

# DASK - en 1. generations computer

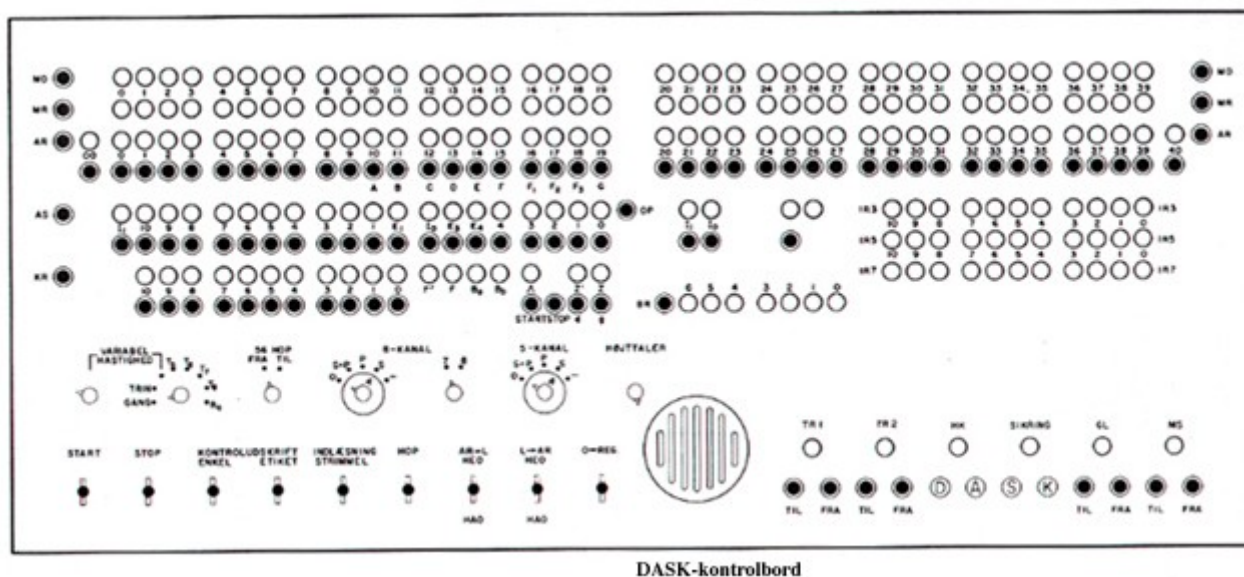
af

H. B. Hansen

2. Del

## Den logiske struktur af Dask

Den logiske struktur af Dask kan i det store og hele aflæses ved at se på kontrolbordet.



De små cirkler er lamper, en for hver binær position i registrene, som er de lange rækker af lamper. De sorte klatter under lamperne og til højre og venstre er trykknapper, ved hjælp af hvilke man kunne ændre indholdet i visse af registrene. Hvis man trykkede på knapperne til højre eller venstre for et register, blev hele registeret nulstillet. Man kunne herefter trykke på knapperne under registeret, hvorved de positioner, knappen er under, bliver 1-stillet. Når en lampe lyste (med et mærkeligt blegrødt lys) betød det at indholdet i vedkommende position var 1. Når Dask kørte blinkede alle lamper lystigt, stundom i ejendommelige og fascinerende mønstre der afspejlede løkkestrukturen i den algoritme der blev benyttet i programmet. Den store runde cirkel med lodrette riller i nederste, midterste del af billedet er en højttaler, der var koblet til nogle positioner i akkumulatorregisteret. Den gav lyd fra sig når maskinen kørte. Hvert program havde sin egen karakteristiske "melodi", som man lærte at kende i testfasen, og som kunne benyttes til at kontrollere om programmet kørte korrekt - fx hvis man blev træt af at sidde ved kontrolbordet i den lange nat, og tog sig et hvil på en divan der stod i dagligstuen.

Ferritlageret på Dask var på 1024 helceller (hec) med hver 40 binære positioner, men det kunne også anskues som 2048 halvceller (hac) med hver 20 positioner. Helcellerne havde de lige adresser, og halvcellerne var nummereret fra 0 til 2047; hveranden halvcelle havde således samme adresse som en helcelle. Ved hjælp af operationsangivelsen i en ordre kunne man bestemme om en lige adresse skulle forstås som en hec eller en hac. En ordre fyldte en hac, så der var to ordrer pr. hec.

Dask regnede med ægte brøker, dvs. alle maskintal opfattedes som liggende i intervallet  $-1 \leq \text{tal} < 1$ ,

hvilket blev kaldt "Dask-intervallet". Position 0 (længst til venstre i en celle eller et register) var fortegnet, og resten var binalerne. Der anvendtes 2-komplementaritmetik. Et tal i en hac havde også sit fortegn længst til venstre, altså midt i en hec hvis hac'en havde ulige adresse. Et maskintal bestod således af et fortegn og 19 eller 39 binaler. Multiplikation og division var dog specielle, idet de kunne operere på hvad man kaldte "den lange akkumulator" som var to registre sat i forlængelse af hinanden (se nedenfor). Dette gav en præcision på 78 binaler (fortegnspositionen i det ene register blev ikke brugt). Dask var derfor, også efter moderne forhold, en maskine med meget stor præcision.

Der var ikke indbygget flydende regning i Dask, men et af de første biblioteksprogrammer der blev fremstillet, var programmeret flydende regning. Det gik jo lidt langsomt, og derfor gjorde man sig store anstrengelser for at holde sig indenfor Dask-intervallet i praktisk programmering. Udformning af programmer til Dask hed forøvrigt ikke "programmering" men "kodning".

Et register med tilhørende knapper på kontrolbordet, som det vises på ovenstående billede, vil jeg i det følgende kalde "en række".

Der var følgende registre i Dask:

- **Multiplikandregisteret** MD (den øverste række på billedet). Her stod multiplikanden ved en multiplikation. Man kunne kun nulstille MD manuelt. Koderen kunne ikke eksplicit fylde noget i MD, men når der blev udført en multiplikation blev indholdet af AR (se nedenfor) kopieret over i MD lige inden multiplikationen.
- **Multiplikatorregisteret** MR (den anden række på billedet). Dette register skulle indeholde multiplikatoren inden en multiplikation. MR kunne manuelt nulstilles. Der var adskillige ordrer der afleverede deres beregningsresultat i MR, klar til multiplikation. Ved division fremkom kvotienten i MR.
- **Akkumulatorregisteret** AR (den tredje række på billedet). Dette var langt det vigtigste register i forbindelse med beregninger. Man kunne nulstille og indsætte et ønsket indhold manuelt i AR. Bemærk at AR har en ekstra position, position 00, helt til venstre; den anvendes til at undersøge om der er spild (det kaldte man overflow dengang). Hvis position 0 og position 00 har forskelligt indhold er der spild.
- **Lang akkumulator** ARMR. AR og MR sat i forlængelse af hinanden udgjorde den lange akkumulator. Fortegnet var position 0 i AR, og position 0 i MR blev ikke brugt. Ved en lang multiplikation fremkom produktet af to 40-bits tal i ARMR; og ved division skulle tælleren stå i ARMR, og kvotienten blev da 40 bit lang. Derfor kunne man ved skiftevis at udføre multiplikation og division opnå meget stor nøjagtighed i beregningerne.
- **Adresseregisteret** AS (dette register befinder sig i fjerde række på billedet og består af 11 binære positioner, nemlig nr. 2 til 12 fra venstre). Dette register indeholdt adressen i den ordre der var ved at blive udført (den aktuelle ordre). De 11 positioner er netop tilstrækkeligt til adressering af de 2048 hac i ferritlageret.
- **Operationsregisteret** OP (dette register befinder sig til højre for AS på billedet, nemlig positionerne nr. 14 til 20). Her finder vi operationskoden for aktuelle ordre. De 7 binære positioner giver mulighed for 128 forskellige ordrer, og de blev næsten alle brugt. Der var dog enkelte blindordrer ind imellem, hvoraf nogle senere blev reserveret til bestemte formål i forbindelse med udvidelser af Dask. Et eksempel er sekvenshoppet til det sydede lager (se 1. del af dette skrift).
- **Indeksregistre** IRB, IRC og IRD (det er de tre 11-bits registre under hinanden til højre for AS og OP). Disse registre blev brugt til adressemodifikation af AS-adressen, nemlig sædvanlig indicering som det også kendes i moderne computere. Indeksregisteradressen blev angivet af de to bits i ASOP registeret, der ikke er omtalt i det forrige, nemlig position nr. 1 og 13 på billedet. 00 i disse positioner betød ingen indicering, 01 betød IRB, 10 betød IRC og 11 betød IRD. At disse positioner ikke ligger samlet i ordreformatet har noget at gøre med at Dask skulle være kompatibel

med SMIL i Lund. Oprindeligt var der en omskifter på kontrolbordet ved hjælp af hvilken man kunne slå indeksfunktionen fra og til; hvis den var slået fra var Dask kompatibel med SMIL. Denne mulighed blev dog mig bekendt aldrig udnyttet, og på et vist tidspunkt blev omskifteren fjernet. Graveringen på kontrolbordet kalder forøvrigt indeksregistre for hhv. IR1, IR2 og IR3, men disse betegnelser blev aldrig benyttet i praksis.

- **Kontrolregisteret** KR (den femte række på billedet, de 11 positioner længst til venstre). Dette register indeholdt adressen på den celle i lageret hvorfra aktuelle ordre blev hentet for at blive udført. Adressen på den næste ordre er således almindeligvis 1 større end indholdet af KR (en ordre er 20 bit lang, altså en hac). Som det ses på billedet kunne man modificere indholdet i KR manuelt, hvorved man kunne springe rundt i programmet i testfasen.

De øvrige registre på billedet var til glæde for teknikerne, og sagde ikke koderne noget. Så vidt jeg ved kunne man her følge med i mikroprogrammerne på Dask, og se hvad der skete under afvikling af en enkelt ordre. Mikroprogrammerne var realiseret i hardware med logiske kredse, og Dask var altså hvad man senere kom til at kende som en RISC-computer. Disse tekniske registre vil jeg ikke omtale nærmere her, for det ved jeg ikke meget om.

Derimod kan der være god grund til at omtale det, der på billedet befinder sig under rækkerne med registre. Lige under registre finder vi diverse omskiftere, og nederst på billedet er der nogle vippekontakter til venstre for højttaleren og nogle trykknapper til højre for højttaleren.

Længst til venstre i gruppen af omskiftere befinder sig et par knapper, der tilsammen udgør en slags speeder for Dask. Knap nr. to fra venstre har følgende indstillinger: GANG, TRIN og VARIABEL HASTIGHED, plus fem andre indstillinger. Hvis denne omskifter står på GANG når Dask startes, så spæner maskinen af sted så hurtigt den kan; hvis den står på TRIN bliver der kun udført én ordre, nemlig den der vises i ASOP-registeret; og hvis den står på VARIABEL HASTIGHED, så kan man benytte knappen længst til venstre til at få Dask til at køre langsommere (fra *meget* langsomt til fuld speed). For at få denne speeder aktiveret skulle man så vidt jeg husker hive den ud inden man indstillede hastigheden, og når man trykkede den ind stoppede maskinen igen. Meningen er at man så (måske) kan følge med i hvad der sker når lamperne på kontrolbordet blinker. De fem andre indstillingsmuligheder var til teknisk vedligeholdelse og fejlfinding; her kunne man følge de enkelte faser i mikroprogrammerne på en eller anden måde, som jeg ikke kender til.

Den tredje omskifter fra venstre på billedet af kontrolbordet hedder 56-HOP og har indstillingerne FRA og TIL. I ordrepertoiret for Dask findes der en særlig ordre, 56-hoppet, der smager på indstillingen af denne omskifter. Hvis omskifteren står i positionen FRA virker 56-hoppet som en blindordre, men hvis omskifteren står i positionen TIL virker 56-hoppet som et sekvenshop (se senere). På denne måde kunne man på udvalgte steder i sit program indsætte 56-hop til fx fejludskrifter o.lign., som så kunne springes over ved blot at stille omskifteren på FRA. Nyttens af denne omskifter var tvivlsom; den blev senere overflødiggjort af et meget udbygget kompleks af kontrolprogrammer (debuggere på nudansk). Som tidligere nævnt var der oprindeligt en omskifter til at slå indeksfunktionen til og fra, og uheldigvis sad den lige ved siden af 56-hop omskifteren. Jeg har ved flere lejligheder set febrilske kodere i kampens hede tage fejl af de to omskiftere, men højst overraskende kørselsresultater til følge.

De næste tre omskiftere på billedet af kontrolbordet vedrører valg af ydre enhed for trykning. De to store omskiftere er for hhv. 5-kanals og 8-kanals udskrift. Man kan vælge imellem at få output på det medium der fremgår af den aktuelle trykkeordre (stilling 0), eller på skrivemaskine (stilling S), eller på perforator (stilling P), eller på både skrivemaskine og perforator (stilling S+P). Den lille omskifter imellem de to store er til valg af om 8-kanals udskriften fremkommer på grundlag af 7-bit alfabetet (analogt til ASCII) eller et særligt 8-bit alfabet. Valg af inputmedium skete ved hjælp af en omskifter på strimmellæseren. Valg af lineskriver skete på den måde at hver af de store omskiftere faktisk er delt i to omskiftere oven på hinanden; den nederste (store) omskifter blev brugt til valg af lineskriver. Det var i det hele taget et

meget kompliceret system der krævede en del indlæring af operatøren.

Endelig er der på billedet vist en lille omskifter øverst til venstre for højttaleren; det er højttalerens volumenkontrol.

I den nederste række på billedet af kontrolbordet ses nogle vippekontakter til venstre for højttaleren, og nogle trykknapper til højre for højttaleren. Vippekontakterne er, taget fra venstre mod højre:

- **START.** Dette var startknappen. Den virkede som beskrevet ovenfor i forbindelse med omskifteren for TRIN/GANG.
- **STOP.** Stopknappen.
- De to næste vippekontakter er til forskellige former for kontroludskrifter. De virkede når maskinen var stoppet. **KONTROLUDSKRIFT ENKEL** gav så vidt jeg husker en udskrift af indholdet af registre (på sedecimal form), mens **KONTROLUDSKRIFT ETIKET** i begyndelsen så vidt jeg husker desuden gav en udskrift af ferritlagerets indhold. Efter fremkomsten af Jørn Jensens kontrolprogrammer gav **KONTROLUDSKRIFT ETIKET** udhop til disse. Udskrifterne kom som en strimmel fra perforatoren, med mindre man valgte et andet outputmedium med omskifterne oven over.
- **INDLÆSNING STRIMMEL.** I begyndelsen af computerens historie, før ROM-lagre blev opfundet, var det et problem at få det allerførste program ind i maskinen (operativsystemet er jo, når alt kommer til alt, også et program). For at løse dette problem fremstillede man såkaldte koldstartprogrammer (bootstrap loadere), der er små programmer der får hele systemet til at rejse sig fra næsten ingenting. Disse programmer findes permanent i computere i vore dage, men på Dasks tid, før tromlen blev tilsluttet, havde koldstartprogrammet form af en lille strimmel, der skulle sættes i strimmellæseren. Denne strimmel skulle så læses ind i Dask ved hjælp af en særlig ordre, med operationskode 19 sedecimalt. Virkningen af denne ordre er at der læses 10 sedecimale cifre fra strimlen, som lagres i den helcelle, der står i ordrens adresse. Vippekontakten **INDLÆSNING STRIMMEL** bevirkede at der kom en ordre med adresse 0 i AS og kode 19 i OP. Når man herefter trykkede på **START** blev denne ordre udført, og der blev indlæst 10 sedecimale cifre til celle 0. Disse 10 sedecimale cifre kan fortolkes som *to* ordrer, da hver ordre fylder en halvcelle. Med én ordre kan man altså indlæse to. Den ene af disse (eller begge) kunne være 19-ordrer, hvorved man yderligere fik ordrer i overskud. Nu betår kunsten i at lave et program der efterhånden opbygger et så stort overskud af ordrer, at der kan dannes en løkke, der sluttelig kan indlæse hele operativsystemet. For at være helt nøjagtig må det nævnes at indholdet i KR-registeret skulle have en ganske bestemt værdi for at få koldstartprogrammet til at virke. Hvis KR fx havde indholdet 7FF sedecimalt, ville næste ordre blive hentet i celle 0 (adressering skete modulo 2048), altså netop den celle hvor de to indlæste ordrer var lagret. Efter installation af tromlen på Dask blev koldstart betydeligt simplere, idet operativsystemet kunne lagres permanent på tromlen. Dog skulle de relevante tromlekanaler hentes ned i lageret, så vippekontakten blev ikke helt overflødiggjort.
- **HOP.** Vipning af denne kontakt bevirkede at der kom operationskoden for et ubetinget hop (10 sedecimalt) i OP. Man kunne lige så godt prikke 001 0000 ind binært ved hjælp af knapperne i OP, men det var måske noget hurtigere at bruge vippekontakten. Ved at prikke en binær adresse ind i AS kunne man hoppe til en udvalgt del af sit program. Det kunne fx bruges til testformål; man prikkede binære værdier ind i registre og hoppede manuelt til et udvalgt sted i sit program - og så hvad der skete!
- **AR-->L.** Operationskoden for en gemmeordre fremkom i OP. Ved at prikke en binær værdi ind i AR og en lageradresse ind i AS kunne man med denne vippekontakt få lagret den binære værdi i den valgte celle, når gemmeordren blev udført på trin. Blev anvendt til testformål.
- **L-->AR.** Operationskoden for en henteordre fremkom i OP. Ved at prikke en binær adresse ind i AS og udføre ordren på trin, kunne man se indholdet i den valgte celle binært i AR. Blev anvendt

til testformål.

- 0-->REG. Denne vippekontakt nulstillede samtlige registre. Det kunne være en fordel i testsituationer, da man så kun behøvede at prikke alle etterne ind.

Trykknapperne til højre for højtaleren på billedet af kontrolbordet var til tænding og slukning af de to tromler, samt et par andre enheder jeg ikke kan huske hvad var.

Jeg håber at denne ret detaljerede beskrivelse af kontrolbordet har givet et indtryk af Dasks logiske struktur, samt kastet nogle strejflys over hvor primitivt det hele var.