ensamma i universum. Skulle så inte vara fallet utan det existerar eller kommer att existera andra interstellära kommunikationsmetoder och kommunikationsprotokoll får vi räkna med att förkasta det här föreslagna protokollet till förmån för ett bättre eller eventuellt att göra systemen kompatibla.

3.12.6 Nodidentitet

Ansats: Låt identitetsfält utgöras av 48 bitar.

Med 48 bitar ges $2.8 \cdot 10^{14}$ möjliga adresser. För att på ett enkelt sätt skilja på unika destinationsadresser och regionala adresser, vilka kan tas emot av ett stort antal noder, kan regional adress markeras av en bit. Då finns fortfarande $1.4 \cdot 10^{14}$ möjliga nodadresser, vilket är drygt 20000 gånger Jordens nuvarande folkmängd, varför de bör räcka under några hundra år.

Anmärkning: 20000 gånger Jordens nuvarande folkmängd är visserligen mycket, men långt ifrån obegränsat. Om vi antar att människan fortsätter att föröka sig och dessutom förlänger sin livslängd samtidigt som tusentals planeter koloniserar, då får adresserna lämpligen utökas genom att låta tex adress ($2^{47}-1$) betyda att ett ytterligare fält på 96 bitar anger identitet.

Det finns säkert många som ifrågasätter att vi överhuvudtaget skall kolonisera rymden. De kanske också ifrågasätter det önskvärda i att kunna kommunicera med varje varelse eller databas i universum. Möjligheten finns ju också att människorna kommer att utveckla nu icke kända förmågor vilka gör hela den föreliggande kommunikationsiden överflödig.

3.13

Centralenhet

De diskussioner som förts en längre tid om vilken som är den bästa centralenheten (ex Motorola 68030, NS 32532, Intel 80386, Transputer TM800 etc..) avfärdas härmed som varande nästan totalt ointressanta för den framtida **TankeNyckeln**.

Om **TankeNyckeln** skall ha en framtid som utbyggbar idé får den helt enkelt inte vara beroende av vilken hårdvara som sitter i den.

De kriterier som är viktigast vid valet av CPU för den första implementeringen av **TankeNyckeln** är :

- Strömförbrukningen. Den måste dra så lite ström att det går att arbeta en längre tid (1 dygn - 1 vecka) utan att den skall behöva laddas.
- Snabbheten. De grundläggande kommunikationsprinciperna i **TankeNyckeln**, tal, grafik, tolkning av handskreven text, tolkning av tryckt text och bild, vilka huvudsakligen kan hänföras till bild och signalbehandling är mycket beräkningskrävande uppgifter. Processorn måste ha tillräckliga prestanda för att dessa uppgifter skall skötas utan att det inverkar störande på det egentliga arbetet som **TankeNyckeln** begärs uträtta.

Vad det gäller strömförbrukningen ligger de flesta av dagens högpresterande 32-bitars processorer på ca 1-2 W, med ett undantag, Acorns ARM - processor som uppges ha en strömförbrukning på ca 0.2 W. Denna låga strömförbrukning till trots uppges den vara ca 2 gånger snabbare att exekvera Lisp än tex 68020. Om strömförbrukningen är så låg går det utan problem att tex ha en processor som sköter det administrativa med mönsterigenkänning etc. Dess prestanda är dock långt ifrån tex Transputer TM800 som är ca 10 gånger snabbare än 68020.

Snabbheten är det som halvledar-tillverkarna främst brukar framhäva med just deras processor. Tekniken för att åstadkomma denna snabbhet är dock känd, dvs att utnyttja Cache-minnen, "pipelining" och parallell exekvering inne i centralenheten etc varför man ej skall stirra sig blind på en viss tillverkares processor av den anledningen. Istället bör arkitekturen studeras för att se hur väl den passar till tillämpningen.

Det har finns idag processorer som är specialicerade på bla Lisp, Forth, Occam, Pascal och C.

Lisp-processor som kommer från Texas Instruments är avsedd för deras Lisp-maskin Explorer. Denna processor "MegaChip", exekverar Common Lisp, strömförbrukningen är dock fn okänd för författaren .

En processor som är mycket intressant i **TankeNyckeln** är den sk Transputern från Inmos. Den är en mycket snabb processor som är specialicerad för språket Occam, vilket är ett språk som innehåller primitiver för bla parallell exekvering och den har ett kanalbegrepp som gör att kommunikation mellan processer är inbyggt i språket. Den har vidare den intressanta möjligheten att kunna kommunicera mycket effektivt med andra Transputers inbyggd i kapseln. Strömförbrukningen är fn (1987) ca 0.9 W och hastigheten ca 20 Mips (Miljoner instruktioner per sekund) och ca 1.5 MFlops (Miljoner flyttalsinstruktioner per sekund). Dess kommunikationsmöjligheter med andra Transputers eller andra enheter består av fyra seriella DMA-kanaler på 10 Mbit/s. Högre kommunikationshastigheter kommer men alla Transputers skall hantera 10 Mbit/s.

Transputer har ej heller höga krav på kristallklocka. För att slippa problem med generering av höga klockfrekvenser har bestämts att Transputer matas med en 5 MHz klocka oberoende av intern klockhastighet. Den interna klockan genereras mha en faslåst slinga.

Pga filosofin med hårdvaruoberoende lämnas valet av processor till konstruktionsfasen.

3.14

Operativsystem

Det behöver nödvändigtvis inte finnas ett operativsystem i maskinen, dvs de vanliga operativsystemfunktionerna kan lika väl vara inbyggda i det språk man tillämpar på maskinen.

Med tanke på den lösning det innebär att välja ett språk med operativfunktionerna inbyggda i denna fas är det bättre att välja ett operativsystem till **TankeNyckeln**.

TankeNyckeln kan ses som en typ av arbetsstation, som Lisp-maskin eller Cad system, specialicerad på kommunikation och användargränssnitt. I de flesta av dagens arbetsstationer använder nästan uteslutande UNIX. Därmed inte sagt att UNIX skulle vara bäst men dess spridning är stor och tillgången på programvara god, vilket kan medföra att utvecklingsarbetet på **TankeNyckeln** kan inskränkas till hårdvarukonstruktion och konstruktion av de speciella användargränssnitt som **TankeNyckeln** kräver.

Då **TankeNyckeln** skall vara oberoende av vald hårdvara vad gäller bla processor bör ett krav på operativsystem och all annan programvara vara att den skall kunna installeras från källkod på varje ny TankeNyckel oberoende av vald hårdvara. Givetvis måste viss anpassning kunna krävas men källkoden skall vara tillgänglig för alla.

Dagens UNIX-system uppfyller inte dessa krav. Det mesta av källkoden släpps inte ut av programvarutillverkarna.

Om som första operativsystem till **TankeNyckeln** istället väljs GNU (Gnu is Not Unix !) kan många kriterier om hårdvaruoberoende uppfyllas då denna programvara är fri och distribueras i källkodsform.

Bakom utvecklingen av GNU står Richard M Stallman och Free Software Foundation. GNU kommer att vara helt UNIX-kompatibelt. Det utvecklas i första hand för att underlätta det fria programvaruutbytet.

3.15 Språk

Här följer en kort genomgång av några programmeringsspråk som i dag förekommer på persondatorer resp arbetsstationer och skulle vara en möjliga kandidater till system resp applikationsspråk i **TankeNyckeln**.

3.15.1 BASIC

BASIC är det första sk högnivåspråket som började användas på persondatorer. Det är en akronym för "Beginners All Purpose Sequential Instruction Code".

Anledningen till att BASIC togs fram var att det behövdes ett litet minnessnålt språk för programmering av små maskiner med låga prestanda.

Fördelar:

- variabler behöver ej deklareraras.
- interpreterande, dvs program behöver ej kompileras.
- lätt att lära sig då egentlig syntax saknas.

Nackdelar:

- Radnummer, vilka dessutom är globala över hela programmet.
- Enda datatyper är flyttal, strängar och arrayer
- Enbokstaviga (plus siffra) variabelnamn.

- Lokala block och variabler saknas.
- Parameteröverföring saknas.
- IF..THEN,GOTO/GOSUB är enda struktureringsverktygen.

Mot de senare nackdelarna kan visserligen anföras att det visst finns BASIC-varianter som ej behöver radnummer, där datatyper integer och fält (jfr Pascal record) finns. Procedurer och funktioner med lokala variabler och parameteröverföring. Långa variabelnamn och struktureringsverktyg som "REPEAT..UNTIL", "WHILE..ENDWHILE", "IF..THEN..ELSE", etc. Frågan är då om detta skall kallas BASIC, snarare någon form av interaktiv PASCAL. "Basic"-varianter av den senare kategorin finns dock bara till vissa maskiner och kan därför ej anses representativ.

BASIC stödjer varken en funktionell, datastrukturmässig, objektlik eller logisk problemformulering. BASIC bör därför endast användas för program upp till tio rader eller inte alls.

Dagens persondatorer (1986), ex Apple MacIntosh, IBM PC-AT etc har så pass höga prestanda att BASIC inte längre behövs. Det kommer förmodligen att dröja några år innan det försvinner även på fickräknedosorna även om det nu skulle vara fullt möjligt att tex använda LISP på dem istället.

3.15.2 LISP

LISP är ett mycket gammalt språk med en oerhört enkel syntax. Det har dock haft den nackdelen att det kräver stora minnesvolymmer varför det inte var användbart på 70-talets persondatorer.

LISP är ett funktionellt språk som är ypperligt väl anpassat att lösa små problem med, det är vidare väl lämpat både för rekursiva och iterativa lösningar.

LISP bygger på idéer från ett formellt språk benämnt Lambda (λ) -kalkyl, vilket introducerades av matematikern Alonso Church 1941. λ -kalkyl användes idag som ett matematiskt hjälpmedel för studier av sematiken hos programmeringsspråk.

LISP arbetar med ett fåtal datatyper integer, flyttal, strängar, arrayer, listor och bignum (heltal av godtycklig storlek). Lisp-program är själva listor. Det finns

vidare en del varianter av objektorienterad LISP på marknaden, bla COMMON LISP i vilken egna typer dessutom kan definieras.

LISP har blivit ett populärt språk idag, det som är mest tilltalande med LISP är dess enkelhet och dynamik. Det anses att en problemlösare eller programmerare som börjar med LISP mycket snabbt blir produktiv just på grund av att språket med sin mycket enkla syntax blir transparent till förmån för det problem som skall lösas.

LISP är normalt interaktivt men kan kompileras till goda prestanda.

Ett av problemen med LISP har varit att det varit dåligt standardiserat, det har florerat ett antal olika versioner som, MacLisp, FranzLisp, InterLisp, XLisp, ZetaLisp etc. Vidare har LISP varit litet besvärligt för stora system pga att alla statiska variabler och funktioner har varit synliga globalt. Nu har dock LISP standardiserats till något som heter COMMON LISP, där har även paketbegrepp införts varför det språket har goda förutsättningar även för stora projekt.

Det finns en släkting till COMMON LISP som heter SCHEME, detta är ett språk som är under ständig utveckling och ännu så länge inte finns i många kommersiella applikationer. SCHEME har utvecklats vid MIT och blev officiellt 1978. Den tredje reviderade rapporten av språket har nyligen utkommit från MIT.

SCHEME är ett sk objektorienterat språk och användas mycket som undervisningsspråk. Till undervisning är det väl lämpat pga sin enkelhet och renhet.

SCHEME finns att få som fri programvara avsedd för UNIX-system från "Free Software Foundation". Denna är skriven i språket C.

SCHEME finns i kommersiella varianter för bla Apple Macintosh och Texas Instruments Explorer samt IBM PC-kompatibla datorer.

3.15.3 PROLOG

Prolog är ett experimentspråk som är speciellt konstruerat för exakt inferenskalkyl baserad på fakta och regler uttryckta på Horn-klausul form.

Detta innebär att logiska slutsatser kan dras genom att frågor mot en databas bestående av fakta och regler innehållande OCH resp ELLER villkor kan ställas och besvaras.

Prolog brukar innehålla datatyperna integer, flyttal, sträng och lista. Dessa typer finns även i Lisp, vilken innehåller några fler typer. I modern Lisp kan också egna datatyper definieras.

Det går att använda Prolog som generellt programmeringspråk men endast rekursion, ej iteration är definierad. Det är ju inte alla problem som har en uppenbar rekursiv lösning även om detta är fördomsfullt.

En mycket intressant sak med Prolog gentemot exempelvis funktionella och objektorienterade språk är att förhållandet mellan in och utparametrar inte är enkelriktat. Om vissa parametrar till en funktion ger en viss lösning är det i Prolog lika rimligt att ge lösningen och erhålla alla tänkbara parametervärden istället.

Den inferenskalkyl som finns i Prolog går ganska lätt att bygga upp i Lisp, varför Lisp snarare är att föredra då detta är ett mer generellt språk, Lisp är väl lämpat både för rekursion och iteration. Prolog är däremot mycket lämpligt att lära sig logikprogrammering i.

Det finns flera implementationer av Prologfunktioner i Lisp, tex LM-Prolog på Lisp-Machine som kombinerar styrkan hos Lisp och Prolog, detta utgör ett mycket starkt programmeringsverktyg.

Ett problem med Prolog har varit att stor olikhet har rått mellan olika implementationer vad gäller syntax.

Prolog är tänkt att utgöra grunden för den japanska satsningen på de sk femte generationens datorer. Det är ett språk som väl lämpar sig för parallellt arbetande processorer, här pågår dock mycket forskning. Det svenska SICS (Swedish Institute of Computer Science), arbetar bla med att ta fram en parallellt arbetande Prolog-maskin.

En begränsning i användbarheten för att tex bygga upp expertsystem med hjälp av Prolog är att det fn ej går att uttrycka inexakta samband. Med hjälp av sk "Fuzzy logic" erhålles möjligheten att uttrycka att något är sant med viss signifikans om något annat är "nästan" sant. Även detta är föremål för omfattande forskning på AI-labben. Moderna verktyg för att bygga expertsystem brukar dock ha vissa möjligheter att uttrycka inexakta samband.

Prolog finns att få i någon form till de flesta maskiner men någon allomfattande Prolog-standard existerar ej. De flesta är baserade på den sk DEC-10 Prolog som redovisas i boken "Programming in Prolog" av Clocksin och Mellish, men många varianter förekommer.

Det finns en Prolog tillgänglig i källkodsform från University of New Hampshire. Denna sk UNH Prolog saknar dock fn "garbage-collection" varför den endast är lämplig för laborativ verksamhet.

3.15.4 LOGO

Språket LOGO skapades av Seymourt Papert för barn och kan ses som ett försök att åstadkomma en förenklad form av LISP där man ej behöver arbeta med paranteser.

Avsikten med språket var inte att lära barn programmera, snarare att lära barn känna sig fram och bygga upp kunskap om verkligheten utgående från sig själva och modeller baserade på den egna kroppens erfarenheter.

Med hjälp av LOGO skall barnen lära sig genom att själva vara aktiva vilket de oftast inte är bortskämda med i skolan.

Idén bakom LOGO är att datorerna, programmen eller problemlösningen med hjälp av dator inte skall vara det intressanta. Barnen skall istället endast se datorerna som abstraktionshjälpmedel, vilka underlättar för dem att bygga modeller av verkligheten och att testa idéer.

LOGO finns att få till alla vanligt förekommande persondatorer.

3.15.5

MODULA-2 och PASCAL

Modula-2 och Pascal är visserligen två olika språk men de har mycket gemensamt och är båda skapade av Niklas Wirth vid Zürich universitet.

Pascal är ett litet språk med en väl definierad syntax men tyvärr med dåligt definierade snitt mot omgivningen. Pascal är utmärkt att lära sig blockstrukturerad programmering i och detta var också anledningen till att Niklas Wirth tog fram detta språk.

Pascal kan ses som en idé till vidareutveckling av de Algolliknande språken som kom på 60-talet.

Niklas Wirth undervisade bla i programmering och behövde för detta syfte ett litet formellt språk som skulle vara lätt att lära sig och lätt att använda som pseudo-språk.

Pascal var därför till en början endast tänkt för undervisning och Niklas Wirth tänkte inte alls skriva någon kompilator till det men blev så småningom övertalad till detta. Då Pascal från början inte var tänkt att realiseras som ett verkligt språk var också definitionen bristfällig på många punkter. Det finns en rad egenheter i språket som gör det mycket stelbent, speciellt för programsystem. Dessa egenheter gör att en del mindre renläriga trix ibland får tillgripas vid tillämpningar med språket.

Bland del allvarligaste nackdelarna med språket Pascal kan nämnas att :

- separatkompilering av moduler är svagt definierad.
- generella procedurer för hantering av arrayer och matriser går inte att skriva enligt definitionen.
- generella typkonverteringar saknas.
- dåligt definierad in och utmatning resp filhantering.
- loopkontrollen är ren men svag.
- felhantering saknar stöd från språket.
- manipulering av adresser kräver ofta knep och knåp.
- svårt att dölja information för en överliggande nivå.

Lika som med Basic finns det Pascal-varianter där man kommit runt de flesta problemen med hjälp av tillägg till språket. En variant som är kraftigt utbyggd, är den sk VAX-Pascal som finns till Digital's VAX-datorer.

VAX-Pascal innehåller så många speciella men dock mycket användbara undantag från definitionen att den knappast längre kan kallas Pascal.

Försök har även gjorts att standardisera vissa tillägg till språket, den vanligast refererade är ISO-Pascal.

Vad gäller Modula-2 har Niklas Wirth försökt komma runt de flesta bristerna med Pascal utan att fördenskull skapa ett stort och ohanterligt språk. Modula-2 utvecklades för att vara systemprogrammeringsspråk på Lilith, en arbetsstation som utvecklats bla under professor Niklas Wirths ledning.

Modula-2 innehåller bla verktyg för att beskriva parallella skeenden ("task") och är där överlägset de flesta former av Pascal.

Styrkan hos Modula-2 är att det inte behöver kännas onödigt stort. Det är ett litet och väldefinierat språk som med sitt modulbegrepp tillåter en robust systemkonstruktion.

3.15.6 SIMULA

Simula har definierats år 1967 och är framtaget vid Norsk Regnesentral i Oslo. Det har reviderats något 1970.

Språket bygger likt Pascal på Algol, det är dock mycket kraftfullare än Pascal och har fått en oförtjänt dålig uppmärksamhet. Språket har en väl utbyggd dynamik och en fungerande objektmodell. Simula saknar Pascals egendefinierade skalära datatyper. I Pascal kan man definiera egna strukturerade datatyper sk RECORD. I Simula finns begreppet CLASS, vilket kan ses som en generalisering av Pascals RECORD. CLASS-begreppet är enligt dagens definition av objektorienterade språk en implementation av objekt.

Språket Simula togs ursprungligen fram för simulering men blev snart populärt som generellt programmeringsspråk.

Idag finns en mängd objektorienterade språk och det anses allmänt att det är Smalltalk från Xerox som är grunden till dessa, Simula förtjänar att nämnas oftare i detta sammanhang.

Simula har inte stöd för "äkta" parallelexekvering, men däremot ett stöd för pseudoparallellism som är mycket kraftfullt och inte går att enkelt efterlikna i många andra språk, ex inte alls i Ada, vilket dock har "äkta" parallellism.

Språket Simula får dock anses som ointressant i frågan om persondatorer. Någon känd implementation existerar ej.

Simulas fader håller på att konstruera ett nytt språk med arbetsnamnet BETA. Även detta blir ett objektorienterat språk.

3.15.7 ADA

Ada hör till de blockstrukturerade språken, typ Algol och Pascal. Det är framtaget med stöd av DoD (Department of Defense, USA:S försvarsdepartement) och är ett språk som det medvetet har investerats stora summor i innan det blivit produktivt.

Till språkets stora fördelar hör en mycket hård specifikation varför program skrivna i detta språk lätt går att flytta från en maskin till en annan.

Ada är normalt mycket lättläsligt varför det är väl lämpat att skriva specifikationer i. Språket har vidare en formalism som gör det till en lämplig ersättare för Pascal inom undervisningen om blockstrukturerad programmering.

Till språkets allvarligaste nackdelar hör avsaknad av objekt och dynamik. Detta med dynamiken är dock helt avsiktligt, språket är helt och hållet avsett för så kallade "embedded systems" alltså "inbyggda system", dvs system vars program inte är avsedda att ändras under drift, tex system som missiler och tvättmaskiner.

Till språkets fördelar kan nämnas möjligheter till generella typkonverteringar och att i det i detalj går att specificera hur data och även programkod skall organiseras i minnet. Språket innehåller också verktyg för att på ett snyggt sätt förvandla normalt maskinberoende handlingar till maskinberoende genom att man i programmet kan ta hänsyn till den aktuella hårdvarukonfigurationen, ex antalet bitar i maskinord, adressenhet etc. Genom dessa möjligheter kan man i hög grad göra sig oberoende av assembler.

Tack vare språkets "generics" kan man tex slippa upprepa en algoritm flera gånger för olika datatyper. Detta förenklar programunderhållet samt ökar läsligheten. "generics" kan betraktas som en form av "macros" jfr assembler, C och Lisp. Allt måste alltså vara känt vid kompileringstillfället.

Ada kan nog vara intressant för många tillämpningar men för persondatorer är det en aning osmidigt och statiskt. För den som utvecklar i ADA finns speciella arbetsstationer. För den som vill prova ADA på persondator (IBM-kompatibel) finns en interpretator att köpa från New York University för 95 dollar. (Januari 1987)

De flesta som tidigare använt Simula hade väntat sig mer av ADA. Speciellt med tanke på att språket är mycket yngre och konstruktörerna av ADA har studerat bla Simula.

3.15.8 FORTH

Forth är ett mycket kompakt språk av funktionell typ som är användbart även på mycket enkla datorer med begränsat minne. Språket är intressant och har många anhängare men räknas av vissa som hörande till kategorin "Write-only" språk.

I Forth finns egentligen bara en datatyp, 16 alternativet 32 bit heltal, men nya typer kan dock lätt byggas.

Flexibiliteten påminner om Lisp. Språket är väl lämpat att skriva små styrsystem odyl med. Det togs också fram för det ändamålet av astronomen Charles Moore, han behövde ett bättre språk än assembler för styrning av radioteleskop.

Forth finns att få till de flesta maskiner. Det finns dessutom att få i källkod från "Forth Interest Group". Det har dessutom nyligen utvecklats både 16 och 32 bitars mikroprocessorer vilka är specialicerade på Forth.

3.15.9 APL

Apl är ett språk speciellt dedikerat till matematiskt modellbyggande och experimenterande och är till detta ändamål mycket effektivt. Det hör dock i ännu högre grad än Forth till kategorin "Write-only" språk, dvs det går oftast snabbare att skriva om en sats än att lista ut vad en speciell sats i ett program gör. Program brukar fö ofta endast vara en rad långa.

Språket kommer att ha sin givna plats under lång tid framöver. Det kräver mycket minne men är fullt användbart på persondatorer idag.

3.15.10

C

C togs fram i huvudsak som systemprogrammeringsspråk för UNIX-implementering av Dennis Ritchie.

C är ett litet språk utan skyddsnät. Det liknar Pascal till stor del men språket tillåter många smarta konstruktioner vilka lätt kan inverka menligt på läsligheten.

Många betraktar språket C som en maskinberoende assembler och är som sådan givetvis att föredras.

C-kod är ofta mer portabel än Pascal varför C alltså bör övervägas om ett program skall kunna flyttas till andra maskiner (vilket förhoppningsvis är fallet).

Bruk av C bör kanske inte uppmuntras men bör heller inte förbjudas (vilket däremot Basic kanske borde). Språket fyller helt klart en viktig uppgift för närvarande genom att vara ett språk som tillåter operationer på låg nivå utan speciella skyddsnät samtidigt som det är möjligt att göra konstruktioner på hög nivå med det. Det är väl lämpat som det systemnära språk som det avsåg att vara.

Varje maskin bör vara utrustad med en C-kompilator då en stor mängd programvara finns tillgänglig i C. En av de mest betydande källorna kommer under de närmaste åren att vara GNU-projektet som genomdrivs av Richard M Stallman och "Free Software Foundation Inc." (en eloge till Stallman).

Richard Stallman har nyligen meddelat (feb 1987) att C-kompilatorn till GNU är färdig. Denna finns att få som fri program- vara i källkodsform från "Free Software Foundation". Denna kompilator är förmodligen unik på det sättet att det inte behöver skrivas någon specifik kodgenerator för varje speciell processor. Kompilatorn får istället en abstrakt beskrivning av processorns arkitektur, instruktionsuppsättning och adresseringsmetoder i ett LISP-liknande språk. Denna kompilator optimerar registerutnyttjandet över hela den kompilerade programmodulen. Det tar ca 2 till 3 dagar att göra en sådan beskrivning av en processor (enl Stallman).

Till denna kompilator medföljer även en "debugger" alltså ett verktyg för programtest. Denna "debugger" skall enligt Stallmans egen utsaga vara det bästa som finns fn ifråga om "debuggers" för C.

3.15.11
OCCAM

Occam är det språk som Inmos släpper ut som lägsta språknivå till sin Transputer. Till Transputern finns mao ingen officiell assembler varför Occam fyller denna funktion. Skälet till detta anges framförallt vara att då Transputern är optimerad för att exekvera Occam så producerar en kompilator minst lika bra kod som om den gjorts för hand.

Occam är det enda av språken i denna studie som har ett direkt stöd inbyggt för processornas förmåga till parallell bearbetning och kommunikation mellan processer och andra enheter. Mekanismen som finns i Ada för detta liknar denna något varför det bör gå att åstadkomma en effektiv implementering av Adas "tasking" i Transputer.

Som programmeringsspråk avsett för "multitasking" är Occam att betrakta som minimalt men långt ifrån primitivt.

Med hjälp av Occam kan ett system beskrivas som en samling konkurrerande processer vilka kommunicerar med varandra genom kanaler.

Ett system av processer behöver vidare inte låsas till en Transputer utan kan spridas ut till ett godtyckligt antal Transputers utan någon som helst programvaruändring för att öka ett systems prestanda.

Processer är hierarkiska, dvs en process kan bestå av parallellt eller sekventiellt arbetande processer vilka i sin tur kan bestå av nya processer osv ner till lägsta instruktionsnivå.

Program i Occam byggs av tre primitiva processer:

- Tilldelning : Ändra värde på variabel.
- Input : Ta emot ett värde från en kanal.
- Output : Sänd ett värde till en kanal.

Processer kan kombineras med följande samband:

- Sekvens : Komponenterna exekveras i följd.
- Parallell : Komponenterna exekveras samtidigt.
- Alternativ : Den som är först färdig exekveras.

Occam stödjer typerna flyttal, heltal 1/2/4 byte, vektorer och record (dvs sammansättningar av typer till fält). Kanalbegreppet i Occam utnyttjar de vanliga typdeklarationerna.

3.15.12

Andra språk

Det finns givetvis en mängd programmeringsspråk som många anser vara överhoppade här men de för arbetsstationer vanligaste språktyperna skall vara representerade.

Språk som Fortran, Pl1, assembler och Cobol har medvetet hoppats över. Smalltalk är nog intressant men där saknar författaren för närvarande erfarenhet. Det råder även en viss brist på effektiva implementationer. Tektronix har dock fått fram en mycket snabb Smalltalk till sin 68020-baserade arbetsstation. De som är närmare intresserade av Smalltalk hänvisas till augusti numret av BYTE 1981.

3.15.13

Vilket språk?

Vilket språk skall då väljas som det för all framtid allena saliggörande språk som **TankeNyckeln** skall utvecklas i?

Svaret på detta blir givetvis : **Inget !**

Inget idag existerande språk kan förmodas innefatta sådana kvalitéer att det inte kommer att föråldras inom en överskådlig framtid. Vi får inte glömma att datavetenskapen ännu är mycket ung och att utvecklingen inom både hårdvara och mjukvara snabbt går framåt.

För **TankeNyckeln**s del gäller att så lite beroende som möjligt av något specifikt språk skall byggas in i designen. Detta är givetvis lätt att säga men svårt att genomföra praktiskt. Det går dock att komma långt genom att hålla kopplingarna mellan moduler på en låg nivå och i mesta möjliga mån endast utbyta data genom meddelanden.

Det språk som väljs som systemspråk exempelvis C, medför ett visst beroende mot systemrutiner.

Applikationer skall endast i begränsad omfattning skrivas i det språk som utgör systemprogrammeringsspråk.

Preliminärt föreslås C alternativt OCCAM utgörande lägsta synliga nivå mot hårdvaran. Valet av OCCAM faller sig naturligt men är inte nödvändigt om Transputer väljs som centralenhet.

Som applikationsspråk föreslås någon form av Lisp, företrädesvis COMMON LISP med utökat stöd för objekthantering ex "CommonObjects".

4VISIONER

INNEHÅLL

4.1	Inledning	61
4.2	Teknisk utveckling	61
4.3	Psykosocial utveckling	63
4.4	Transport	63
4.5	Produktion	64
4.6	Datalogins utveckling	64
4.7	Avslutning	65

4

VISIONER

4.1

Inledning

Vi "skriver" i år 2037. Det har gått ett sekel sedan Konrad Zuses konstruerade den första elektromekaniska digitalräknaren vars fundamentalt enkla princip gjorde datorn, människans genom tiderna underbaraste uppfinning, möjlig.

Förmodligen skulle samhället existerat fortfarande även utan datorer men ingen är längre riktigt säker på detta.

Under föregående århundrade drabbades mänskligheten av en rad problem som idag endast finns i historiedatabaserna. Här nämns sådana oegentligheter som svält, drogmissbruk, apartheid, miljöförstörelse, byråkrati, utbildningsproblem, arbetslöshet och det allorstädes närvarande hotet att en ledare för någon av den tidens "länder" skulle välja att lösa en uppkommen konfliktsituation med icke fredliga medel. Ord som "VÅLD" och "KRIG" var vanliga under denna tid.

Idag har de flesta äldre människor i regel förträngt att dessa problem har existerat. Många unga vägrar ofta under sin primärutbildning tro på att vi människor har burit oss åt på detta vis. Det händer dock ofta att unga människor övertalar någon med associationsbanor kvar från den tiden att "berätta" om den tidens fasor och oförstånd.

Den tekniska och psykosociala utvecklingen har bara under de senaste 50 åren förändrat samhället genom att dra fram de goda egenskaperna hos människan i en så hög grad att det så sent som 1970-1990 knappast var möjligt att föreställa sig. Under denna tid fanns tom de som tappat tron på både tekniken och människan.

4.2

Teknisk utveckling

Datortekniken utvecklades från början ganska sakta och spreds inte mycket. I och med uppfinnandet av "mikroprocessorn" 1971 kom dock en rad leksaker att spridas de närmaste 20 åren under namnet datorer. De flesta av dessa led av så stora brister både tekniskt och programvaru-

mässigt att många av dem ens under denna tid var värda namnet. Det var inte dessa datorer som skulle föra mänsklighetens kultur framåt.

Under slutet av 1980-talet hade bla en ung civilingenjör från området Sverige insett att utvecklingen höll på att gå in i en återvändsgränd som det skulle ta mycket lång tid att vända tillbaka ur, om någonsin.

Han konstruerade därför "MindWrench" eller "TankeNyckeln". Detta i och för sig enkla verktyg kombinerade denna tids bästa teknik ifråga om snabbhet, generalitet, portabilitet, inmatning och presentation med det som alla tidigare liknande datorer egentligen saknade ; kommunikation !

TankeNyckeln blev en succe, den var redan från början så enkel och generell att den under lång tid inte kunde förbättras på andra sätt än att göras mindre, snabbare, energisnålare och framförallt billigare.

Från början var den så pass stor att den gick lätt att arbeta med genom att hålla i ena handen. Efter en tid när presentationstekniken förbättrats kunde den bäras runt halsen med "display chipen" i en slags glasögon.

Sedan en längre tid har man inom mikrobiologin behärskat tekniken att erhålla synaptiska kontakter mellan elektroniska respektive optiska kretsar och nervceller. Grunderna för detta har sitt ursprung från 1987 då några forskare lyckades få nervändar att växa ihop med kiselceller. Tack vare detta och att kretsarna nu är så strömsnåla att de kan erhålla sin strömförsörjning direkt från kroppen, brukar många operera in sina tankenycklar direkt under huden. Att "se" har under senare år blivit ett mycket mångtydigt begrepp.

TankeNycklar började från 1998 att delas ut i underutvecklade områden som bistånd. 2012 lades varje människas rätt till CPU-kraft och kommunikation till FN:s fördrag om de mänskliga rättigheterna.

De flesta hem har idag optiska kommunikationskanaler som klarar 10 GigaBit/s, en lokal stationär "dator" finns även för att distribuera och behandla denna information. Angett med forna dagars prestandanormer anses dessa lokala datorer ligga runt 10 - 100 TIPS (Tera instruktioner per sekund).

Forna tiders bildskärmar och högtalare finns numera mest på muséer och hos nostalgiker. Bild och ljud kan nu återges med en tidigare oöverträffad briljans och distorsionsfrihet genom att hoppa över det tidigare nödvändiga analoga inslaget i överföringen. Upplevelsen sker nu på ett för

hjärnan optimalt sätt utan att gå omvägar via synceller och hörselnerver. Det kanske är onödigt att påpeka, men äldre begrepp som "blindhet" och "dövhet" används inte längre.

En intressant detalj med den fast installerade datorkraften är att varje människa vid varje ögonblick är garanterad en åttamiljarddel av denna, men kan närhelst tillfälle ges, tack vare TOS (Terrestrial Operative System), erhålla upp till 50% av världens samlade datorkraft för tex krävande simuleringar.

4.3

Psykosocial utveckling

Människor har idag inget klart begrepp om vad som skiljer arbete från fritid. De flesta människor skulle med äldre tiders mått betraktas som mångsysslare och tusenkonstnärer.

Ingen lider idag av okunnighet, alla som i något sammanhang inte finner någon viss information i hjärnans lokala minne hämtar, i regel omedvetet, via sin TankeNyckel upp informationen ur mänsklighetens samlade databank, vilken ständigt växer då så stor del av mänskligheten nu ägnar sig åt forskning.

Alla människor över hela världen kan nu umgås med varandra utan hänsyn taget till vilket språk de talar. De flesta människor behärskar något av de accepterade globalspråk som vuxit fram. Språket spelar normalt inte så stor roll utom i mycket känsloladdade situationer. Två människor som samtalar med varandra via sina TankeNycklar märker normalt inte att samtalet simultanöversätts. Vardera personen hör en i det närmaste perfekt syntes av av den andre personens röst laddad med de sinnesstämningar som denne känner för ögonblicket.

4.4

Transport

Forna tiders bilar används numera inte för transport, de betraktas mest som kuriosa av samlare. När någon bestämmer sig för att resa någonstans tar det i regel mellan en halv och fem min inom tätort, upp till tio min inom glesbygd, innan fordonet, "PersonKonten" dyker upp efter att beställningen har "tänkts".

PersonKonterna, dessa små kapslar som finns i flera olika storlekar för en, två, fyra eller åtta personer, rymmer en oerhörd komfort och tillåter att människor kan förflytta sig mellan olika punkter på jorden mycket snabbt.

Systemet fungerar som en global energipendel, Vid en resa över tex Atlanten accelereras kapseln halva vägen elektromagnetiskt i en nästan lufttom tunnel, den andra halvan av tiden retarderar den genom att fungera som generator.

Flygning och båttrafik existerar men används inte längre som transportmedel inom atmosfären. För tyngre transporter används dock luftskepp. För att kommunicera mellan rymdstationerna och forskningsbaserna på månen används även här en form av energipendeln, men med något kraftigare kapslar. Från några av jordens högsta berg mynnar dessa tunnlar till rymden.

4.5 Produktion

Världens samlade produktionsmaskineri består av en nästan underhållsfri maskinpark som är så flexibel att lagerlokaler inte längre behövs.

Den som behöver en viss detalj tillverkad, formulerar endast behoven via sin TankeNyckel varvid detaljen produceras i den mest lämpliga fabriken map svårighetsgrad, avstånd och hur snabbt den behöver vara klar. Om en detalj ej går att tillverka med en rimlig insats av de rådande produktionsresurserna vidtas istället en analys av vilken produktionsenhet i världen som är mest lämpad att modernisera för att detaljen skall kunna tillverkas.

4.6 Datalogins utveckling

Här är en kort ungefärlig sammanställning av utvecklingen inom programmeringsmetodik under de senaste 50 åren räknat i generationer.

3:e Generella språk med hög frihetsgrad som LISP, SETL etc.

4:e Logik och databasspråk som Prolog, Mapper, OPS5 etc.
Hjälpmedel som MacSyma.

5:e I huvudsak högpresterande logikspråk med höga inslag av parallellism , lättbearbetade användarsnitt och stora inslag av sk AI .

6:e Logikspråken har utvecklats till att behandla även osäkra fakta, regler och samband. En egenskap liknande intuition kan formaliseras.

- 7:e Distribuerade operativsystem och arkitekturer för optimal parallell bearbetning.
Begrepp som TOS ("Terrestrial Operative System") och TVM ("Terrestrial Virtual Memory") definieras.
- 8:e Problemformuleringar och hypotesprövningar görs genom att direkt relatera fakta, hypoteser och teorier inom TVM .
- 9:e Förfiningen i kommunikationen mellan människa och TOS har möjliggjort att man nu kan arbeta med en abstraktionsnivå som ligger på gränsen till människans fantasi. Ett problem eller en lösning som tidigare har varit nästan omöjlig att abstrahera kan nu formuleras genom att med små antydningar "tänka" så nära problemet som möjligt. Den slutliga formaliseringen erhålles efter automatisk eller "manuell" korrelation mellan alla möjliga och många omöjliga formuleringar eller lösningar tills tillräcklig signifikans uppnåtts.

4.7

Avslutning

Den civilingenjör som konstruerade den första **TankeNyckeln** arbetar nu med forskning rörande åldrandet, hans kommentar:

- "När vi kommit så här långt under de senaste hundra åren, vem vill då missa de kommande tusen."

REFERENSER

- 1 BYTE augusti 1981, Vol 6, No 6 (Smalltalk).
- 2 IVA, "Framsteg inom forskning och teknik 1986". ISBN 9170824169. (Flytande kristaller)
- 3 Olof Beckman, "Värmelära". ISBN 91-20-04465 (Värmetransport)
- 4 Frank P. Carau, Henry T. Hetzel, Lawrence E Brown, "Easy-to-Use, High-Resolution Digitizer Increases Operator Efficiency", Hewlett Packard Journal, s 2-13, dec 1978. (kapacitiv digitaliseringsteknik)
- 5 Arun Madan "Amorphos silicon: from promise to practice", IEEE Spectrum, s 38-43, sep 1986.
- 6 King-Sun Fu, Y.T.Chien, Gerald P.Cardillo, "A Dynamic Programming Approach to Sequential Pattern Recognition", IEEE Transactions on pattern analysis and machine intelligence, Vol PAMI-8, No 3, May 1986.
- 7 ELFA-katalogen nr 35, 86/87.
- 8 Jean Renard Ward, Barry Blessner, "Interactive Recognition of Handprinted Characters for Computer input", SIGCHI Bulletin, s 44-57, july 86, vol 18, n 1. reprinted from IEEE Computer Graphics, vol 5, no 9, s 24-37, sep 1985.
- 9 Douglas R.Hofstadter, "The Mind's I, Fantasies and reflections on self and soul", ISBN 014 00.6253 X
- 10 Xerox, "The Ethernet, a Local Area Network, Data Link Layer and Physical Layer Specifications", version 1.0, sep 30, 1980.
- 11 Fujitsu Microelectronics, inc, "SCSI protocol controllers, a comparison of Fujitsu's MB87030 vs NCR's 5385E/5386"
- 12 Toshiba Corporation, "MOS MEMORY PRODUCTS", data book '85-9
- 13 Jonathan Rees, William Clinger (Editors), "Revised³ Report on the Algorithmic Language Scheme", SIGPLAN Notices, vol 21, no 12, dec 1986.

- 14 David Spector, "Ambiguities and Insecurities in Modula-2", MIT/LCS/TR-225, MIT Laboratory for Computer Science oct 79.
- 15 Joel McCormack, Richard Gleaves, "Modula-2 A Worthy Successor to Pascal", BYTE, s 385-395, april 1983.
- 16 Otis Port, "Computers that come awfully close to thinking", Business Week, june 2, 1986.
- 17 W.F.Clocksinn, C.S.Mellish, "Programming in Prolog", ISBN 3-540-15011-0 alt ISBN 0-387-15011-0.
- 18 Seymour Papert, "Tankestormar (Alternativ pedagogik med dators hjälp.)" , ISBN 91-37-08510-7.
- 19 Murray R.Spiegel, "Mathematical Handbook", Schaum's outline series in mathematics, McGraw-Hill Book Company.
- 20 Harol Abelson, Gerald Sussman, Julie Sussman, "Structure and Interpretation of Computer Programs", The MIT Press, ISBN 0-262-01077-1 alt ISBN 0-07-000-422-6.
- 21 Göran Mård, "24 timmars musik på en kassett", Elektronikvärlden (om Sonys 8-bit PCM-ljud), s 16-18, februari 1986.
- 22 Acorn Computers Limited, "Acorn RISC Machine CPU Software Manual", Issue 1.00, inkluderade några prestandamätningar.
- 23 INMOS Corporation, "The Transputer Family", Product overview june 1986.
- 24 INMOS Corporation, "IMS T424 Transputer Reference Manual".
- 25 Alan Snyder/Hewlett Packard Laboratories, "CommonObjects: An Overview", SIGPLAN NOTICES, s 19-28, vol 21, nr 10, oktober 1986.