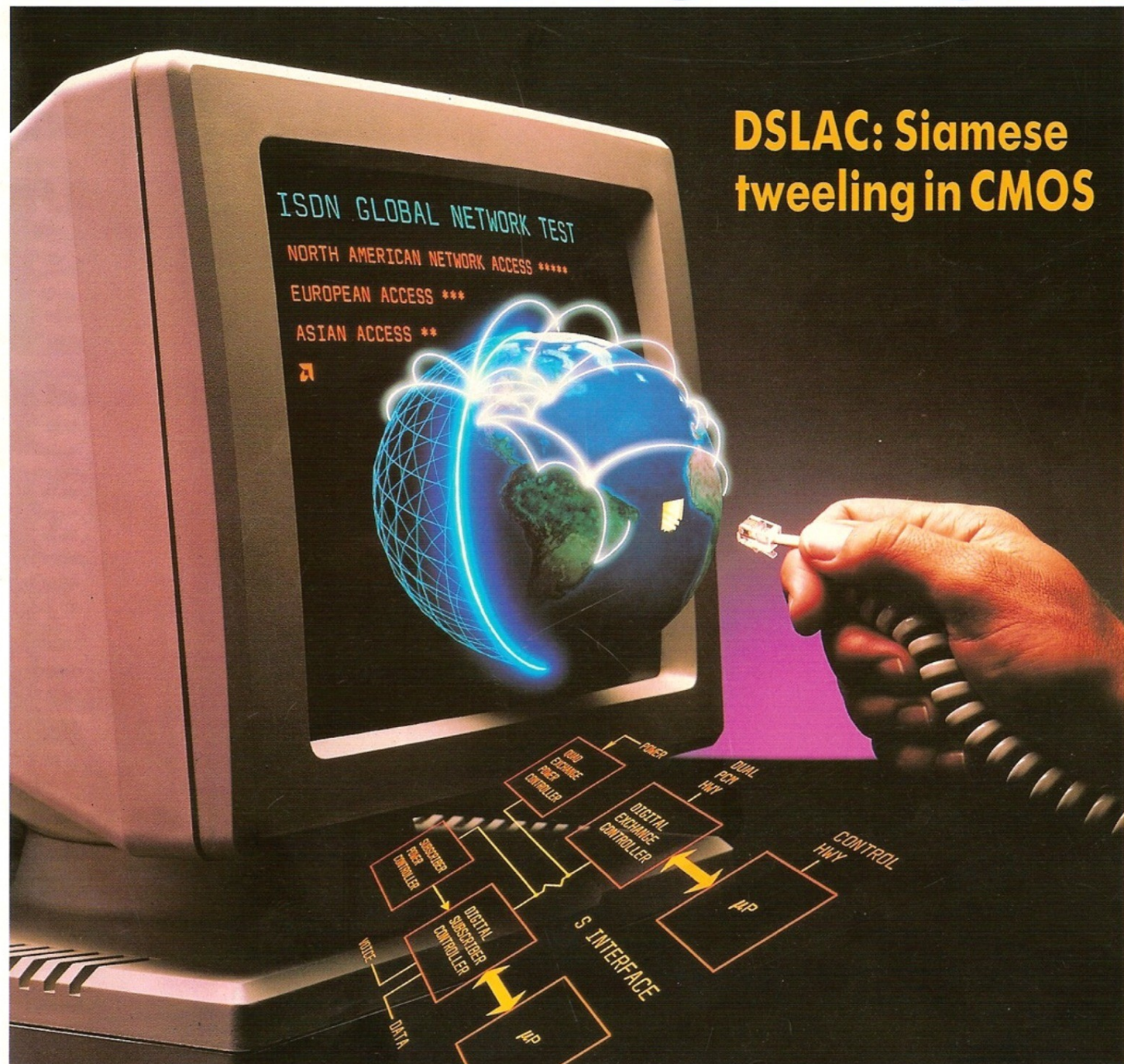


Dataflow wordt marktrijp

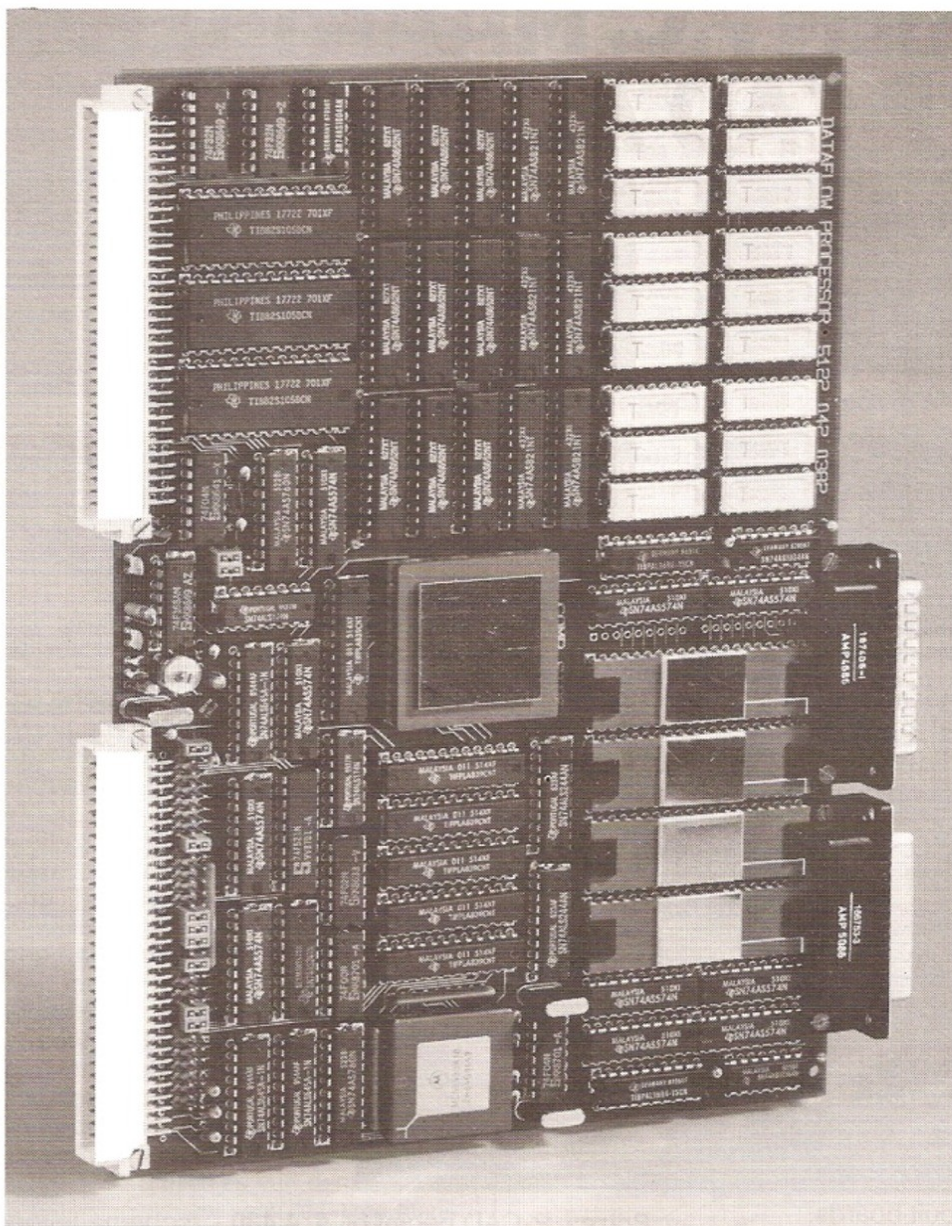
Verlichting in het platte vlak

Realisatie van sensoren

ELEKTRONICA



DSLAC: Siamese tweeling in CMOS



Dataflow wordt marktrijs

Grafische minisuper van eigen bodem werkt parallel

De belangstelling voor dataflow groeit zienderogen en dat is zeer begrijpelijk. Deze methode van programmasplitsing op instructieniveau biedt onderzoekers een hoogst boeiende uitdaging. De techniek maakt in principe een maximale paralleliteit mogelijk en beperkt het grootste knelpunt, de uitwisseling met het geheugen, tot een minimum. Tegenover deze riante theoretische voorsprong staan echter enkele grote praktische problemen. Een daarvan betreft de zogenaamde koppelenheid, die de voor een bewerking benodigde operanden bij elkaar moet brengen. Dataflow Technology Nederland stelt die moeilijkheid te hebben opgelost en heeft inmiddels op basis van hun methode een systeem ontworpen. Samen met enkele andere ontwikkelingen uit eigen koker resulteerde dat in de DFC, tot nu toe een van de weinige commerciële dataflowcomputers.

Elke dataflowkaart in de DFC bevat onder andere vier processoren, een geheugenbesturing en 2 Mbyte RAM.

Dataflow Technology Nederland belooft nogal wat. Zo is de DFC (Dataflow Computer) optimaal parallel: toevoeging van extra verwerkingskracht leidt tot een evenredige stijging van prestaties. Een dergelijke eigenschap lijkt in theorie vanzelfsprekend maar de praktijk wijst uit dat naarmate het aantal processoren toeneemt de prestatieverbetering steeds verder achterblijft. Debet daaraan is de voor de onderlinge gegevensuitwisseling benodigde communicatietaak, die bij uitbreiding van het aantal processoren steeds meer verwerkingskracht wegneemt ten koste van eigenlijke uitvoering van het programma.

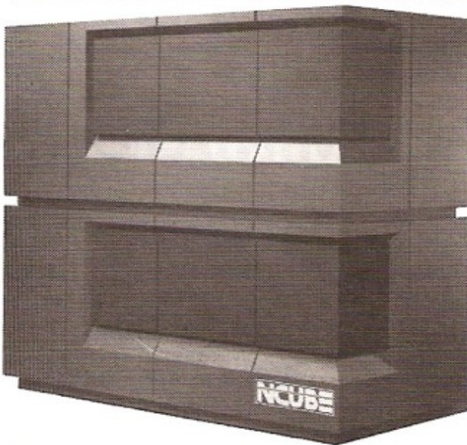
Bij de DFC speelt dit probleem niet en leidt de toevoeging van een tweede set van 32 verwerkingseenheden tot een verdubbeling van de verwerkingsnelheid, vooropgesteld uiteraard dat de software de extra capaciteit volledig benut. Voor zover bekend beschikken alleen de NCube en op transputers gebaseerde computersystemen over dezelfde eigenschap.

Werkt de transputer van Inmos met grofmazige paralleliteit, op het niveau van taken en processen, de DFC voert – zeer fijnmazig – simultaan enkelvoudige instructies uit. Het dataflowprincipe heeft als voordeel dat het niet langer nodig is om operanden uit het geheugen te lezen. In plaats van een chronologische afwerking van de objectcode met behulp van een instructieregister banen de extern ingevoerde of intern gegenereerde operanden zich zelfstandig een weg naar de bijbehorende instructie. Zodra data en opcode bij elkaar komen, kan de feitelijke uitvoering beginnen.

PROGRAMMATUUR

De meeste dataflowcomputers blijken moeilijk overweg te kunnen met de overvloedig gegenereerde data: herhaaldelijk treden bottlenecks op (ref. 1 en 3). DTN heeft nogal eigenzinnige hardwarevoorzieningen ontwikkeld om flessehalzen de kop in te drukken. Ook aan een ander probleem van dataflowcomputers, de enorme programmeerinspanningen, heeft men de nodige aandacht besteed. Volgens de vorig jaar van Philips afgesplitste firma is het mogelijk om bestaande applicaties met relatief weinig moeite geschikt te maken voor de DFC.

Momenteel werkt het Centrum voor Wetenschap en Informatica (CWI) aan een paralleliserende C-compiler voor het systeem. Het werk aan deze vertaler, die in staat is om zelf paralleliteit in de broncode te detecteren en te benutten, staat onder leiding van Arthur Veen. Hij is ook degene die soft- ▶



Hoewel gebaseerd op een ander principe, haalt dataflow dezelfde evenredige snelheidsverhoging bij toename van het aantal CPU's als dat bij de transputer en de NCube het geval is.

▷ ware ontwikkelde om standaard programmeertalen als C, Fortran en Pascal geschikt te maken voor dataflowcomputers. Tot nu toe maken deze computers doorgaans gebruik van functionele talen, die zich vanwege hun impliciete parallelisme in het bijzonder lenen voor op dataflow gebaseerde verwerking. Veen ontwierp bovendien een compiler voor de eerste dataflowcomputer, de Britse Manchester Machine.

GRAFISCHE WERKSTATIONS

In tegenstelling tot de Manchester Machine en andere tot nu toe gebouwde realisaties van het dataflowprincipe wordt de DFC een commercieel verkrijgbaar product. De fabrikant denkt aan introductie begin volgend jaar en mikt in eerste instantie op de top van de markt voor grafische werkstations. In de enkelvoudige uitvoering ontloopt het systeem de prestaties van dergelijke superwerkstations niet veel: real-time manipulatie van beelden met een resolutie van 1280 bij 1024, kleurwaardeberekening van 44 miljoen pixels per seconde met een keuze uit een 24-bit palet alsmede grafische bewerkingen als lichtstraaltracering, gammacorrectie, patroonprojectie, Phong- en Gouraud-schakering (dat laatste met een snelheid van 200 000 driehoeken per seconde).

Koppeling van meerdere DFC's maakt het mogelijk om desnoods nog complexere afbeeldingen real-time aan het oog te presenteren. Daarbij hangt de complexiteit van een tekening zowel af van de hoeveelheid objecten per afbeelding als de hoeveelheid polygonen per object. Deze koppeling levert, dankzij de inherente fijnmazigheid van dataflow, een recht evenredige capaciteitsverhoging op.

GEGEVENSSTROOM

Een minstens net zo belangrijk voordeel

van dataflow is dat de tijd die gepaard gaat met dataverzending niet hoeft te leiden tot een snelheidsvermindering van het systeem. Elke instructie staat immers op zichzelf en kan onafhankelijk van de rest van het programma worden uitgevoerd. Zijn bepaalde gegevens niet voorhanden dan schakelt de processor over op een wel aanwezige combinatie van opcode en operanden.

Zolang er voldoende instructies in de wachtrij voorradig zijn, kan een vrijwel onbeperkt groot netwerk van verwerkingseenheden worden gerealiseerd. Ondanks dat dit onherroepelijk leidt tot grotere afstanden tussen de processoren, zowel in ruimte als in tijd, zorgt de continue aanvoer van opdrachten er voor dat er toch sprake is van een evenredige capaciteitsvergroting.

Een praktische manier om meerdere DFC's te koppelen is door middel van de hyperkubus (fig. 1). Deze projectie van een N-dimensionale kubus heeft als eigenschap dat er 2^N knooppunten in kunnen worden opgenomen. Verder is elke verwerkingseenheid is verbonden met N aangrenzende stations via evenzoveel communicatiekanalen. Het grote voordeel van deze topologie is dat de afstand tussen twee knooppunten over nooit meer dan $N - 1$ tussenstations loopt, wat de routing vereenvoudigt en de looptijd bekort.

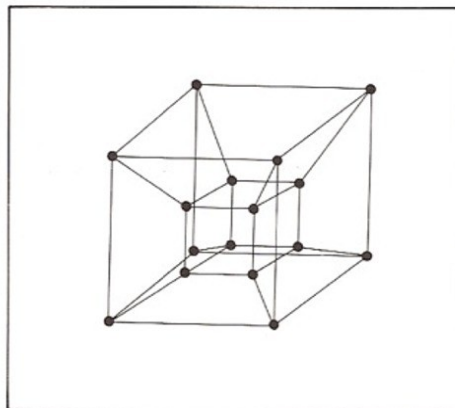


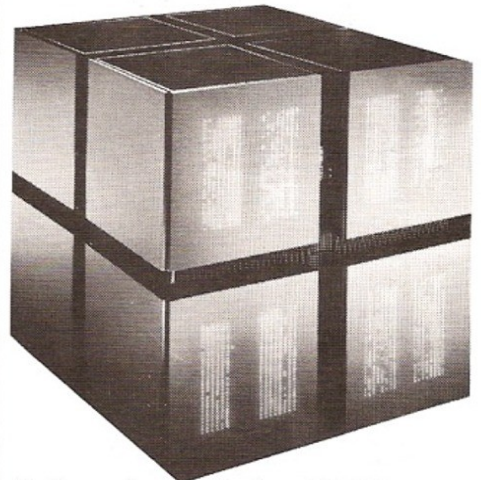
Fig. 1. De DFC16 is voor te stellen als een hyperkubus waarin 16 DFC's aan elkaar zijn gekoppeld.

LICHTSNELHEID

De prestaties van de DFC zijn allereerst het gevolg van de parallelle opzet. Dat in toenemende mate wordt gewerkt met verschillende vormen van simultane verwerking vindt zijn oorsprong in de fysieke beperkingen van de huidige IC's. Hoewel de theoretische bovengrens nog niet is bereikt, houdt de snelheidswinst ooit op omdat signalen zich niet sneller kunnen voortplanten dan het licht. Uitgaande van de huidige CMOS-technologie is de verwerkingsnelheid hooguit met een factor tien op te

voeren. Ongetwijfeld doen zich in de nabije en verdere toekomst nieuwe doorbraken voor, maar de momenteel gewenste verwerkingskracht gaat de geboden capaciteit van een enkele chip nog steeds ver te boven.

Het koppelen van meerdere CPU's die gelijktijdig hetzelfde probleem werken biedt wel een – in theorie – onbeperkte door-groei en dit verklaart de opkomst van de parallelle computer. De opmars voltrekt zich vooral binnen de gelederen van de hardware, zij het dat de op papier uitgedachte architecturen de uiteindelijk gerealiseerde ontwerpen ruimschoots overtreffen. Zo is een van de bekendste parallelle computers, de Connection Machine van Thinking Machines Corporation (TMC), gebaseerd op een hyperkubus met 65 536



De Connection Machine bevat 65 536 eenvoudige processoren. Knelpunt hier is de gelijkmatige benutting van alle eenheden.

zeer eenvoudige 1-bit processoren. In de praktijk blijkt dit systeem echter maar beperkt bruikbaar terwijl alleen in hele specifieke situaties de verwachte snelheidsverhoging wordt benaderd.

ONVOLKOMEN DELING

Toch vormt niet de hardware maar de software het voornaamste knelpunt bij de uitwisseling van informatie tussen verwerkingseenheden. Het aanvankelijke enthousiasme over parallelisme kreeg een knauw toen het buitengewoon lastig bleek om een programma gelijkmatig in stukjes te knippen. De potentiële snelheidswinst werd voor een groot deel teniet gedaan door een ongelijkmatige aanwending van de processoren. Rekencapaciteit was volop aanwezig maar ze werd niet benut.

Deze onvolkomen verdeling van taken is onvermijdelijk omdat elk programma sequentieel van aard blijft. Het gaat steeds om een reeks achtereenvolgens uit te voeren opdrachten die vaak afhankelijk zijn van voorgaande bewerkingen; pas als be- ▷

▷ paalde gegevens als tussenresultaten beschikbaar zijn, kan de uitvoering van een programma verder. Deze data-afhankelijkheid stelt beperkingen aan de mate van paralleliteit van een programma en die varieert met de applicatie.

Inmiddels is wel de nodige vooruitgang geboekt bij dit als *load balancing* aangeduide probleem. Programmeurs nemen eerst goede nota van de hardware-architectuur – met name het aantal verwerkingseenheden – voordat ze beginnen aan een min of meer gelijkmatige verdeling van de processen. De bouwers van compilers, zo belangrijk voor het waardebewoud van bestaande software, streven naar een hiërarchie van processen zodat een nadere verdeling lang mogelijk blijft. Lukt het een compiler niet om een programma efficiënt in subroutines te splitsen, dan wellicht wel op lussniveau binnen een subroutine.

Het laagste niveau is dat van de instructies. Daar is de paralleliteit weliswaar maximaal maar ontstaat tegelijk een nieuw probleem. Om alle processoren van data te voorzien wordt de communicatielast enorm. Dit dan nog afgezien van het feit dat elke opdracht vergezeld moet gaan van leesacties uit het geheugen die de interne bussen en kanalen nog eens extra belasten.

DATAFLOW

Bij dataflow zijn leescyclussen niet nodig. De aanwezigheid van data bepaalt de voortgang van een programma en niet de achtereenvolgende afhandeling van instructies op basis van een program counter. Met andere woorden: zodra de betreffende operanden zijn aangekomen bij het instructiegeheugen kan de uitvoering van de bijbehorende opdracht beginnen. Een dergelijke handelwijze beperkt de communicatie met het geheugen aanzienlijk en maakt een program counter overbodig. In het algemeen werkt dataflow als volgt: Een compiler deelt een gegeven applicatie op in instructies en kent elk daarvan een label toe. De gegevens krijgen eveneens een label, dat moet corresponderen met de eerstvolgende instructie. Deze voorziet niet alleen in een opcode die de uit te voeren bewerking bepaalt maar ook in een bestemmingslabel voor het resultaat. Op die manier vinden de gegenereerde gegevens steeds opnieuw hun bestemming.

Zodoende ontstaat als het ware een ring, gevuld met *tokens* – operanden vergezeld van een bestemmingslabel – die zich steeds opnieuw naar de verwerkingseenheden spoeden, daarbij en passant hun instructie ophalend (zie figuur 2). Een token krijgt een plaatsje in het geheugen wanneer de bewerking twee operanden nodig heeft en de tweede nog niet aanwezig is.

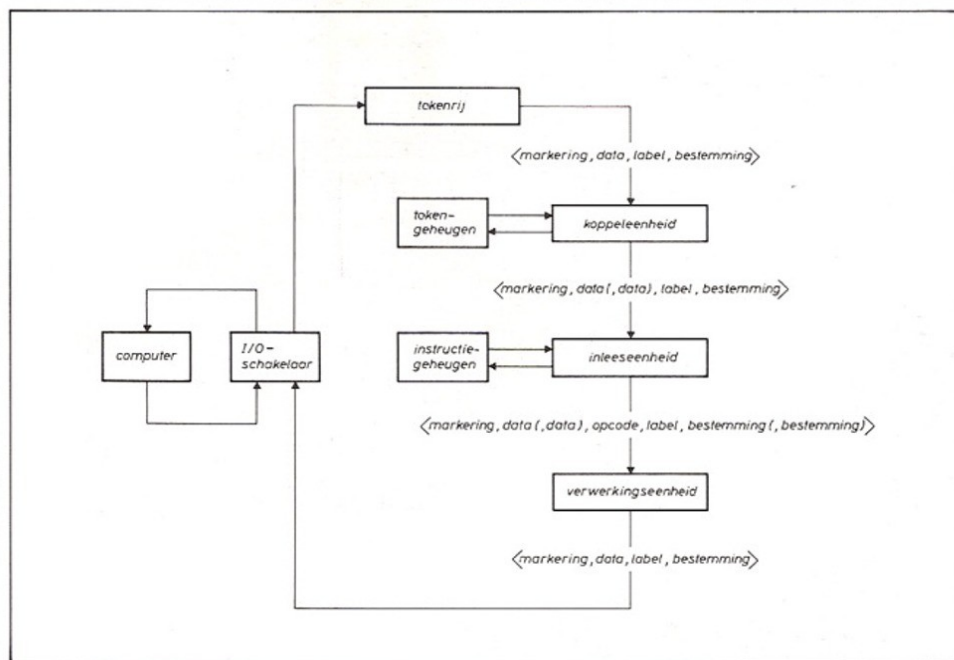


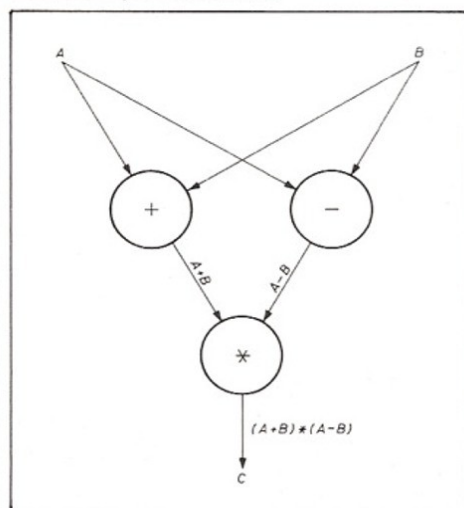
Fig. 2. Elke dataflowcomputer beschikt over een ring waarin gegevens circuleren op weg naar hun instructie. Veel bewerkingen vereisen koppeling met een andere operand.

GRAAF

De wijze van verwerken en van programmeren kan heel goed worden gerepresenteerd met behulp van een gerichte graaf. Daaruit blijkt tevens heel duidelijk de geschiktheid ervan voor parallelisering. Neem bijvoorbeeld de volgende instructiereeks:

```
add A, B
store T1
sub A, B
store T2
mult T1, T2
store C
```

Fig. 3. Uit representatie met behulp van een graaf komt de mate van paralleliteit goed naar voren. Bovenstaande bewerking zou in geval van sequentiële verwerking zes stappen vereisen in plaats van twee.



De eerste twee hoofdbewerkingen zijn volkomen onafhankelijk van elkaar en kunnen dus simultaan worden afgehandeld. Dit gebeurt in de graaf van figuur 3 en daaruit komt naar voren dat de gelijktijdige uitvoering van de optelling en de aftrekking de bovenstaande reeks van zes cyclussen terugbrengt tot een reeks van twee.

In de graaf geven de pijlen de datapaden aan en de cirkels de knopen waar zich de bewerkingen voltrekken. Hij vormt zo in feite de visualisatie van het verdelingsproces dat een paralleliserende compiler voor zijn rekening neemt.

STATISCH

Ten aanzien van de realisatie wordt er onderscheid gemaakt tussen statische en dynamische dataflowcomputers. De DFC valt in de eerste categorie, een direct gevolg van de gekozen CPU-chip, en dit betekent dat de uitvoering van een programma precies zo verloopt als de graaf aangeeft (fig. 4). Wijzigingen in het aantal knopen tijdens de verwerking doen zich niet voor. Dat heeft als nadeel dat een extra mogelijkheid tot gelijktijdigheid onbenut blijft. Komt er ergens een vorm van data-afhankelijkheid voor dan ligt het voor de hand om de op gegevens wachtende processoren in te schakelen bij het genereren van diezelfde data. Ter wille van de snelheid zouden de CPU's dus een andere taak moeten kunnen krijgen.

Het statische karakter van de DFC gaat echter in de praktijk niet ten koste van de prestaties, die zelfs aanmerkelijk beter zijn dan die van de bekendste dynamische versie, de Manchester Machine. Volgens ▷

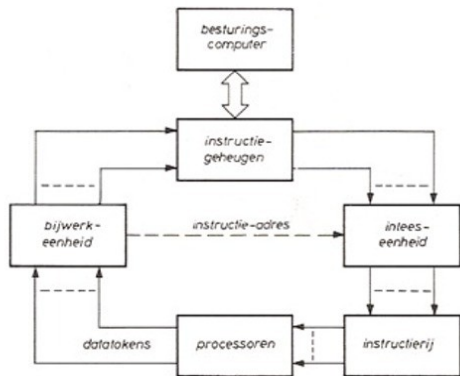


Fig. 4. Bij statische dataflow blijft het aantal knopen gelijk en komen de gegevenspaden overeen met de graaf, zodat alle gegevens hun bestemming kennen en er geen speciale koppelenheid nodig is.

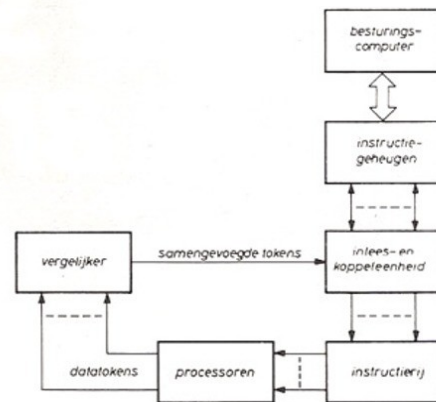


Fig. 5. Door niet-benutte knopen in te zetten voor andere taken neemt het rendement in theorie toe. De benodigde koppelenheid levert in de praktijk netto een vertraging op.

▷ technisch directeur Hans de Vries komt dat omdat een bestaand programma al bij voorbaat gemakkelijk kan worden geoptimaliseerd. Daardoor hoeft de parallelisering niet tijdens het draaien van de applicatie te gebeuren.

TIJDWINST

De door DTN aangedragen oplossing komt erop neer dat het aanbrengen van parallelle constructies in de broncode zo weinig moeite kost dat ook aanpassingen achteraf geen problemen hoeven op te leveren. Er is niets op tegen om het programma na compilatie eerst uitvoerig te laten proefdraaien, waaruit blijkt of de bestaande code – indien gewenst – nog verder kan worden geoptimaliseerd. Wanneer een paralleliserende compiler beschikbaar is, kan de programmeur zich concentreren op de laatste potentiële paralleliteit die nog in de broncode besloten ligt.

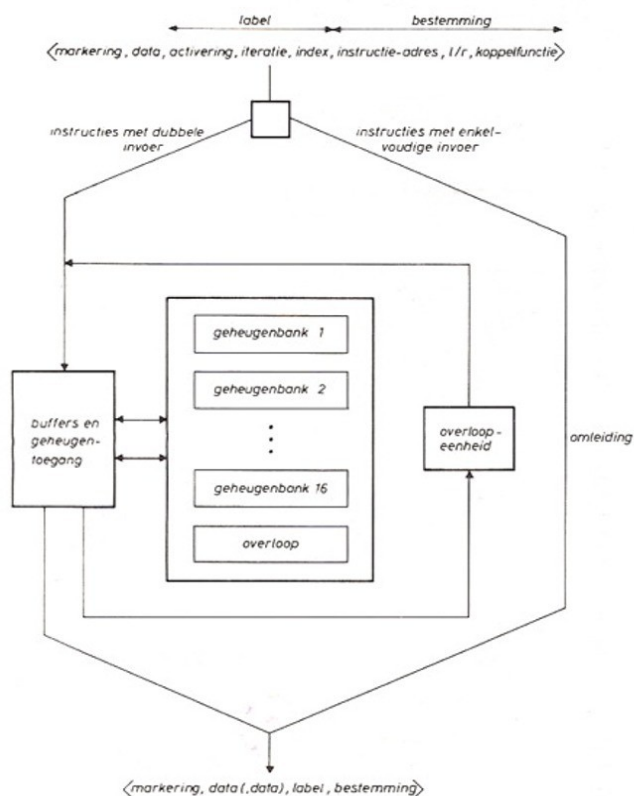
Het gevolg van deze benadering is wel dat er geen vergelijker nodig is, de zogenaamde tagging-eenheid, die tijdens de programma-uitvoering onderzoekt welke bestemming de gegevens kunnen krijgen en die voor de bijbehorende labels zorgt (fig. 5). Een dergelijke voorziening vergt nogal wat extra tijd en vermogen. Daar komt bij dat de statische opzet het mogelijk maakt direct audio- en videodata in de graaf te injecteren. De bijbehorende opcode is immers altijd aanwezig. Bij de dynamische tegenhanger is dat niet het geval, zo komen de gegevens bij de Manchester Machine eerst in een buffer terecht.

Fig. 6. De Manchester Machine versnelt het zoekproces door meerdere geheugenbanken tegelijkertijd aan te spreken.

AUTOMATISCHE KOPPELING

De belangrijkste reden dat statische dataflow in de praktijk efficiënter werkt dan dynamische is het ontbreken van een afzonderlijke koppelenheid (matching unit). Deze zorgt voor het combineren van operanden onderling, mochten ze gezamenlijk nodig zijn. Hoewel veel instructies genoeg hebben aan één operand, komt de situatie

met twee ingangsgegevens eveneens vaak voor. De aanwezigheid van een tokengeheugen is daarbij een eerste vereiste. Het zou louter toeval zijn wanneer twee tokens met hetzelfde label tegelijk de koppelenheid binnenkomen om vervolgens als één geheel hun weg te vervolgen naar het instructiegeheugen. Probleem bij dataflowcomputers is dat het ▷



▷ tokengeheugen zich dynamisch vult, afhankelijk van het verloop van het programma. Van tevoren staat dus niet vast waar zich een token in het geheugen bevindt zodat steeds een zoekactie door het hele geheugen noodzakelijk is. Vanzelfsprekend zijn er algoritmen om deze speurtocht te bespoedigen. Zo werkt de Manchester Machine met segmentering: hij destilleert uit een label een adres dat wordt gebruikt voor een parallelle zoekactie in zestien geheugenbanken. Dit levert 16 in aanmerking komende tokens op waardoor een verdere zoektocht beperkt kan blijven (fig. 6). Helaas is het vooralsnog niet goed mogelijk om de zoekactie geheel van zijn sequentiële karakter te ontdoen. Associatieve geheugens zijn momenteel nog veel te kostbaar en fysiek te omvangrijk. Kortom: in de koppelenheid treedt tijdverlies op. Dit in tegenstelling tot de inleeseenheid (fetching unit) die zorgt voor de samensmelting van het token met een opcode uit het instructiegeheugen. Het adres daarvan ligt al vanaf het begin vast zodat een zoekactie achterwege kan blijven.

WACHTRIJ

Doordat de koppelenheid het langzaamste onderdeel van het systeem is, heeft hij de neiging het geheel te vertragen. Juist omdat de mate van paralleliteit in dataflowmachines zo groot is, doet dit probleem van snelheidsafkalking zich maar al te vaak voor. Uit figuur 2 en 5 blijkt duidelijk dat de inleeseenheid en de CPU's niets anders kunnen doen dan hun capaciteit afstemmen op die van de koppelenheid. Nog afgezien hiervan leidt de gebrekkige snelheid van de koppelenheid tot een derde geheugen in het systeem, aangeduid als overloopenheid of tokenwachtrij. In deze buffer maken de tokens hun opwachting alvorens voor koppeling in aanmerking te komen.

Figuur 2 staat in zoverre model voor alle dataflowcomputers dat er nog geen machines zijn waarin een centrale koppelenheid ontbreekt. De DFC brengt daar verandering in. Uiteraard maakt de architectuur zelf koppeling onontkoombaar maar dat gebeurt decentraal bij elk van de 32 processoren.

MODULENUMMER

DTN heeft gebruik gemaakt van de μ PD7281 van NEC, een geïntegreerde dataflowprocessor die zelf zorgt voor het combineren van tokens respectievelijk de vereniging daarvan met de bijbehorende instructies. Daartoe beschikt de CPU over een speciale sectie voor gegevensbeheer, bestaande uit een datageheugen – gericht op tokenparing – en een functietabel die wordt gebruikt bij aanhechting van de opcode. Het tokengeheugen in de NEC-pro-

cessor beslaat ruim 1 Kbyte (512×18 bit). Bij de Manchester Machine is dit daarentegen 256 K, wat uiteraard ook een langer zoekproces inhoudt.

Deze winst is alleen maar mogelijk wanneer een token bekend is met een van de 32 bestemmingen. Een centrale koppel- en inleeseenheid maken het nog mogelijk om betrekkelijk onverschillig te zijn over de vraag welke CPU de uitvoering van een bepaalde operatie voor zijn rekening moet nemen. Koppeling en verwerking staan dan los van elkaar, waardoor de processoren zich altijd verzekerd weten van een compleet, hapklaar pakketje (gegevens inclusief opcode). In de DFC krijgen zij echter alleen maar tokens en kunnen ze niets

sen twee ingangen en twee uitgangen (figuur 7).

Het is niet nodig om voor ieder token afzonderlijk schakelfuncties te verrichten. In plaats hiervan sorteert een achter elk vier-tal dataflowprocessoren geplaatste splitter op modulenummer. Aldus ontstaan 32 gegevensstromen, waarvan er 28 parallel de CPU-kaart verlaten en de resterende vier lokaal naar de eenheden op de eigen print gaan. Deze procedure doet zich voor op elk van de acht processorkaarten. Het netwerk-circuit zorgt er voor dat al deze tokenstromen gesorteerd op modulenummer bij elkaar komen. Vervolgens gaan ze weer via eenzelfde databus en opnieuw geheel parallel, naar de module waarmee

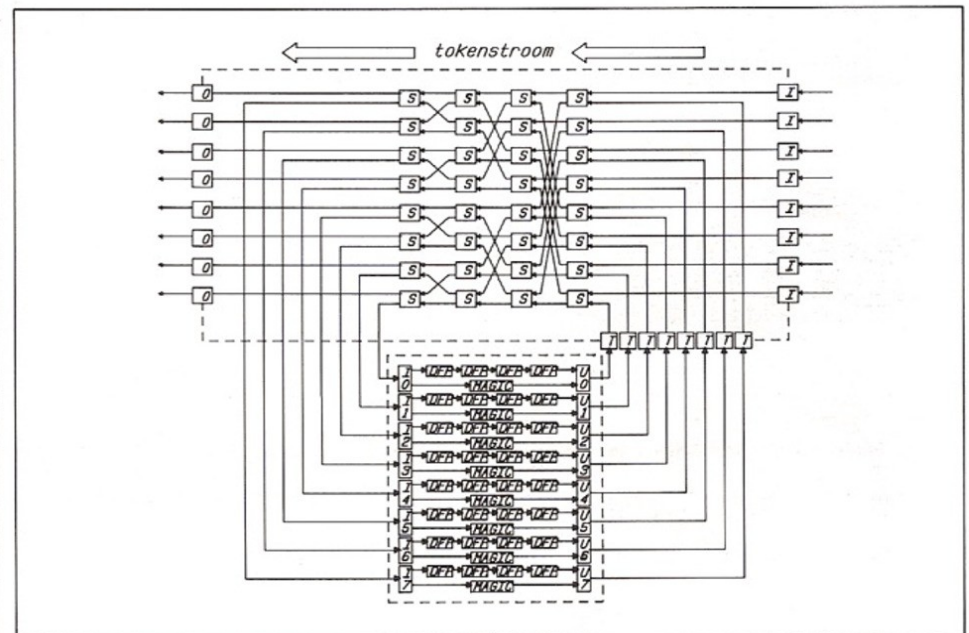


Fig. 7. Globale opbouw van de DFC: een schakelnetwerk zorgt voor de routing van de in- en uitgaande tokenstromen bij de acht processorkaarten.

uitrichten als ze zelf niet de bijbehorende instructie beschikbaar hebben.

Dit betekent dat het van groot belang is dat de tokens bij de juiste processor terechtkomen. Dat gebeurt aan de hand van het modulenummer van elke dataflowchip. Elk token gaat behalve van een label – door DTN, in navolging van NEC's terminologie, *identifier* genoemd – ook vergezeld van dit modulenummer. Externe gegevens krijgen via de in- en uitvoerkaarten direct bij binnenkomst een label en een modulenummer mee. Vervolgens gaan ze als tokens de ring in, op weg naar de betreffende module.

SPLITSSEN EN SAMENVOEGEN

Een speciaal netwerkcircuit zorgt ervoor dat de tokens op de juiste plaats belanden. Op deze kaart bevinden zich acht gate-array's, elk geprogrammeerd met vier twee-naar-twee-schakelaars die schakelen tus-

de nummers corresponderen. Het is dan de taak van de afzonderlijke verwerkings-eenheden om het modulenummer te controleren en het token eventueel door te verwijzen naar een volgende collega.

MENGEN

Aan de uitvoerkant van een processorkaart is het niet mogelijk om de stromen dezelfde, gelijkmatige tokendichtheid mee te geven. Tenslotte is de intensiteit van het gegevensverkeer aan de uitgang afhankelijk van de verloop van het programma. Dat heeft wel weer tot gevolg dat aan de ingangen van de andere kaarten sprake is van een ongelijkmatig aanbod van tokens per stroom.

Hier deed bij de ontwikkeling van het systeem heel lang een knelpunt voor omdat een synchrone wijze van splitsing metastabiliteit – vertraging door het inlezen van de data op de flank van de systeemklok – ver-

▷ oorzaakte. Een asynchrone aanpak verliep aanvankelijk veel te langzaam. Uiteindelijk heeft technisch directeur De Vries naast een splitser ook een mengenheid (merger) ontworpen die op elke kaart de tokenstromen van andere kaarten met de stroom vanuit de eigen ring samenvoegt om vervolgens zonder enige vertraging alles in te kloppen. Elk viertal processoren accepteert aldus evenveel tokens als iedere kaart per seconde genereert (maximaal 5 miljoen).

OPBOUW

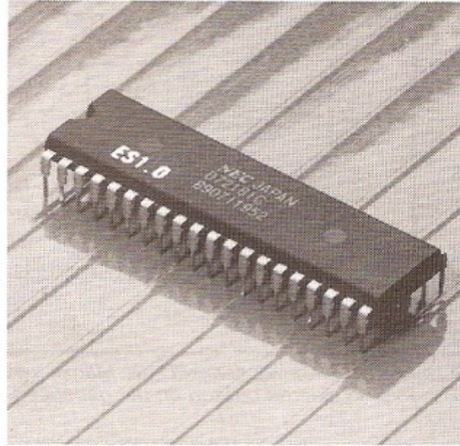
Uit figuur 7 blijkt behalve de structuur van de netwerkmodule ook de opbouw van de gehele tokenring, althans in een configuratie voor een enkelvoudige DFC. De acht dataflowkaarten bevinden zich in de onderste helft van de figuur en krijgen hun tokens van en sturen hun data naar het netwerkcircuit bovenin. Deze schakeling bestaat voor het grootste deel uit een Omega-netwerk. Elke tokenstroom krijgt bij binnenkomst via een vertaaltabel *T* een 4-bit extensie mee die de route binnen het netwerk bepaalt.

De routing hangt af van zowel bron als bestemming en werkt in wezen heel eenvoudig. Elk van de vier bits staat voor een kolom schakelaars. Een 1 zet de schakelaars naar het bovenste van de twee uitgaande kanalen, een 0 naar de onderste uitgang. Als bijvoorbeeld een verbinding van kaart 0 naar kaart 7 tot stand moet worden gebracht, gebeurt dat met de bitcode (1,1,1,0). De combinatie (1,1,1,1) zou hebben geleid tot het routeren van de gegevensstroom naar uitvoerkanaal 7. Deze mogelijkheid is vooral van belang als meerdere DFC's onderling moeten worden gekoppeld.

IC

Het is niet verwonderlijk dat de ontwerpers voor de realisatie van de DFC hebben gekozen voor de dataflowprocessor van NEC, de μ PD7281. Op zich is het mogelijk om een systeem samen te stellen uit andere standaardcomponenten maar dat vergt extra hardware, zoals de koppel-eenheid, terwijl bovendien een deel van de potentiële snelheidswinst blijft liggen.

Dat de μ PD7281 speciaal is ontwikkeld met het oog op dataflowverwerking blijkt vooral uit het feit dat de koppeling van operanden en opcode binnen het circuit zelf plaatsheeft. Een dergelijke benadering past geheel binnen de filosofie van de DFC, waar koppeling op decentraal niveau immers essentieel is. Daarnaast heeft de μ PD7281 nog een ander belangrijk voordeel: implementatie van dataflowtechnieken op één chip maakt het betrekkelijk eenvoudig om capaciteitsverhogende technieken, zoals pijplijnen, mee te integreren.



Basis van de verwerkingseenheden is de μ PD7281 dataflowprocessor van NEC.

Een en ander heeft geresulteerd in een CPU met een verwerkingsnelheid van 10 Mips bij een maximale klokfrequentie van 20 MHz.

PIJPLIJN

De dataflowprocessor heeft in feite als enige taak om 10 miljoen keer per seconde een knoop uit een graaf te imiteren. Maar voordat het zover is, heeft de chip al een flink aantal bewerkingsschakelingen op het token uitgevoerd (zie figuur 8). Zo zendt de invoerbesturing elk modulenummer door dat niet overeenkomt met het eigen 4-bit nummer. Het token gaat in dat geval rechtstreeks naar de uitvoerbesturing en blijft buiten de pijplijn van de processor.

Is het modulenummer op een gegeven moment niet langer van belang, namelijk zodra het gegevenspakketje in de processor is opgenomen, het label is dat dan des te meer. Dit is een adres van 7 bit dat via de koppeltabel de 6-bit locatie oplevert van de

plaats waar de instructie zich in de functietabel bevindt. Binnen de koppeltabel krijgt elk token ook een nieuw adreslabel voor bewerking binnen een volgende knoop. De functietabel voorziet niet alleen in de opcode maar zorgt ook voor het bijbehorende adres van de operand in het datageheugen en staat hiertoe in contact met een adresgenerator. Binnen het datageheugen treffen de operanden elkaar voor zover er sprake is van instructies die meer dan één parameter nodig hebben.

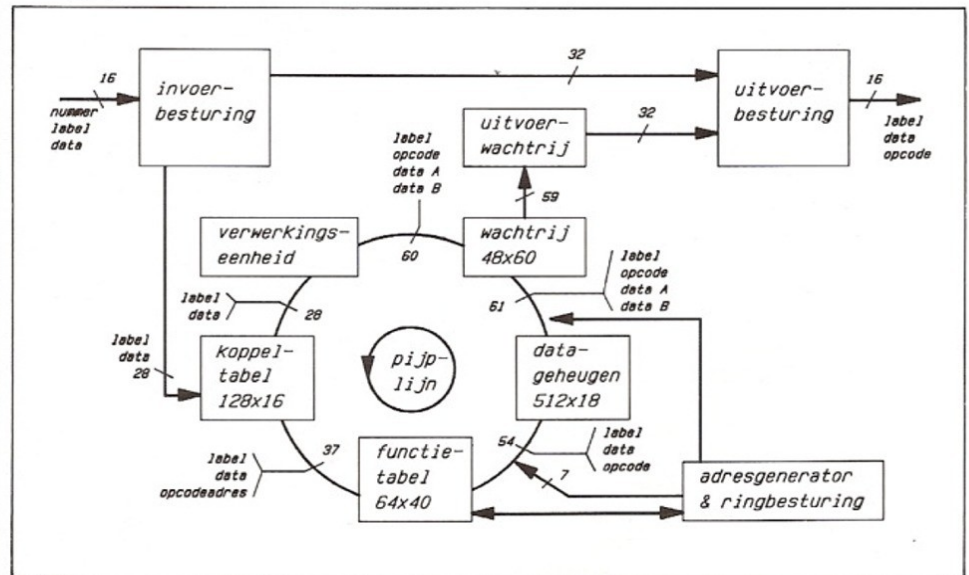
Uit de bovenstaande beschrijving van de manier waarop de processor functioneert blijkt tevens dat nergens zoekacties nodig zijn als gevolg van een zich dynamisch vullend tokengeheugen. Het label is een adres voor de koppeltabel, die geeft op zijn beurt de locatie voor de functietabel die het dynamische datageheugen beheert met behulp van een adresgenerator. Zodoende komt elk token linea recta op de goede plaats aan. Behalve de parallele opzet bij de koppeling van operanden is dus ook het IC zelf verantwoordelijk voor de efficiënt verloopende tokenparing.

CIRCULATIE

Na het datageheugen zijn de gegevens gecombineerd met de instructie en komt het pakket in een wachtrij van 48×60 bit. Deze buffer vangt pieken in het tokenaanbod op en zorgt er in rustiger momenten voor dat de processor in staat is om constant aan het werk te blijven. Uiteindelijk is het de beurt aan de verwerkingseenheid zelf. De CPU doet niets anders dan continu de binnenkomende opcode uitvoeren op de meegeleverde operanden.

Het resultaat van de bewerking verlaat de interne ring niet direct. In vier van de vijf gevallen bevindt de volgende bewerking zich ▷

Fig. 8. Interne structuur van de chip en de token-opbouw bij de verschillende stadia van de interne pijplijn.



▷ binnen dezelfde chip en maakt een token opnieuw een ronde door de pijplijn. Maar ook wanneer de volgende bewerking zich op een andere processor bevindt, is hercirculatie nodig om een nieuw modulenummer op te halen (zie figuur 8).

Deze hernieuwde circulatie is enigszins omslachtig en heeft te maken met de ontoereikende woordbreedte van de bus tussen de verschillende interne eenheden van de μ PD7281. Hierdoor is het niet mogelijk om een token direct al een nieuw adreslabel en een nieuw modulenummer mee te geven. Derhalve moet eerst de bitlengte van een token worden teruggebracht en dit gebeurt binnen de CPU. Daarna zijn instructie en operanden niet meer nodig zodat alleen een label en een resultaat overblijven. De omvang daarvan is klein genoeg om wel een modulenummer te kunnen toevoegen.

Na aanhechting van een modulenummer in de functietabel verlaat het 32-bit token de chip in twee gedeeltes: 16 bit data en 16 bit routeringsinformatie, waaronder het modulenummer en het label. Het inlezen in een volgende processor gebeurt eveneens in twee klokcyclussen van elk 100 ns.

MAGIC

Overigens komt de objectcode niet direct in de functietabellen van de processoren terecht. Via de *Magic*, de Memory Access and General bus Interface Chip (μ PD9305), belandt de code eerst in een bij elk viertal dataflowkaarten geplaatst geheugen van 2 Mbyte. Dit IC verzorgt vervolgens voor alle bijbehorende verwerkingseenheden de aansturing van en de toegang tot dit lokale geheugen door lezen en schrijfoverdrachten uit te voeren (fig. 9). Het in de processor laden van door de *Magic* in tokenvorm omgezette instructies geschiedt op de achtergrond door middel van de DMA-functies van de chip en duurt gemiddeld een paar microseconden.

NEC is van plan om in eind 1991 of begin 1992 met een 32-bit versie van de μ PD7281 te komen. In die uitvoering wordt de externe communicatie aanzienlijk versneld omdat tokens nu in één klokcyclus kunnen worden weggeschreven en ingelezen. Verder moet de opvolger gaan beschikken over een interne numerieke co-processor terwijl het bovendien de bedoeling is dat de interne bus wordt verbreed zodat hernieuwde circulatie niet meer nodig is. Toepassing van een dergelijke chip zou de DFC geschikt maken voor de supercomputermarkt en DTN bevestigt te zijner tijd deze markt te willen betreden.

GRAFISCHE FUNCTIES

Op zich vormen de dataflowkaarten samen een systeem dat in principe geschikt is voor allerlei doeleinden. Maar de DFC is in

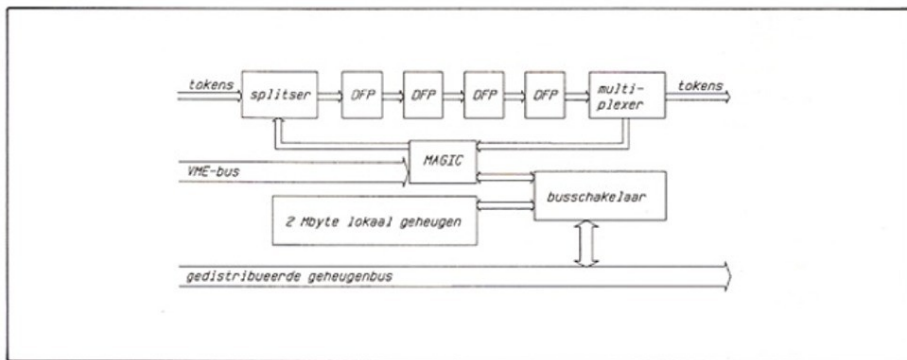


Fig. 9. Een afzonderlijk IC, de *Magic*, verzorgt de communicatie – zowel direct als via DMA – met de externe bussen en het geheugen.

eerste instantie bedoeld als grafische co-processor voor een werkstation teneinde real-time bewegende beelden te kunnen creëren. Om echter circa een miljoen beeldpunten in een fractie van een seconde te bewerken, is een capaciteit van 320 Mips – acht kaarten à vier processoren die elk 10 Mips leveren – niet genoeg. Ter aanvulling is dan ook voorzien in vier grafische kaarten met in totaal veertig gespecialiseerde CPU's die samen goed zijn voor 448 Mips.

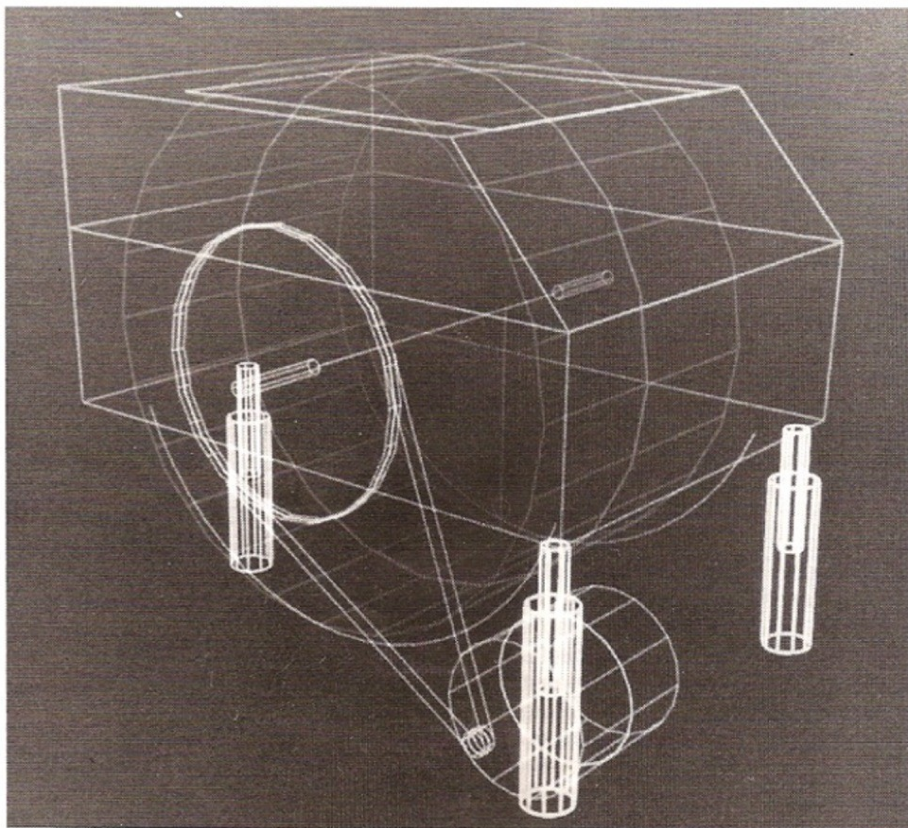
Ook hier is het gevaar van flessenhalzen levensgroot: een continue stroom grafische informatie moet over een centrale bus naar de juiste kaart respectievelijk de gewenste processor en vervolgens, na bewerking,

weer terug. Bovendien maakt de aard van de gegevens het onmogelijk om een andere dan sequentiële afhandeling te hanteren. Teneinde knelpunten te voorkomen hebben de ontwerpers voor twee verschillende, elkaar aanvullende invalshoeken gekozen. In de eerste plaats is de gegevensuitwisseling tussen dataflowkaarten en grafische eenheden via hoog-niveau informatie zoveel mogelijk gecomprimeerd. Hardwarematige aanvulling hierop vormen een zelf ontwikkelde, snelle databus en parallelle toegang tot het videogeheugen.

DATATRAJECT

Zoals gebruikelijk bij het werken met grafische informatie worden afbeeldingen zo-

Fig. 10. Het mathematische draadmodel is weliswaar driedimensionaal maar behoeft nog enkele bewerkingen voordat het als massief object kan worden gezien.



veel mogelijk gerepresenteerd door middel van objecten. Deze objecten bestaan in principe uit het soort element, de bijbehorende coördinaten en eventuele attributen. Voor de representatie maakt de DFC gebruik van standaarden als PHIGS+ en GKS.

Alle objecten bestaan uit een verzameling polygonen. Met behulp van transformatiematrices, de meest efficiënte rekenkundige beschrijving om een veelhoek vast te leggen en van vorm te kunnen veranderen, brengen de dataflowprocessors – al dan niet interactief – wijzigingen in de elementen aan. De transformatiematrix voorziet daarbij in de belangrijkste operaties: verplaatsing, rotatie en vergroting of verkleining in X- en/of Y-richting.

Wat resteert is nog niet een verzameling punten maar een algoritmische beschrijving met behulp van een coördinatenstelsel. Dit levert weliswaar een driedimensionale afbeelding op maar alleen in de vorm van een draadmodel zonder de grafische eigenschappen van een massief object. Zo is van inkleuring, schaduwwerking en eventuele kleurschakering nog geen sprake en vlakken in de achtergrond, die normaal gesproken door andere delen van het voorwerp aan het oog zouden worden onttrokken, zijn nog gewoon zichtbaar (fig. 10).

Het model uit figuur 10 na invulling van de elementen en het verwijderen van verscholen vlakken. (foto: AEG)

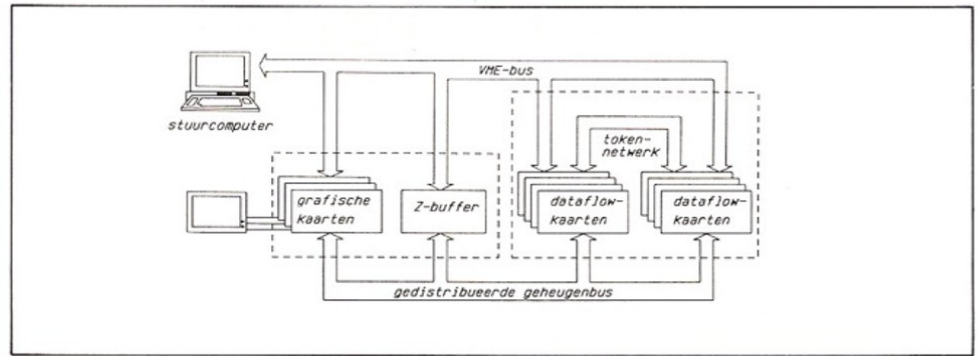
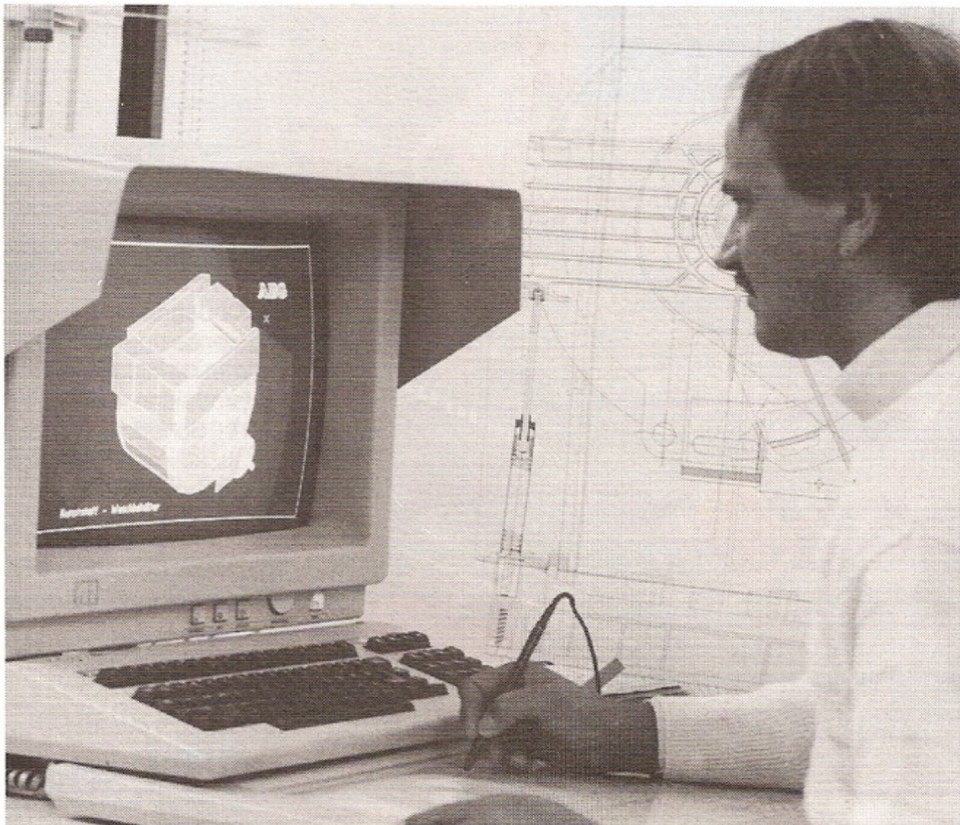


Fig. 11. De DFC bestaat uit een dataflow-gedeelte en een grafisch subsysteem.

INTEGRATORKAART

Door te werken met algoritmen op coördinatenniveau kan de uitwisseling tussen het dataflowgedeelte en de grafische modulen beperkt blijven. Het daadwerkelijk tekenen van een object – het vertalen van algoritmische beschrijvingen in pixels – gebeurt op de zogenaamde integratorkaart, een van de vier grafische modulen. Vier processoren op deze kaart verzorgen de omzetting naar vlakken, het inkleuren en het onzichtbaar maken van onderliggende elementen. Als CPU fungeert hier de IDT49C402A van Integrated Device Technology, een 16-bit bitgesegmenteerde processor met RISC-architectuur die goed is voor een piekcapaciteit van 16 Mips.

Hierna wordt het object opgeslagen in de Z-buffer, een 16-Mbyte geheugen waar de beeldpunten met diepte-informatie in komen te staan. Aan de hand van die informatie worden de achterliggende vlakken onzichtbaar gemaakt.

De integratorkaart stuurt vervolgens de volledige afbeelding naar de drie resterende grafische eenheden. Elk daarvan genereert voor iedere pixel een 8-bit waarde die respectievelijk de intensiteit van het rood, groen en blauw vertegenwoordigt. De aldus ontstane 24-bit kleurwaarde kan rechtstreeks naar een RGB-monitor (fig. 11).

KLEUREN

De drie RGB-kaarten ontvangen hun kleur-informatie direct van de dataflow-eenheden. Ook deze informatie is algoritmisch van aard en omvat voor elke veelhoek de randkleuren en de parameters om het verloop van de kleur in de X- respectievelijk Y-richting te berekenen. Alleen deze waarden, voor elk grafisch element afzonderlijk, moeten worden doorgegeven. De RGB-kaarten genereren op basis daarvan elk de feitelijke 8-bit kleurwaarden, met een maximale capaciteit van 44 miljoen beeldpunten per seconde.

Om die snelheid te behalen, loopt de verwerking via twaalf verwerkingseenheden, geordend in een systolische matrix waarin elk element steeds alleen in verbinding staat met de twee voorgaande en de twee volgende. Vier bitgesegmenteerde processoren zorgen respectievelijk voor bewerkingen als maskering en X- en Y-adressering. Acht snelle EPROM's met een cyclus van 125 ns vormen een pijplijn van functietabellen voor de ingewikkelder operaties zoals kwadrateren en worteltrekken (fig. 12).

Het systolische array staat in rechtstreekse verbinding met een videogeheugen van 4 Mbyte dat is georganiseerd als driedimensionale matrix van 128 Kvensters van elk 4×8 beeldpunten. De processoren kunnen elke instructiecyclus een kolom

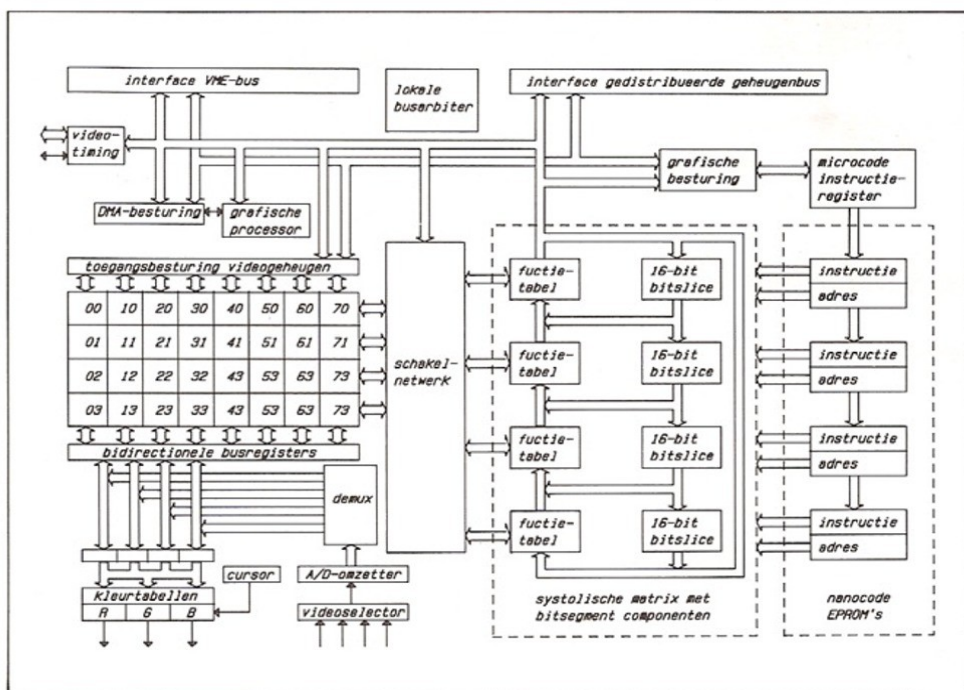


Fig. 12. Elke RGB-kaart beschikt over twaalf bewerkingseenheden voor de omzetting van kleurparameters naar beeldpunt.

- ▷ van vier pixels lezen of schrijven, resulterend in een maximale overdrachtsnelheid van 64 miljoen beeldpunten per seconde.

BLOKKEN

Het grote voordeel van deze bloksgewijze toegang is dat de relatief lage busbandbreedte per pixel (16 MHz) het gebruik van DRAM mogelijk maakt. Dit drukt de kostprijs aanzienlijk en laat ook een compacte print toe. Wanneer echter een bewerking zich uitspreidt over twee of meer blokken ontstaat er vertraging. In dat geval moet het algoritme in twee of meer onderdelen worden opgesplitst om de operatie in evenzoveel stappen uit te laten voeren. Naarmate de bewerkingen complexer zijn, komen blokoverschrijdingen vaker voor, wat de snelheid van het systeem navenant nadelig beïnvloedt.

Bij de DFC vermindert de snelheid bij ingewikkelde berekeningen niet omdat blokoverschrijdingen simpelweg uitblijven. De Vries heeft een adresseringsmechanisme ontwikkeld om een te behandelen groep beeldpunten op een willekeurige plaats binnen een blok te kunnen te situeren. Door ophoging in de X- en/of Y-richting is elke pixelgroep naar iedere gewenste locatie te brengen en zijn blokoverschrijdingen van de baan.

Deze opzet heeft een tweede voordeel, namelijk dat de aan elk blok verbonden bitsegmenteerde processoren voor eenzelfde operatie steeds dezelfde code kunnen gebruiken. Wanneer een pixelgroep niet op deze manier te manipuleren zou zijn,

moeten de verwerkingseenheden voor dezelfde operatie toch steeds verwisselen van code omdat de plaats van de te bewerken bytes is verschoven.

NYQUIST

De zelf ontwikkelde gedistribueerde geheugenbus (Distributed Memory, DM) is in hoofdzaak bedoeld voor het dataverkeer tussen de dataflowkaarten en de grafische modules (fig. 9, 11 en 12). Ofschoon met uitzondering van de communicatie tussen de drie kleurkaarten en integratoreenheid de DM-bus niet wordt belast met laag-niveau pixelverkeer, moet dit kanaal toch een hoge transmissiecapaciteit hebben om het grafische deel constant van data te voorzien. De DM-bus kan maximaal 16 miljoen 18-bit woorden per seconde aan oftewel 36 Mbyte/s.

Gegeven de bandbreedte van de bus, 8 MHz, is een overdrachtssnelheid van 16 Mwoord/s een opmerkelijke specificatie die het volgens Nyquist theoretisch haalbare maximum bereikt. Volgens dit natuurkundig theorema kunnen, wanneer de bandbreedte van een kanaal voor elektrische signalen niet verder gaat dan n MHz, er in principe $2n$ Mbit per seconde worden overgestuurd. In de praktijk ligt de transmissiecapaciteit lager en volgens DTN is de DM-bus zelfs de enige die het theoretisch maximum haalt. Hoe de ontwerpers dit varentje hebben gewassen blijft geheim. „Het spul draait, dus moet men ons wel geloven”, aldus commercieel directeur Bob Schmidt.

De snelheid op de DM-bus wordt niet nadelig beïnvloed door de diversiteit van de communicatie. Unidirectioneel of bidirectioneel, uitwisseling tussen twee, drie of acht kaarten tegelijk: iedere willekeurige combinatie laat de snelheid onverlet.

COMPILER

Een ketting is zo sterk als zijn zwakste schakel. Het werk aan paralleliserende of zelf-allocerende compilers vordert weliswaar maar nog steeds zijn hun prestaties niet indrukwekkend. „Paralleel programmeren is een kunst, geen kunde”, luidt een in vakkringen bezigde stelling. De dataafhankelijkheid laat zich in een hogere programmeertaal niet altijd even gemakkelijk vaststellen en juist dat bepaalt de mate van parallelisering: voortzetting van het programma is pas mogelijk wanneer bepaalde tussenresultaten voorradig zijn.

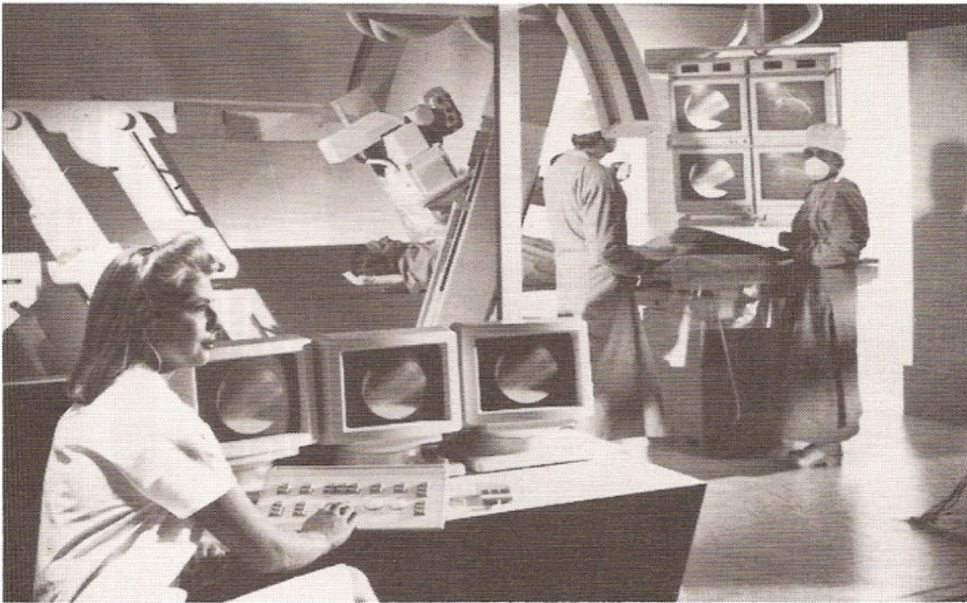
In dataflowtermen komt het erop neer dat continu moet worden nagegaan wanneer gedurende de programma-uitvoering de graaf breed is – en dus in hoge mate parallel – en wanneer smal. Een compiler zal die dataafhankelijkheid steeds nauwkeurig moeten nagaan om een maximum aan paralleliteit te bereiken.

Voor de DFC is nog geen paralleliserende vertaler beschikbaar maar momenteel is het al wel mogelijk om de volledige capaciteit van het systeem te benutten. De huidige ontwikkelprogrammatuur laat het werken op verschillende niveaus toe, ook op het voor dataflow zo typerende instructieniveau.

C

DTN stelt nadrukkelijk dat programmeurs niet overspannen zullen worden van het paralleliseren van bestaande applicatiesoftware. De transputer laat zich, als gevolg van de vereiste zend- en ontvangopdrachten en de voor optimaal rendement benodigde programmeertaal Occam, veel omslachtiger programmeren, zo stelt het bedrijf.

De onderneming twijfelt er bovendien niet aan of binnen een jaar is een zelf-allocerende C-vertaler voor de DFC op de markt. Deze verwachting is gebaseerd op de tot nu toe geboekte resultaten bij het CWI. Overigens is het niet de bedoeling om voor elke programmeertaal een aparte compiler te schrijven. Eén vertaler is genoeg en wel die voor de C-versie van het Sun-werkstation dat als besturingscomputer fungeert. Voor de rest ziet men meer in omzetting van andere standaardtalen, in eerste instantie C-dialecten, naar deze C-versie. Dit zou veel minder werk kosten dan het schrijven van afzonderlijke compilers. Verwisseling van een hogere programmeertaal wordt „simpelweg een kwestie van een andere interface”.



Medische beeldverwerking en robotzicht lijken veelbelovende toepassingsgebieden.

TOEPASSINGEN

▷ Als het goed is kunnen de gebruikers dus straks alle bestaande software op de DFC implementeren. Daarbij denkt de producent niet alleen aan grafische applicaties of daaraan gelieerde toepassingen in de CAD/CAM-sfeer. Behalve dat gebieden als scanning – vooral van belang in de medi-

sche sector – en robotzicht aantrekkelijk lijken, hoeft het bewerken van analoge signalen zich niet alleen tot videobeelden te beperken. Ook op het audiovlak lijkt DFC het nodige te kunnen bieden. De verwerkingssnelheid is hoog genoeg om vijf- tot tienduizend achter elkaar uitgesproken woorden te herkennen. Verder is het af-

handelen van communicatie via ISDN-netwerken en satellieten van belang op zowel video- als audiogebied.

Inzet bij kunstmatige intelligentie is eveneens een aantrekkelijke mogelijkheid, bijvoorbeeld voor patroonherkenning in combinatie met grote grafisch georiënteerde kennisbanken. Volgens de specificaties moet een DFC256 10 000 3D-afbeeldingen per seconde kunnen vergelijken. De vraag of Dataflow Technology Nederland daarmee de stelling bevestigt dat de hardware de software te ver vooruit spoedt, bevalt de directie niet. Men gaat er van uit dat, hoewel de software voor een dergelijk intelligent ogende machine nog niet voorhanden is, er tegen de tijd dat dergelijke toepassingen zinvol worden het programmeren daarvan veel gemakkelijker is door de aanwezigheid van een paralleliserende compiler. □

Referenties:

- [1.] N. Fokma: *Dataflow*, Databus, 1987, nummer 10, pag. 24-33.
- [2.] K. Hwang, F.A. Briggs: *Data Flow Computers and VLSI Computations*, Computer Architecture and Parallel Processing, McGraw-Hill, 1985, pag. 732-808.
- [3.] J. Herath, Y. Yamaguchi, N. Saito, T. Yuba: *Dataflow Computing Models and Machines for Intelligence Computations*, IEEE Transactions on Software Engineering, 1988, nummer 12, pag. 1805-1828.
- [4.] J.R.G. de Vries: *DFC – The Dataflow Computer*, Dataflow Technology Nederland, 1988.

HIOKI MULTIMETERS

VELE WEGEN LEIDEN NAAR

HIOKI
MEETAPPARATUUR

PROFESSIONEEL & PRAKTISCH

Op meer dan 100 verkooppunten verkrijgbaar



hartogs

- METEN EN REGELEN
- INDUSTRIËLE AUTOMATISERING
- MODULAIRE INSTALLATIETECHNIEK

Ingenieursbureau Hartogs B.V.
Strevelsweg 700/603 3063 AS Rotterdam
Tel. 010-4817833. Telex 28925. Fax 010-4808521.

TECHNI-MEUBEL B.V. TE DONGEN PRODUCEERT IN EIGEN BEDRIJF MEET- EN WERKTAFELS VOOR DE ELECTRONISCHE- EN COMPUTERINDUSTRIE, MET LEGIO MOGELIJKHEDEN EN IN ESD VEILIGE UITVOERING.



Vraagt vrijblijvend offerte en bezoek onze toonkamer aan ons bedrijf: Steenstraat 16, Industrierrein "Tichelrijt" te Dongen.

Techni-meubel B.V.

Postbus 103,
5100 AC DONGEN
tel: 01623 - 12076
fax: 01623 - 18657

TECHNI-MEUBEL B.V. UW SPECIALIST
voor ALLE VAKMEUBILAIR

