

Hå hå hååå — god jul i stugorna!

Nu när vintern har kommit, snön ligger tjock i Uppsala och glöggångorna redan fläktat oss i näsan i flera veckors tid känns det som att det är dags för julpyntet att komma upp här i Uptime. Det har varit ett pressat och utmattande år för Update 2023, och undertecknad är redo för en välbehövlig vintersemester. Det ska bara ut ett julnummer av den här blaskan först.

I september invigde vi till sist vår utställning av mini- och mikrodatorer, som fick många besökare på Kulturnatten. Vi deltog i Engagemangsmässan, och efter en tids utmattning bland de aktiva medlemmarna återupptog vi också sponsjakten, om än med blygsam framgång. En större donation gav oss dock lite andrum, och vi har nu tillfällig finansiering till slutet av 2024. Spelkvällar anordnades vid flera tillfällen, regelbundna sweatshops upptogs, ett guldstjärnesystem för utfört föreningsarbete infördes, månadens dator och föreläsningsserien Updateringar genomfördes de flesta månader, och föreningen firade 40 år med en middag i den egna lokalen. Många nya medlemmar har också bidragit till såväl verksamheten som gemytet på Svartbäcksgatan 65. Ett händelserikt år sjunger på sista versen. Hejdå 2023.



Efter jul och nyår samlas vi igen med ny energi och nya ideer, och vi hoppas att 2024 ska få se oss komma till en lösning av vårt finansiella besvär så att vi kan lägga mer tid på datorerna.

Vill du komma i kontakt med Update kan du skriva till styrelsen@dfupdate.se eller till vår allmänna diskussionslista update@lists.dfupdate.se. Du är också välkommen att besöka vår IRC-kanal [#update](#) på EFnet eller lokalen på Svartbäcksgatan 65 i Uppsala. Uptime nås på uptime@dfupdate.se.

Närmast planerade händelser

Inte så jämrans mycket såhär i andra halvan av december.

Nyheter

- Höstmöte hölls och avgiften dubblerades: 480kr minimiavgift, och rekommenderad avgift 2040kr.
- En stor donation av verktyg och instrument mottogs från SM5DEH, en av våra nya medlemmar, som skänkte hela sitt elektroniklabb. Mer om detta senare i numret!

November/december månads dator: PDP-12

Update har en PDP-12 som heter Elsa, som vi fick från Neurofysiologiska laboratoriet vid Ulleråkers sjukhus via docent Anders Fredriksson. Den användes där under 1970-talet för bl.a. EEG-analys och annan relaterad forskning, och hamnade sedan i ett oanvänt rum där den stod under 80- och 90-talen. Under dessa år var troligen den enda personen som startade maskinen Anders son Mikael, som körde den lite på skoj omkring 1989-1990. Maskinen var då endast delvis fungerande. När institutionen sedan cirka 2000 skulle flytta och rensade ut gammal utrustning fick Mikael Fredriksson höra att PDP-12:an skulle slängas, och kontaktade då Update, som kom och hämtade den. Det ska också ha funnits ett expansionskabinett, kanske med A/D- och D/A-omvandlare, som dock troligen redan hade slängts när Update kom till platsen.

Elsa stod sedan trasig i den gode hädangångne kebas kontor på Polacksbacken några år, innan den flyttades ner till Updates dåvarande lokal T-salen, vår terminalsal i källaren på hus 1 på ITC. Åren 2012–2013 reparerade undertecknad tillsammans med sadropol och pontus datorn, som saknade en spänning från nätaggregatet och hade en trasig grind på ett av korten i processorn. Därefter var maskinen en huvudattraktion i T-salen under ett antal år, och demonstrerades ofta för besökare, innan den åter gick sönder och blev stående som prydnadsobjekt fram till flytten från Polacksbacken. Vid denna utflytt, eller snarare vid inflytten på Svartbäcksgatan 65, ställdes Elsa som ett objekt i det särskilda utställningsrum som förbereddes där, och där maskinen ännu står nu idag. I november i år blev det dags att försöka återställa datorn i fungerande skick igen, vilket inte har varit helt lätt. Mer om det senare i artikeln.



Elsa är kanske världens mest kända PDP-12, tack vare detta fotografi som representerat modellen på Wikipedia sedan 2008, och som helt eller i del kopierats och använts på många andra håll på nätet.

Modellen PDP-12 introducerades av Digital Equipment Corporation (DEC) 1969, och tillverkades fram till 1972 i totalt omkring 725 exemplar. Några tiotal av dessa finns idag kvar i världen, och Elsa är en av dem som är i bäst skick. PDP-12 avsågs som laboratedator, för att samla in och presentera mätdata samt styra laboratorietrustning.

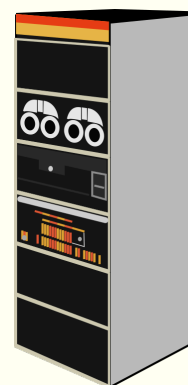
Att döma av en prislista som finns på [Bitsavers](#) är Updates exemplar en PDP-12/30. Det angivna priset för den modellen är 36.900 USD, över 3 miljoner kronor idag. DEC hade dock ett system med rabatter, så många kunder betalade inte hela katalogpriset. Elsa är utrustad med:

- 8K 12-bitars ord primärminne (varav 4K är en utökning för 350.000kr)
- KW12-A: Timer/eventräknare
- AD12: 10-bitars A/D-omvandlare, 16 kanaler varav 8 är rattar och 8 kvartstumsjack
- DR12: 6st datorstyrda reläer som kan slå på och av extern utrustning
- TC12: Bandkontroller av typ LINCtape, med 2st bandare modell TU55
- VC12: LINCscope displaykontroller, med 12 tums bildrörsskärm modell VR12
- Terminalgränssnitt avsett för Teletype, 10 tecken per sekund

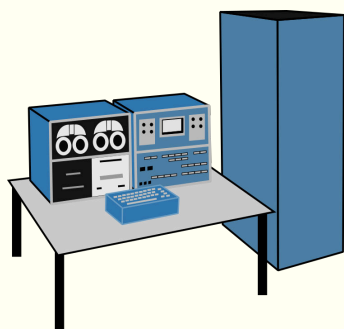
Den billigaste modellen, PDP-12/10, ska ha kostat 15.900 USD och haft 4K minne, ingen bandare, ingen display, och ingen timer. Den dyraste modellen var timesharingsystemet TSS/12, med 16K minne och stöd för upp till 15 terminaler samt lagring på disk, till ett pris av 79.330 USD. Man kunde också köpa till en flyttalsprocessor, FPP12, som kom i ett eget extra kabinett och kostade motsvarande 900.000kr om man ville ha stöd för double precision (endast 650.000kr utan).

Under 1960-talet hade DEC tillverkat diverse olika typer av datorer, huvudsakligen avsedda för mindre företag och andra kunder med mindre budgetar än de då traditionella datorköparna. DEC:s datorer var små så att de fick plats i ett vanligt rum på ett kontor, och tillhörde genren *minidatorer*, som var ny för 60-talet. DEC själva använde dock inte termen "dator" (eller "computer"), utan gav istället alla sina datormodeller namn bestående av förkortningen *PDP*, för Programmed Data Processor, följt av ett nummer: PDP-1, PDP-5, PDP-8, osv.

Deras största succé vid tiden var PDP-8, en minimalistisk dator de introducerat 1965 till det för tiden mycket låga priset 18.000 dollar, motsvarande ungefär 1,8 miljoner kronor i dagens penningvärde. Den ursprungliga modellen, som idag ofta kallas "Straight 8", var tillverkad i diskreta transistorer. 1968 kom en uppföljare som implementerade om arkitekturen i integrerade kretsar (vanlig 7400-logik, faktiskt). Denna modell heter PDP-8/I. PDP-8 utvecklades därefter till en hel familj av kompatibla modeller, som blev mycket populära. Update har en PDP-8/I, vid namn Crap-1, som vi borde laga någon dag.



PDP-8/I



LINC

En annan minidator som fanns vid samma tid var LINC, Laboratory Instrument Computer, som utvecklades vid MIT 1962 och därefter konstruerades av bland andra DEC. LINC hade fördelen att den med lätthet kunde kopplas till laboratorieinstrument för att styra och mäta. DEC bestämde sig för att de ville kombinera LINC:s laborievänlighet med sin egna PDP-8-arkitektur, och på så vis föddes 1966 en dator vid namn LINC-8.

LINC-8 var ett underligt djur. Datorn innehåller både en LINC- och en PDP-8-processor, i samma kabinett, men endast en av dem kan köra åt gången. Datorn startar i PDP-8-läge, och laddar in ett särskilt stödprogram, PROGOFOP, i minnet. PDP-8:an kan sedan närhelst den vill lämna över till LINC-processor, och stoppas då medans denna kör.

Vissa instruktioner och viss I/O på LINC-processor är inte implementerade i hårdvara, utan orsakar traps, som gör att LINC-processor pausas medans PDP-8:an kör en stödrutin i PROGOFOP. LINC-processor kan också lämna över till PDP-8-processor genom att köra en HLT-instruktion. Processorerna delar inga register eller programräknare, utan båda kör sina separata program, som kan pausas för att lämna över till den andra processor för att sedan fortsätta där de var. LINC-datorns minne, som är 2K ord stort, utgör övre hälften av PDP-8-datorns 4K ord stora minne.

Ett par år efter att de introducerat LINC-8 gjorde DEC en omimplementerad och förbättrad version baserad på PDP-8/I, som de kallade PDP-12. Denna modell kombinerar ihop processorerna till en enda, med en gemensam programräknare och övriga register likaledes delade. Denna kan startas antingen i PDP-8-läge eller LINC-läge, och kan växla fritt mellan dem mitt under körande program med två särskilda instruktioner: instruktionen LINC ifrån PDP-8-läget, och instruktionen PDP som fanns i LINC-läget. Dessa växlar endast instruktionsuppsättning, och programmet fortsätter från samma ställe med den andra instruktionsuppsättningen. PROGOFOP finns inte längre, och de två halvorna av arkitekturen är likvärdiga.

Både LINC och PDP-8 är 12-bitars ackumulatormaskiner. De skiljer sig dock mycket åt i instruktionsuppsättning och använder inte alltid samma representation av tal i minnet - i princip hanterar de 12-bitars heltal, men medans PDP-8 använder tvåkomplementsnotation använder LINC en blandning av ettkomplement och tvåkomplement. Minnesadressering är 12 bitar på PDP-8 (4K ord adressrymd), men bara 10 eller 11 bitar på LINC (1K ord programrymd, 2K data). Dock tillhandahåller båda arkitekturerna flera minnesbanker programmet kan växla mellan, för totalt upp till 32K ord minne med maximal expansion.

I/O görs med särskilda I/O-instruktioner. En del av I/O-enheterna kan bara kommas åt i PDP-8-läget, och andra kan bara kommas åt i LINC-läget.

Processorn arbetar vid en klockfrekvens om cirka 625kHz.



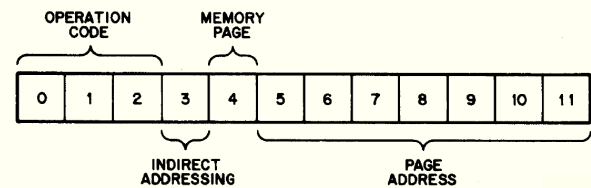
LINC-8 hos Uppsala universitet

PDP-8-arkitekturen

Som sagt är PDP-8 en mycket enkel arkitektur. Det är en ackumulatormaskin, 12 bitar bred, med 12 bitars minnesadressering för totalt 4K 12-bitars ord minne. Detta minne är indelat i 32 stycken sidor om 128 ord. Alla instruktioner är ett enda ord. Instruktioner som gör minnesreferenser kan direkt komma åt två sidor av minnet: sida 0, alltså sidan som utgör adresser 0–127; och *current page*, som är den sida instruktionen själv befinner sig på i minnet.

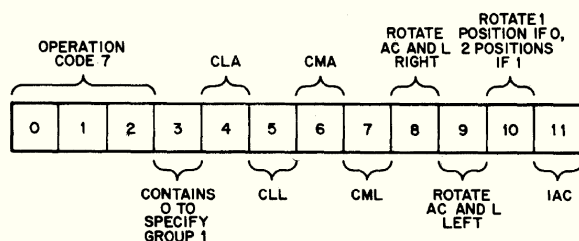
Det finns endast 6 sådana instruktioner, och de följer alla formatet i bilden här nedan. De översta tre bitarna anger opkoden, en bit anger vilken minnessida som ska läsas eller skrivas, sju bitar anger adressen inom sidan, och en bit anger om indirekt adressering ska användas. Om denna bit är noll är den angivna minnescellen själv operanden till instruktionen. Om bitten är en etta innehåller minnescellen istället en 12-bitars pekare till operanden. På så vis kan man komma åt hela minnet. Instruktionerna i denna klass är:

- DCA: deposit and clear accumulator
- TAD: two's complement add to accumulator
- AND: logical AND to accumulator
- ISZ: increment and skip if zero
- JMP: jump
- JMS: jump to subroutine



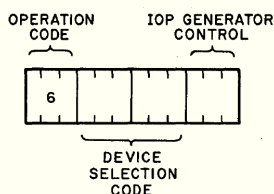
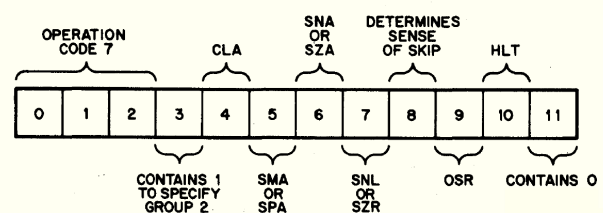
De flesta av dessa är ganska självförklarande. DCA, TAD och AND använder ackumulatorn som den andra operanden. ISZ inkrementerar ett värde i minnet och hoppar över nästa instruktion i programmet om resultatet blir noll. JMP hoppar till den angivna adressen, medan JMS sparar programräknaren vid den angivna adressen och hoppar till adressen efter. På så vis kan man implementera icke-rekursiva subrutiner, genom att varje subrutin har en egen minnesplats för att lagra sin returadress. Datorn har ingen stack.

Vad gäller minnesadressering behöver bara tilläggas att minnescellerna på adress 8–15 har den speciella funktionen att de ifall de används till indirekt adressering automatiskt inkrementeras. De tillhandahåller således ett effektivt sätt att iterera över arrayer och strängar.



Operate II tillhandahåller nollställning av ackumulatorn, skip av nästa instruktion på villkoren att ackumulatorn är positiv, negativ, noll, eller nollskild, samt att link-flaggan är en etta eller en nolla. Flera villkor kan kombineras. Instruktionen tillåter också avläsning av knappraden "Right switches" på kontrollpanelen på datorn, samt halt.

Förutom de 6 minnesinstruktionerna finns två stycken s.k. *operate*-instruktioner, där bittarna i instruktionsordet står för olika operationer som kan slås på och av i valfri kombination. Operate I tillhandahåller nollställning och komplementering av ackumulatorn och carryflaggan (eller *link* som den heter), inkrementering av ackumulatorn, samt rotation till höger eller vänster ett eller två steg av ackumulatorn.



Slutligen har PDP-8 en instruktion som heter IOT, som gör I/O-operationer. Sex bitar i instruktionen väljer en I/O-device, och tre anger en operation specifik till denna device. Devices anslutna till I/O-bussen kan som svar på en I/O-instruktion läsa och skriva ackumulatorn samt signalera till processorn att hoppa över nästa instruktion. En konvention är att varje device har en statusflagga som anger om devicen är redo, och en I/O-instruktion som testar denna och skippar. Växlingen till LINC-läge är implementerad som en IOT-instruktion. Detta är hela PDP-8-arkitekturen —

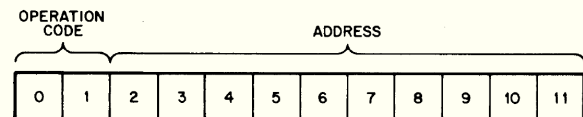
nio instruktioner, fyra instruktionsformat, 12 bitars data och adresser, tvåkomplementaritmetik.

LINC-arkitekturen

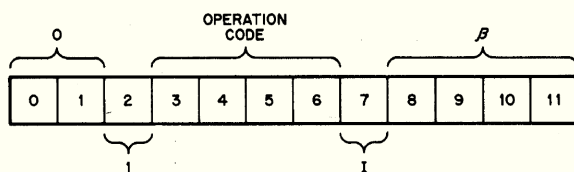
LINC å andra sidan är helt koko. Datorn har en tolv bitars ackumulator precis som PDP-8, men programräknaren är bara 10 bitar, så programmet kan bara vara upp till 1K ord långt. Förutom ackumulatoren finns också ett 12 bitars "multiplier quotient"-register MQ. Tre av instruktionerna har en 10-bitars adress inbakad och kan direkt komma åt ord i detta 1K programfält. Andra instruktioner har istället en 4-bitars adress som anger ett av 16 ord i programfältet som kan innehålla 11-bitars pekare som kan peka till en 2K rymd som utgörs av programfältet (adresser upp till 1K) och datafältet (adresser från 1K till 2K). En del instruktioner är dock byte-adresserade, dvs adresserar ett 6-bitars halvord i minnet, och använder då den översta biten i pekaren för att ange udda eller jämnt halvord.

De tre instruktionerna med en 10 bitars direkt adress är:

- STC: store and clear accumulator
- ADD: add to accumulator (ettkomplement)
- JMP: jump



JMP verkar som en uppenbar instruktion, men är i själva verket en kombinerad jump to subroutine och return — om adressen inte är 0 sparas först programräknaren på adress 0 i programfältet. En subrutin kan alltså inte anropa en annan utan att själv spara undan returadressen någon annan stans. Returadressen sparas för övrigt som en JMP-instruktion, och om man hoppar till adress 0 gör processorn ett undantag och sparar inte programräknaren först. På så vis kan man hoppa tillbaka från en subrutin genom att hoppa till adress 0, som då innehåller en JMP tillbaka till adressen subrutinen anropades från. Notera att instruktionerna STC och ADD är begränsade till samma halva av minnet som programkoden ligger i.



Sedan finns en klass instruktioner som kallas "β-class", som alla använder samma instruktionsformat. Instruktionsordet innehåller en 4 bitars adress som kallas β samt en bit som kallas I. Om β är noll används ordet som följer direkt efter instruktionen som operand. I detta fall står "I" för "Immediate", och anger om det följande ordet är

en immediate eller en pekare till operanden. Om β är något annat än 0 avser det en adress 1–15 i programfältet, som används som en 11 bitars pekare till operanden. I detta fall står "I" för "Index", som anger om den angivna pekaren ska inkrementeras. Dock inkrementeras bara de lägsta 10 bitarna av pekaren. Instruktionerna i β-klassen är:

- ADA: add to accumulator, ettkomplement
- ADM: add to memory, ettkomplement (sparar *både* i ackumulatoren och i minnet)
- LAM: link add to memory, **TVÅ**komplement (adderar med carry, skriver i både AC och minnet)
- LDA: load accumulator
- STA: store accumulator
- MUL: multiply (Gör antingen heltalsmultiplikation eller bråkmultiplikation beroende på högsta biten i β-minnesplatsen. Endast heltalsmultiplikation kan göras med immediates. Multiplikationen är 12x12 bitar, men resultatet är inte 24 bitar, trots att både AC och MQ-registret används: heltalresultat är 11 bitar plus teckenbit, och resten av bitarna slängs bort; bråkresultat är de övre 12 bitarna i AC och absolutvärdet av de nedersta 11 bitarna i MQ. Ettkomplementaritmetik.)
- BCL: bit clear (AND NOT)
- BSE: bit set (OR)
- BCO: bit complement (XOR)
- SAE: skip if accumulator equal
- SRO: skip and rotate (roterar ett steg till höger och skippar nästa instruktion om resultatet är jämnt)
- SHD: skip if half differs (jämför halvord)
- LDH: load half (laddar nedre halvan av AC med ett halvord)
- STH: store half (tvärtom ovan)
- DSC: display character (blinker till en 2x6 pixlars "character" på skärmen ett kort ögonblick. Operanden anger pixlarna, Y-koordinaten ges i ackumulatoren, X-koordinaten ges i adress 1 i minnet)

Sen finns det ett antal instruktioner som manualen presenterar som om de vore en klass, den så kallade α -klassen, som alla har samma format (se bilden intill). Så är dock inte fallet, utan det rör sig om tre instruktioner med gemensamt format och tre instruktioner som allihop har olika format.

Instruktionerna med gemensamt format är skift- och rotationsinstruktionerna:

- ROL: rotate left (α anger antal steg, $I=1$ för att ta med link)
- ROR: rotate right (format som ovan)
- SCR: scale right (skiftar höger aritmetiskt, format som ovan)

Sedan kommer undantagen:

- XSK: index and skip (om $I=1$ inkrementeras minnesplats α i 10-bitars tvåkomplement; om alla de nedre tio bitarna är ettor, skippas nästa instruktion)
- DIS: display (blinker till en punkt på skärmen; Y-koordinaten anges i ackumulatorn, X-koordinaten i minnesplats α . Om $I=1$ inkrementeras X-koordinaten i 10-bitars tvåkomplement)
- SNS: skip if sense switch (tar ett nummer 0–5 i α -fältet, läser av denna knapp på kontrollpanelen)
- SXL: skip if external level (α anger en "external level"-anslutning som testas; dessa inkluderar status för tangentbord och bandstation)

Sedan kommer en instruktion som är två ord, där det andra är källoperanden:

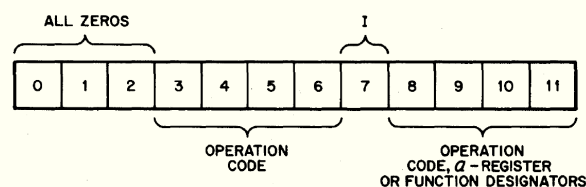
- SET: set α -register (Sparar ett värde i minnet på adress α . I -bitten anger immediate eller absolut adressering för källoperanden.)

Därtill finns ett antal skipinstruktioner som allihop betar sig fel om nästa instruktion råkar vara två ord lång, samt ytterligare lite diverse instruktioner utan operander:

- APO: skip if accumulator positive
- AZE: skip if accumulator zero
- LZE: skip if link zero
- QLZ: skip if bottom bit of MQ zero
- FLO: skip if overflow
- SKP: skip always
- IBZ: skip if LINCtape inter-block zone
- HLT: halt
- CLR: clear (nollställer ackumulatorn, MQ och link)
- COM: complement accumulator
- NOP: no operation
- QAC: place MQ in accumulator (övre 11 bitarna)
- LSW: left switches (läser kontrollpanelen)
- RSW: right switches (dito)
- PDP: växlar till PDP-8-läge

Sedan finns DJR-instruktionen, disable JMP return, som stänger av sparandet av returadress för en följande JMP-instruktion.

Och så ESF, enable special functions, som slår på och av diverse blandade features: trap för odefinierade opkoder, trap för bandemulering, storleken på tecken på skärmen, samplingsläge för A/D-omvandlaren, TTY-interrupt, och "I/O preset", som resettar I/O-enheter inklusive övriga special functions)



Utöver detta finns på PDP-12 även instruktionen IOB, som tar en opkod för en IOT-instruktion och kör den. Användbart för att kunna prata med enheter på PDP-8-bussen från ett LINC-program utan att behöva växla fram och tillbaka mellan lägena; och så finns också instruktioner för att läsa och skriva program- och datafälsregistren, för att kunna välja vilken del av minnet man kommer åt för tillfället. I PDP-8-läget är dessa register tre bitar (eftersom minnet är 4K), och på LINC (där program- respektive datafältet är 1K vardera) är registren fem bitar.

Nå, nu när vi har sett på vilka sätt PDP-8 och LINC skiljer sig ska vi ta och titta på hårdvaran i PDP-12. Dett finns flera användbara I/O-enheter inkluderade:

AD12

Detta är en 10 bitars A/D-omvandlare med 16 kanaler. 8 av dessa är kopplade till kvartstumsjack på en panel bakom den gröna luckan till vänster på datorns framsida. De andra 8 är kopplade till 8 rattar. Till jacken kan man koppla analoga signaler från extern utrustning, upp till $\pm 1V$. Dessa 16 inputtar samplas med LINC-instruktionen SAM.

KW12-A

Detta är en timer/eventräknare som kan räkna pulser på tre inputjack, generera interrupter vid givna intervall, och trigga sampling på AD12. Kontroller för att välja polaritet och tröskelvärde för de tre inputtarna finns på en panel bakom den gröna luckan.



VR12

Detta är en bildskärm på vilken programmet kan visa kurvor, text och enkel grafik. Upplösningen är 512x512 pixlar, men det finns inget skärminne (kom ihåg att hela datorns minne i grundutförande endast är 4K ord, och 4K till kostade 350.000kr), utan

programmet ansvarar för att upprepat tända de punkter som ska visas på skärmen med tillräcklig hastighet för att de inte ska flimra. I texteditorn i systemprogrammet LAP6-DIAL används en av rattarna på AD12-panelen för att välja hur många rader text som ska visas, så att man själv kan välja accepterbar mängd flimmar. Skärmen har två "kanaler", som kan visas var för sig eller båda tillsammans, och en extern skärm kan också anslutas via en kontakt på en panel bakom luckan, så att man t.ex. kan visa den ena kanalen på den inbyggda skärmen och den andra kanalen på den externa.



DR12

Six stycken reläer med anslutningar på en panel bakom luckan, som kan styras från programmet för att slå på och av extern utrustning.

Det finns också på panelen två rattar för att ställa in "auto restart delay", som är en feature som används för att automatiskt stega igenom ett program via single step- och breakpoint-funktionerna på kontrollpanelen. Det finns också en inbyggd högtalare, med en volymkontroll på panelen. Högtalaren är kopplad till högsta biten i ackumulatören, och kan användas för att höra vad datorn har för sig eller, via försiktig programmering, möjligen för att ge ifrån sig ljud för att signalera till användaren.

Längst ner på panelen finns också nyckeln som slår på datorn. Den har också ett "lock"-läge, som stänger av kontrollpanelen så att man inte råkar komma åt någon knapp och sabba körningen.





LINCtape

Lagring av data och program på PDP-12 (och LINC) sker huvudsakligen på små rullband, som heter LINCtape. Bandrullarna är 10cm i diameter och kommer i runda dosor i grön färg. Ett snarlikt medium vid namn DECTape användes också av DEC, men trots att banden ser identiska ut är de inte kompatibla. DECTape brukar förvaras i dosor i blå färg, förutom om man äger band av båda typerna, då de förvaras i dosor av slumpmässig färg.

Ett LINCtape-band lagrar 512 block om 256 12-bitars ord vardera, totalt 128K ord per band. Bandenheten kan själv söka ett block givet dess nummer, spolande bandet i den riktning som behövs, och kan sedan inte bara läsa block utan också skriva över block in-place. Detta kan inte alla andra typer av bandenheter och band göra. Alla operationer med LINCtape utförs helt autonomt av bandenheten, och processorn behöver bara köra en I/O-instruktion som ber om ett visst block för att få detta hittat och läst direkt in i minnet. Mjukvara på PDP-12 använde LINCtape både som filsystem och som swaputrymme.

Data är lagrade på LINCtape i form av tio parallella spår, som läses av tio huvuden. Dessa utgör två redundanta grupper om fem spår sida vid sida, varav tre är data, ett är klocka, och ett är "mark", eller metadata. Bandenheten kompenserar för variationer i rotationshastigheten på bandet med hjälp av klockspåret, och använder mark-spåret för att hitta början och slutet på block samt blocknummer och checksum. Eftersom alla data är lagrade redundant två gånger bredvid varann på bandet kan bandet skadas ganska mycket innan det slutar vara läsligt.

Bilden nedan till höger visar ett litet avsnitt av band, med klockspåren ytterst, mark-spåren innanför, och de tre databittarna i mitten. Eftersom varje sådan "skiva" tvärs över bandet innehåller tre databitar krävs fyra skivor för att representera ett 12-bitars ord. Detta ger utrymme för fyra bitar mark-kod för varje ord, som anger för bandenheten vad detta ord är för typ av information. Dessa typer är:

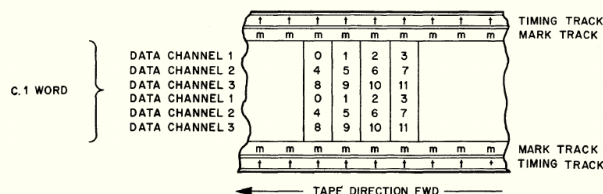
BM block number, 0-511

GM guard word, för att hinna slå på/av skrivning

DM/FM data word/final word, markerar data-ord

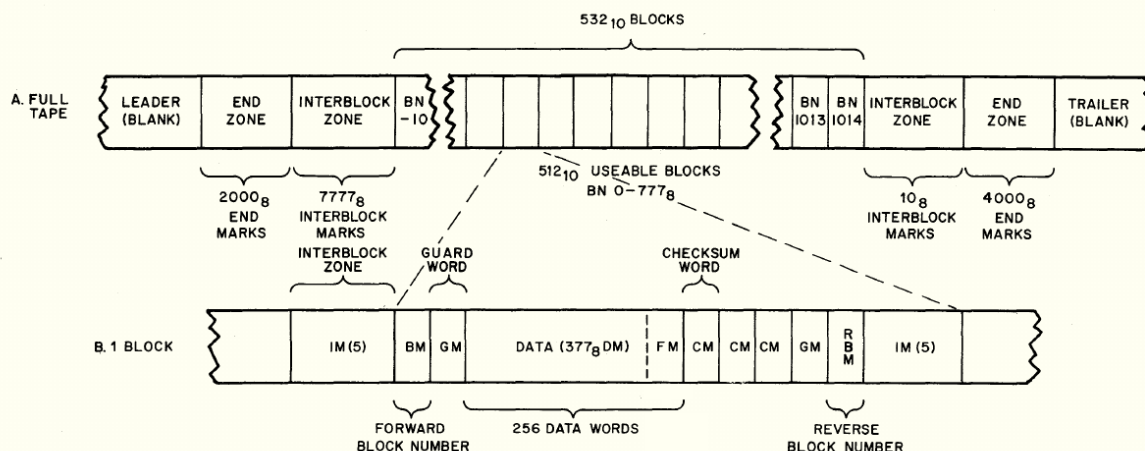
CM checksum, inkl två dummy-ord för timing

RBM reverse block number



Formatet på ett helt band är som i den övre remsan i bilden nedan: ändarna av bandet

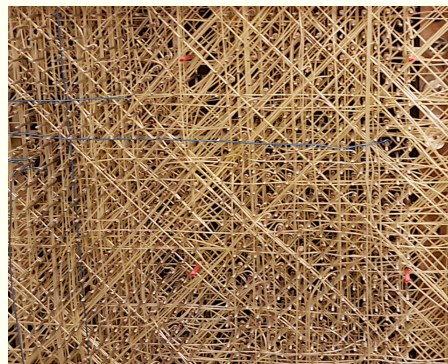
markeras med end zones och interblock zones, och däremellan kommer **fastan** de 512 datablocken. Den nedre remsan visar layouten för ett enda block, som avgränsas av interblock zones och forward respektive reverse block numbers, detta för att sökning ska kunna ske med bandet spolande antingen framåt eller bakåt. Där innanför finns så data och checksum.



Kontrollern i PDP-12 heter TC12, och de två bandenhetererna TU55. Det går att ansluta ytterligare sex TU55. Det finns också en option man kunde köpa till för att kunna läsa och skriva DECTape på samma bandare. Från kontrollpanelen på TU55 kan bandet spolas med pilknapparna om bandaren är ställd i "local"-läge, medans "remote"-läget innebär datorstyrning. Skrivskydd kan också slås på.



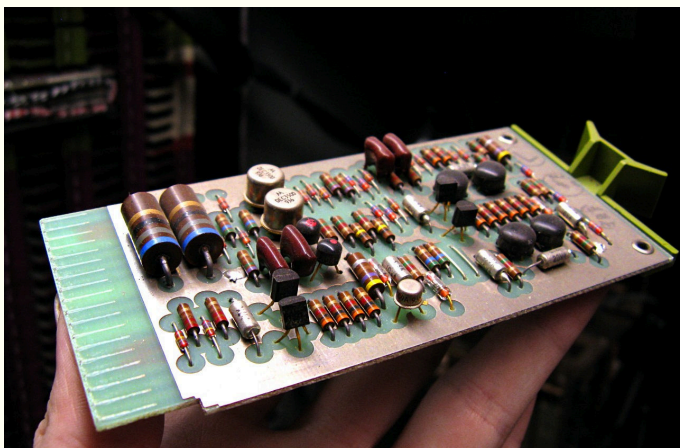
Om vi öppnar dörren på baksidan av kabinettet kan vi se hur datorn är konstruerad. I dörren hänger två stora bakplan fulla med små kretskortsmoduler av typen *Flip Chip*, av vilka all logiken är konstruerad. Dessa kretskort sitter istuckna i kontaktdon på sidan av dörren som är inåt i datorn (mot oss i bilden). Synliga i bilden är huvudsakligen de färgglada plasthandtagen på korten, som gör det möjligt att ta ur och sätta i korten, och på vilka typnumret för varje kort står tryckt. DEC tillverkade Flip Chips för ett antal olika standardfunktioner, såsom grindar och flip-floppar, och byggde sina datorer av dessa. I datorn finns alltså ett stort antal moduler av samma typ, med samma nummer.

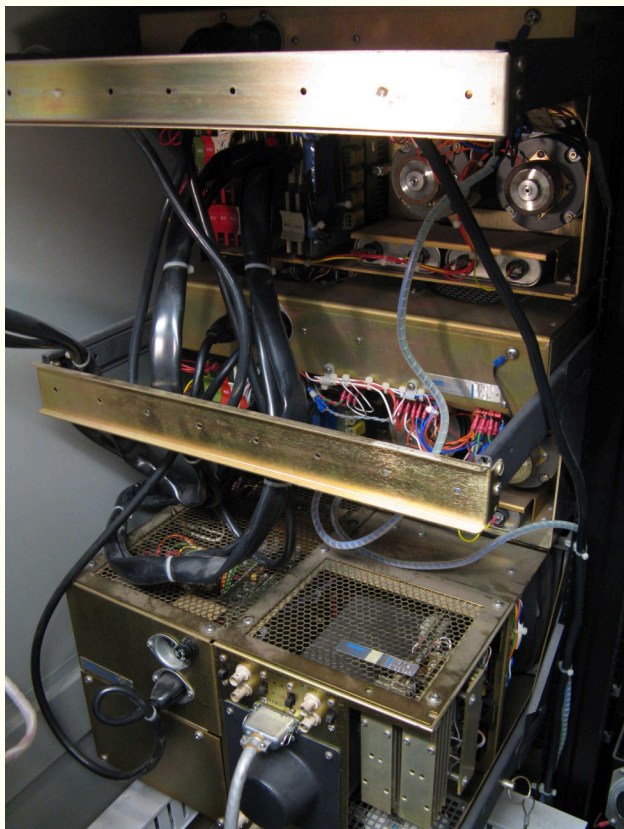


På baksidan av bakplanen är alla kortplatserna sammankopplade med varann via wire-wrap, dvs trådar som är virade runt kontaktpinnar. Lödningar används inte. Det går att hänga på oscilloskopprober på pinnarna för att felsöka kretsarna, men själva bakplanet vill man nog helst inte felsöka...

De lila korten som nästa hela datorn är byggd av är TTL-logik, de gröna är analogkretsar för kärnminnet, som är de två gråaktiga klossarna som syns i övre delen av det översta bakplanet. A/D-omvandlarna och displayen är de orangea korten, och de lila korten i övre bakplanet utgör bandkontrollern och annan I/O, medans nedre bakplanet innehåller processorn. Vita kort är kontakter för kablage etc.

En del av korten är nästan som små konstverk. ⇒





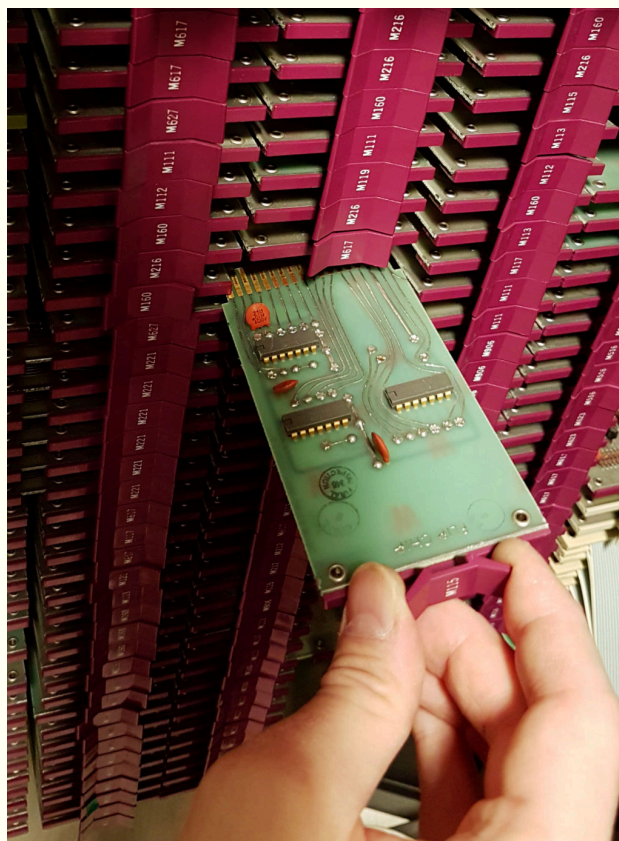
Jag upptäckte förresten ett perfekt verktyg för att ta loss LINCtape-band från bandaren. Det är i stort sett omöjligt att få av dem med bara händerna, de sitter liksom påtryckta med friktion på navet och man har ingenting att ta något ordentligt grepp om. Men det funkar att bara lyfta under kanten med en rackfot:



Tittar man sedan inuti kabinettet ser man överst de två TU55-bandenheterna, och under dessa VR12-skärmen. Dessa är vanliga 19-tumsenheter och sitter på skenor så de kan dras ut på framsidan av datorn. Racket består liksom av ett 19-tumsrack med en extra bit på ena sidan för den gröna luckan med panelerna innanför.

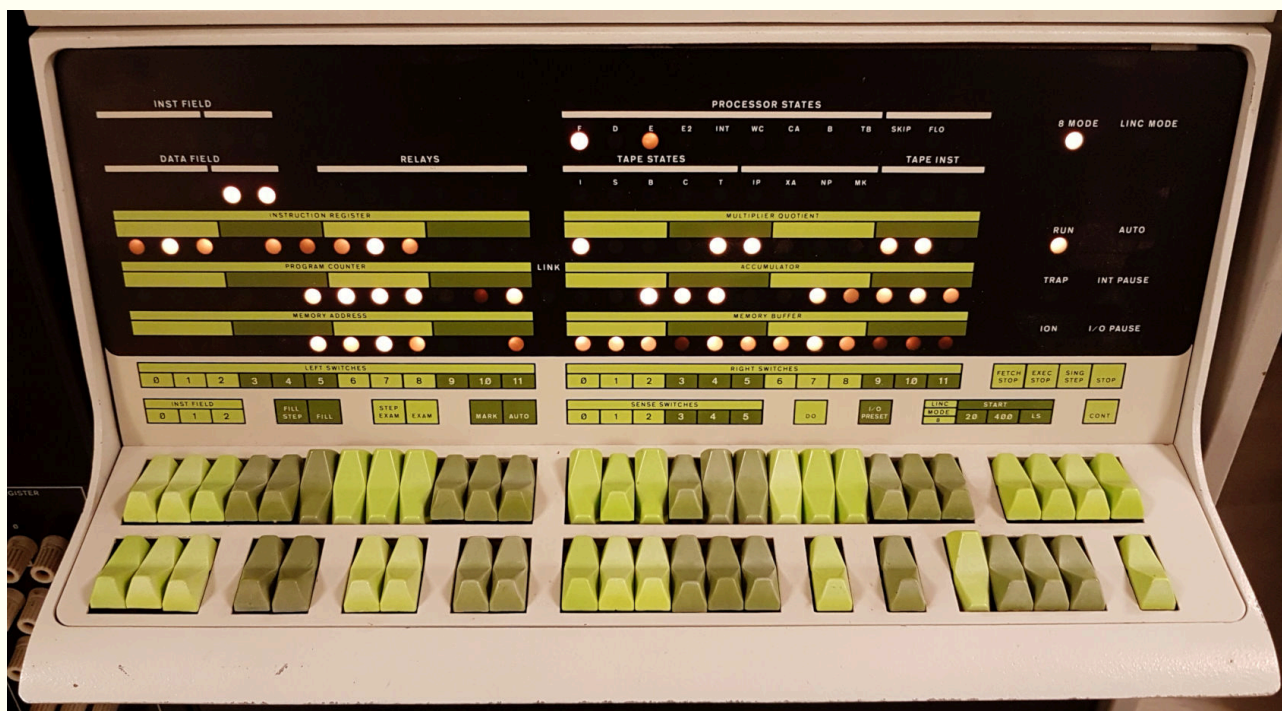
PDP-12 har förresten en I/O-buss av PDP-8/I-typ virad nånstans på bakplanet, och kan använda periferienheter avsedda för PDP-8/I, t.ex. plotter, remsläsare, bandare, hålkortsläsare, diskenheter, skrivare, osv. Denna buss har stöd för DMA, som på PDP-8-språk heter *data break*. En multiplexeroption behövs om man vill ha fler än en device som använder DMA.

Bakplanet är förvirrat för: EAE-utökning KE12, timer KW12-A/B/C, plotter XY12, och TTY-gränssnitt DP12-A. EAE-utökningen ger PDP-8-processorn stöd för multiplikation och division m.m. Det finns också en expansion för vektoriserade interrupts med automatiskt sparning av registren, som heter KF12B, och en expansion för att ansluta fler I/O-enheter vid namn BA12. Det finns också som tidigare nämnt en flyttalsexpansion, som heter FPP12.



Så ser det ut när man drar ut ett kort. Man kan behöva vicka en hel del för att få tillbaka dem. Urtagande och istoppande av kort har varit ett konstant tema under felsökningsarbetet den senaste månaden.

Kontrollpanelen på datorns framsida ser ut på följande sätt:



Från denna panel kan man se processorns tillstånd, läsa och skriva minnet, stega program, med mera. Överst till vänster finns indikatorer för Instruction Field och Data Field-registren, och bredvid dessa indikatorer för reläerna. Tända lampor är ettor och släckta nollor. Överst i mitten finns så alla processor-states: (F)etch, (D)efer, (E)ecute och (E)ecute (2) är cyklerna en instruktion går igenom under körning. SKIP visar skip-flaggan som anger om nästa instruktion kommer att hoppas över. Vidare finns indikatorer för bandkontrollerna, interrupt, DMA, etc.

Under de gröna remsorna visas innehållet i de olika registren: programräknaren, ackumulatoren och MQ, samt de interna registren: instruktionsregistret, minnesbufferten och minnesadressregistret.

De senare två registren används också tillsammans med de två raderna om 12 knappar på den nedre delen av panelen: Left Switches används för att mata in en minnesadress, och Right Switches för att mata in data. Fill- och Exam-knapparna skriver resp. läser minnet på den angivna adressen. Fill Step och Step Exam även inkrementerar adressregistret. Do-knappen tillåter körning av en instruktion direkt från Left Switches. I/O preset resettar datorn och ställer den i det läge som den ljusgröna Mode-knappen anger: LINC eller 8. Start-knapparna startar datorn från en av adresserna 20 och 400, eller från adressen på Left Switches. Fetch Stop och Exec Stop används för att automatiskt stoppa datorn när den gör en minnesåtkomst från den angivna adressen, och Stop och Sing Step tillsammans med Cont används för att stoppa samt instruktion- och sykelstega processorn.

Mer komplett dokumentation finns på Bitsavers i [referensmanualen](#) och [ritningarna](#). Dessa har varit ovärderliga när vi har genomfört...

Underhåll och reparationer

Datorn var alltså inte i fungerande skick sedan några år tillbaka. Den startar, kan ladda in programkod från band, men spårar sedan mer eller mindre omedelbart ur vid körning. Under november (och december) månad har vi försökt felsöka detta beteende för att hitta vilket eller vilka kort i datorn som kan tänkas vara trasiga, och reparera dessa.

Först av allt ville vi dock göra lite underhåll på kontrollpanelen, vars glappiga kontakter gick oss på nerverna varje gång vi skulle testköra någonting. Så vi började med att plocka loss panelen, plocka isär den, avlägsna så mycket smuts och damm som möjligt med



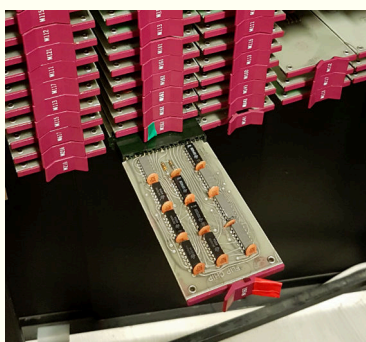
Kontrollpanelen loss



Det faktum att vi har kompletta kopplingsscheman för datorn på papper (i jätteformat!) är enormt hjälpsamt vid felsökning, vanligtvis, men i det här fallet gick vi bet. Hur vi än kliade oss i huvudet kunde vi inte komma fram till vad som kunde orsaka felet vi observerade.

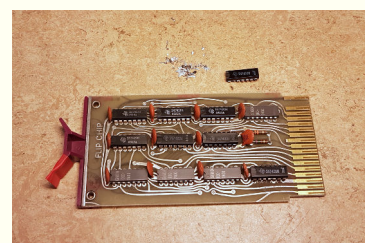


Nåväl, undertecknad började helt enkelt att byta plats på likadana kort — om felet ändrar sig har man flyttat det trasiga kortet. Efter att ha hittat ett suspekt kort använde jag mig av det extender-kort vi har för att dra ut kortet så att jag kunde titta på signalerna på det medans datorn körde.



Extender

Ett 7400-chip visade sig outputta ogiltiga logiknivåer, så detta byttes ut. Komponenterna är ju vanlig 7400-logik som fortfarande finns att få tag på. Denna reparation hjälpte lite grann, och datorn kunde nu köra instruktioner korrekt, men det var fortfarande nånting fel. Jag fortsatte att flytta runt kort, och upptäckte att samma kort också (eller kanske endast?) hade några dåliga lödningar. Efter att jag lödit om dem kunde datorn boota LAP6, men visade felaktig bild på skärmen. Jag flyttade runt fler kort, och har nu flyttat runt alla kort i hela CPU:n, utan att hitta felet. Nu fungerar datorn



Utbytt 7400

inte alls igen, och bootar över huvud taget inte. Jag tog ingen bild medans den fungerade lite grann, så ni får inte veta hur LAP6 ser ut. Och nu är tiden slut och jag åker utomlands, så det blir ingen upplösning på mysteriet i det här numret tyvärr. Fortsättning följer i januari.

tops, och sedan spraya alla strömbrytare med kontaktrengöringsspray och vicka dem fram och tillbaka en stund, och sedan spraya med kontaktsmörjning. Detta tog vol, zeltophil och mig en god stund, men det var värt besväret och panelen fungerar nu utmärkt, vilket har gjort resten mycket bekvämare.

Efter att vi fixat panelen började vi på att testköra instruktioner i minnet för att se om vi kunde hitta någon som fungerade. Det gjorde vi omgående, och under cykelstegning såg vi att instruktionerna inkrementerade programräknaren två gånger under körningen. Det var alltså två cykler som triggade laddning av programräknaren, istället för bara en. Vi plockade fram ritningarna och började leta efter vilka kretsar det kunde vara som orsakar denna extra programräknarladdning.



Apptajm: pahole

Den här gången har undertecknad brist på tid, så det blir en ganska kort Apptajm detta nummer, men jag har i alla fall dragit mig till minnes ett program jag hittade en gång för några år sen och som jag hade användning för då och som jag tänkte jag kunde presentera helt kort för er alla. Detta program heter pahole — Poke-a-Hole.

Detta verktyg läser debug-informationen i en objektfil och listar ut minneslayouten för structar. Detta kan vara riktigt användbart om man behöver skriva portabel assemblerkod (vilket var det jag höll på med), eller om man försöker göra bindningar för något bibliotek och behöver veta hur dess datastrukturer är utformade i minnet. Det kan också vara användbart för att debugga ens egna C-program, för att se t.ex. var paddingen har hamnat i en strukt som har annan storlek än man förväntar sig att den borde ha.

Så en typisk sak man kan göra är att inkludera några rader i sin Makefil (eller i ett skript) som ekar namnet på en header och en deklaration av en struktvariabel till en fil, kompilarar denna med -g för att inkludera debuginformation i objektfilen, och sedan kör pahole på objektfilen och gör nånting intelligent med outputten, t.ex. generera assemblerdefinitioner för offseten till medlemmarna i strukten och stoppa dem i en includefil.

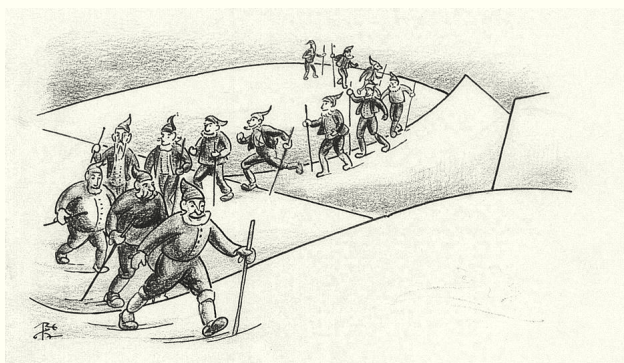


Pahole kommer med julklappar till barnen

Exempel:

Här använder jag en C-fil som deklarerar en variabel av typen struct sockaddr, som används för nätverkssockets.

```
Philemon:~/temp>cat test.c
#include<sys/socket.h>
struct sockaddr sa;
Philemon:~/temp>gcc -g -c test.c
Philemon:~/temp>pahole test.o
```



Och här kommer de tretton jultomtarna för att stjäla mat och slå i dörrarna

Och resultatet som kommer ut ur Pahole visar hur strukten ser ut, i C-syntax med en kommentar på varje rad som anger offset och storlek för varje medlem i strukten:

```
struct sockaddr {
    sa_family_t      sa_family;          /*      0      2 */
    char             sa_data[14];        /*      2     14 */

    /* size: 16, cachelines: 1, members: 2 */
    /* last cacheline: 16 bytes */
};
```

Det finns en massa parametrar man kan använda, till exempel en som automatiskt stavar om strukten för att få bort padding. Så om det verkar användbart för dig, läs mansidan!

Updates nya elektroniklabb

En av de roligaste sakerna som inträffat den senaste tiden är att en av Updates nyaste medlemmar, SM5DEH, bestämt sig för att skänka hela sitt hemmalabb till föreningen. Update har således nu ett digitalt oscilloskop, ett fickoscilloskop, tre stycken bänknättagg, två lödstationer (en sedan tidigare), två logikanalysatorer, plus en stor mängd olika handverktyg, mätinstrument, tvingar, skruvstycken, etc.

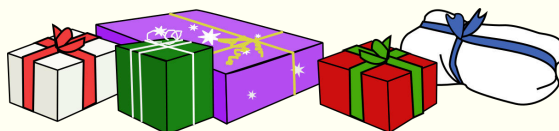


Vi har sedan tidigare ett enklare analogt oscilloskop (i visst behov av service), men saknade helt bänknättagg och logikanalysator, och uppsättningen verktyg var mycket begränsad.

Utöver verktyg och instrument har vi också fått en mängd komponenter, däribland GPS- och RFID-moduler och RFID-taggar, IR-sensorer, antenner, ett antal Raspberry Pi-kort av olika typ, kameror, ytmonterade passiva komponenter, med mera med mera. Även en liten hyllmöbel följde med.

En flytt fick donatorn att omprioritera sina hobbyintressen, och tack vare hans stora givmildhet har vi nu avsevärt bättre möjligheter till underhåll och reparationer av våra datorer utan att det har kostat oss någonting. Vi har redan ställt upp ett labbord i L-salen och börjat ställa iordning med instrument och verktyg, och påbörjat tillverkandet av en enkel verktygstavla för några av de mer frekvent använda handverktygen. Några timmars städande och omstuvande har också gjort plats i hyllorna för förvaring av de övriga verktygen. Mer arbete samt förvaringskärl krävs för att få ordning på alla sakerna.

Stort tack till SM5DEH, alias Nils Söderman, för hans julkapp till föreningen!





Bland de verktyg vi inte har sedan tidigare syns gängtappar, tvingar, filar, skalpeller, plåtsax, böjlig bitsnyckelhals, peanger, specialbits, spårtyttartänger, figursåg samt en liten geringslåda. Sortimentet tänger och skruvmejslar har också avsevärt utökats, samt även sortimentet av skruv (ej i bild), och vi har nu också ett 50m måttband.



Donationen omfattade ävenledes experimentkort för olika radiotelemetri-, RFID- och GPS-tillämpningar, Raspberry Pi-kort flera nya i förpackning, ytmonterade kondensatorer och motstånd, samt diverse olika pinaler inköpta för olika projekt: solid-state-relän, luftfuktighetssensorer, IR-sensorer, USB-uartar, kameror, SD-kort, en TFT-panel, antenner, osv. Kort sagt en mängd delar användbara för projekt även hos Update.

Det var allt för Uptime för den här gången. Nu återstår bara att önska eder alla

EN FRÖJDEFULL JUL!

