



Stichting NIOC

Stichting NIOC en de NIOC kennisbank

Stichting NIOC (www.nioc.nl) stelt zich conform zijn statuten tot doel: het realiseren van congressen over informatica onderwijs en voorts al hetgeen met een en ander rechtstreeks of zijdelings verband houdt of daartoe bevorderlijk kan zijn, alles in de ruimste zin des woords.

De stichting NIOC neemt de archivering van de resultaten van de congressen voor zijn rekening. De website www.nioc.nl ontsluit onder "Eerdere congressen" de gearchiveerde websites van eerdere congressen. De vele afzonderlijke congresbijdragen zijn opgenomen in een kennisbank die via dezelfde website onder "NIOC kennisbank" ontsloten wordt.

Op dit moment bevat de NIOC kennisbank alle bijdragen, incl. die van het laatste congres (NIOC2025, gehouden op donderdag 27 maart 2025 jl. en georganiseerd door Hogeschool Windesheim). Bij elkaar zo'n 1500 bijdragen!

We roepen je op, na het lezen van het document dat door jou is gedownload, de auteur(s) feedback te geven. Dit kan door je te registreren als gebruiker van de NIOC kennisbank. Na registratie krijg je bericht hoe in te loggen op de NIOC kennisbank.

Het eerstvolgende NIOC vindt plaats op 18 maart 2027 in Arnhem en wordt georganiseerd door HAN University of Applied Sciences.

Reacties over de NIOC kennisbank en de inhoud daarvan kun je richten aan de beheerder:

R. Smedinga kennisbank@nioc.nl.

Vermeld bij reacties jouw naam en telefoonnummer voor nader contact.

Hardware voor process-control en data-acquisitie

Recente ontwikkelingen in elektronische componenten

A.J. Borgers en W. Lourens
Faculteit Natuur- en Sterrenkunde
Werkgroep Fysische Informatica
Rijksuniversiteit Utrecht
Postbus 80.000
3508 TA Utrecht

Samenvatting

Het hoe en waarom van het gebruik van "digitale signaal processoren" en "programmable gate arrays" in data-acquisitie systemen zal worden toegelicht. Daarnaast wordt aandacht besteed aan onze keuze om de ADSP2100, resp. de Xilinx XC3020 in het onderwijs te behandelen.

1 Inleiding

De automatisering van meet- en regelsystemen geeft de mogelijkheid in korte tijd een enorme stroom aan data uit een experiment binnen te halen. Daarbij vormt de hoeveelheid gegevens die moet worden opgeslagen om achteraf te worden verwerkt steeds meer een bezwaar. Niet alleen ontstaat er een gebrek aan opslagcapaciteit, maar ook de tijd die nodig is voor de off-line data-verwerking loopt vaak in de orde van manjaren. Daarnaast blijkt pas bij de analyse van de meetresultaten, dat veel data betrekking hebben op stoorsignalen of 'oninteressante' nevenverschijnselen. Hiervan zijn er legio voorbeelden in de kernfysica, de hoge-energie-fysica, de plasmafysica en de astrofysica aan te wijzen. Vormt data-compressie nog een oplossing voor een reductie van de benodigde opslagcapaciteit, de benodigde off-line verwerkingstijd kan slechts effectief terug gebracht worden door middel van een on-line uitgevoerde selectie. Dat kan slechts door een real-time verwerking van de binnenkomende data, vaak uit meerdere kanalen afkomstig, waarbij alleen die informatie bewaard blijft, die betrekking heeft op het te bestuderen verschijnsel. Deze vorm van selectie moet door snelle multi-processor-systemen of dedicated hardware uitgevoerd worden. Daar de beschikbare tijd ligt tussen ruwweg 100 ns en 100 μ s, afhankelijk van de uit te voeren discriminatie, is het

meestal onmogelijk gebruik te maken van standaard micro-processoren, die hier te kort schieten in snelheid en capaciteit. Daarentegen is het met digitale signaal processoren (DSP's) en programmable gate arrays (PGA's) wel mogelijk modules te bouwen die voldoende snel zijn om in real-time een eerste, intelligente verwerking van de data te realiseren. Zowel DSP's, als PGA's, zijn 'software programmable'. Gaat men daarbij uit van een logic cell array (LCA) van Xilinx als PGA, dan kan men bovendien gebruik maken van het feit, dat het ontwerp in een LCA na configureren niet definitief is, maar software-herprogrammeerbaar blijft. Dit geeft extra flexibiliteit; raakt tijdens het experiment bijvoorbeeld een van de detectoren defect, dan kan on-line worden gereageerd door het opnieuw, aangepast configureren van de hardware. De genoemde voordelen gelden ook voor regelsystemen, waar on-line en in real-time data moeten worden verwerkt om op tijd beslissingen voor process control te kunnen nemen.

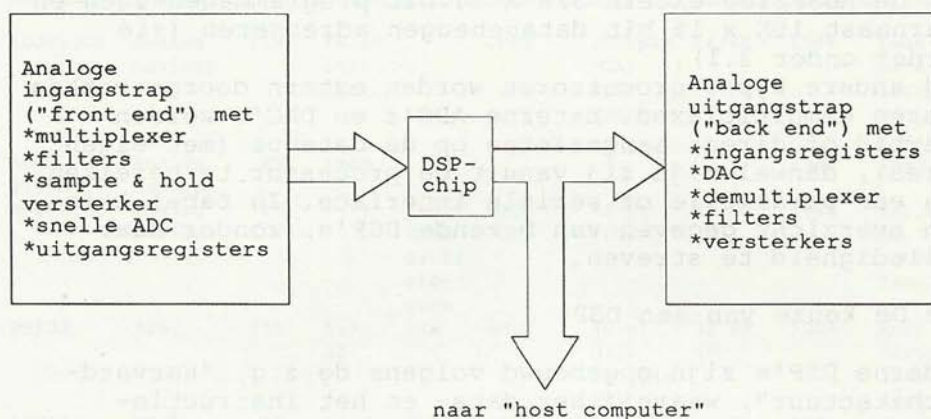
Natuurlijk biedt het gebruik van applicatie specifieke IC's (ASIC's) ook een oplossing, maar met als nadeel dat de ontwikkelings- en productiekosten erg hoog zijn. Dit is een bezwaar bij toepassingen op kleine schaal, zoals vaak het geval is in een laboratorium of het onderwijs. Het is o.i. een eerste vereiste, dat R & D medewerkers die zich bezighouden met experimenten, waarin snelle, digitale signaalverwerking een essentiële rol speelt of kan gaan spelen, op de hoogte zijn van de mogelijkheden, die moderne elektronische componenten te bieden hebben. Daarom achten wij het van belang dat bijvoorbeeld HBO-studenten telematica, en WO-studenten natuurkunde met Fysische Informatica in hun vakkenpakket, tijdens hun opleiding ervaring opdoen met data-acquisitie systemen rond DSP's en PGA's als centrale bouwstenen.

2 Digitale signaalprocessoren (DSP's)

2.1 De rol van de DSP in een data-acquisitie systeem

Bij digitale signaalverwerking komen bewerkingen als vermenigvuldigen en optellen, afronden en het werken met databuffers veel voor. Voorbeelden zijn digitale filters, waarbij nieuwe en opgeslagen samples met coëfficiënten worden vermenigvuldigd en opgeteld om een uiteindelijk sample voor output te verkrijgen. Vandaar dat een vermenigvuldiger & opteller (MAC) en een arithmetisch/logic unit (ALU) als rekenkundige functieblokken, essentiële onderdelen van elke DSP zijn. Analoge signalen worden ingelezen via een analoog-naar-digitaal omzetter (ADC). Het aldus ontstane digitale signaal wordt om optreden van aliage-fouten te vermijden zonodig door een laagdoorlaat filter geleid. Het signaal

wordt vervolgens bewerkt met behulp van een DSP, waarna de resultaten worden afgegeven aan een host of aan een digitaal-naar-analoog omzetter (DAC) - zie figuur 1 - .



Figuur 1
Schematische opbouw van een DSP-systeem

2.2 DSP-chips

De eerste doelgericht ontworpen DSP was de Intel 2920, geïntroduceerd rond 1980. De 2920 had een vier-kanaals ADC, evenals een acht-kanaals DAC on-chip. Daarnaast beschikte de 2920 over een 24-bit processor met een on-chip geheugen van 192 x 24-bit ROM-woorden en 40 x 25-bit RAM-woorden. De processor kende geen hardware-vermenigvuldiger. Bij gebruik in de telecommunicatie, was de chip geschikt om er een full duplex 1200 baud modem in te implementeren, inclusief de filters voor zenden en ontvangen.

Latere DSP's missen de analoge in- en uitgangssecties, waardoor de gebruiker niet langer gebonden is aan de beperkingen, die een on-chip A/D en D/A conversie hem oplegt. Zo kwam er ruimte vrij voor meer geheugen. Tevens kwam de nadruk te liggen op een snellere bit-manipulatie en op de mogelijkheid snel floating-point operaties uit te voeren. Immers de snelheid waarmee de data worden verwerkt, bepaalt de snelheid waarmee aangeboden analoge signalen kunnen worden bemonsterd. De verschillende types processoren onderscheiden zich verder door verschillen in woordbreedte, en door de hoeveelheid geheugen op de chip. Het ene uiterste wordt gevormd door de populaire NEC μ PD7720, die zijn gehele programma en data geheugen (inclusief een blok van data-EPROM voor coëfficiënten) aan boord heeft; daarmee komt de noodzaak om de interne parallelle bussen extern door te trekken, te vervallen, wat het benodigde aantal

aansluitpennen voor de chip aanzienlijk verkleint. Tot het andere uiterste behoort de Analog Devices ADSP2100, die slechts een cache-geheugen voor 16 instructies op de chip heeft. Via de extern doorgetrokken interne bussen kan de ADSP2100 extern 32K x 24 bit programmeergeheugen en daarnaast 16K x 16 bit datageheugen adresseren (zie verder onder 3.1).

Bij andere types processoren worden extern doorgetrokken bussen gemultiplexed. Externe ADC's en DAC's worden daarbij of direct aangesloten op de databus (met eigen adres), danwel zijn zij vanuit de processor te bereiken via een parallelle of seriële interface. In tabel 1 is een overzicht gegeven van bekende DSP's, zonder naar volledigheid te streven.

2.3 De keuze van een DSP

Moderne DSP's zijn opgebouwd volgens de z.g. "Harvard-architectuur", waarbij het data- en het instructie-geheugen gebruik maken van aparte bussen. Zo wordt de bottleneck van de enkele bus in de Von Neumann architectuur vermeden. Het gebruik van parallelle bussen verkleint de benodigde verwerkingstijd per instructie. Sommige types processoren kennen daarbij ook nog de mogelijkheid van een functionele uitwisseling tussen de bussen onderling.

Met de DSP's hebben ontwerpers nieuwe mogelijkheden voor een snelle en flexibele signaalverwerking verkregen. De vraag naar een steeds grotere verwerkingscapaciteit en daarbij het verlangen naar steeds meer on-chip geheugen, gaven echter alras aanleiding tot problemen.

Bij de single-chip DSP moet de beschikbare ruimte worden verdeeld tussen verwerkingslogica en systeemgeheugen. Slechts zolang de combinatie van logica en geheugen kan voldoen aan de eisen die het programma stelt, kan een dergelijke processor optimaal werken. Een uitbreiding van de benodigde geheugenruimte door het inschakelen van een extern geheugen, vraagt een access-tijd die meestal groter is dan die tot het on-chip geheugen. Dit is zeker het geval, als de interne parallelle bussen niet rechtstreeks, maar via een multiplexer naar buiten doorgetrokken zijn. Als dan via deze bussen ook nog het I/O-verkeer moet worden afgewikkeld, blijft er van de voordelen van een interne Harvard-architectuur weinig over. Gelukkig zijn programma's voor real-time dataverwerking meestal klein, zodat dan met weinig on-chip instructiegeheugen kan worden volstaan. Bij deze toepassingen vormt de multiplexing geen bezwaar.

Is men toch geneigd om te zien naar een DSP, waarbij de interne, parallelle bussen rechtstreeks naar buiten worden gevoerd, dan verkrijgt men al gauw een chip, die door het aantal aansluitpennen 'onhandelbaar' wordt.

Processor	Fabr.	Cycl. tijd (ns)	On-chip geheugen ROM RAM		Techno- logie	Package	Data bits	Jaar	Opm.
ADSP2100	Analog Devices	125	16x24 inst cache		CMOS	100-pin PGA/ PLCC (PQFP)	16/40	1986	twee paral- lelle I/O bussen
ADSP 2100A	Analog Devices	80	idem					1988	idem
ADSP2101	Analog Devices	80	-----	1kx16 data 2kx24 pro- gram	CMOS	68-pin PLCC	24	1988	seriële I/O inter- face
DSP32	AT&T	250	512x 32	1kW	NMOS	40-pin DIP/ 100-pin PGA	32 FP 40	1986	non- harvard archi- tecture
DSP32C	AT&T	80	2048x 32	1536x 32	CMOS	133-pin PGA	32 FP 32/40	1988	seriële I/O
DSP56000	Motorola	100	2kx24 + 512x 24	256x 24 + 256x 24	CMOS	88-pin PGA/ PLCC	24/56	1986	seriële I/O inter- face
DSP96002	Motorola	74	1088x 32	2048x 32	CMOS	214-pin PGA	32 FP 32/96	1988	twee paral- lelle I/O bussen
LM32900	National Semicon- ductor	100	-----	-----	CMOS	172-pin PGA/ PLCC	16/32	1986	2 I/O bussen + ser. inter- face
μPD7720	NEC	250	512x 23	128x 16	NMOS	28-pin DIP	16	1981	ser. int. only
μPD77230	NEC	150	3kx32	1kx32	CMOS	68-pin PGA	32 FP 32/55	1986	seriële I/O
TMS32010	Texas Instru- ments	320	1536x 16	144x 16	NMOS	40-pin DIP/ 44-pin PLCC	16/32	1982	
TMS32020	TI	200	-----	544x 16	NMOS	68-pin PGA	16/32	1985	seriële interf.
TMS 320C25	TI	100	4kx16	512 W	CMOS	68-pin PLCC	16/32	1986	seriële interf.
TMS 320C30	TI	60	64x32 cache 4kx32	inst cache 2kx32	CMOS	180-pin PLCC	32 FP 32/40	1988	seriële I/O interf.
TMS 320C50	TI	50	-----	8k7				1989	fixed- point

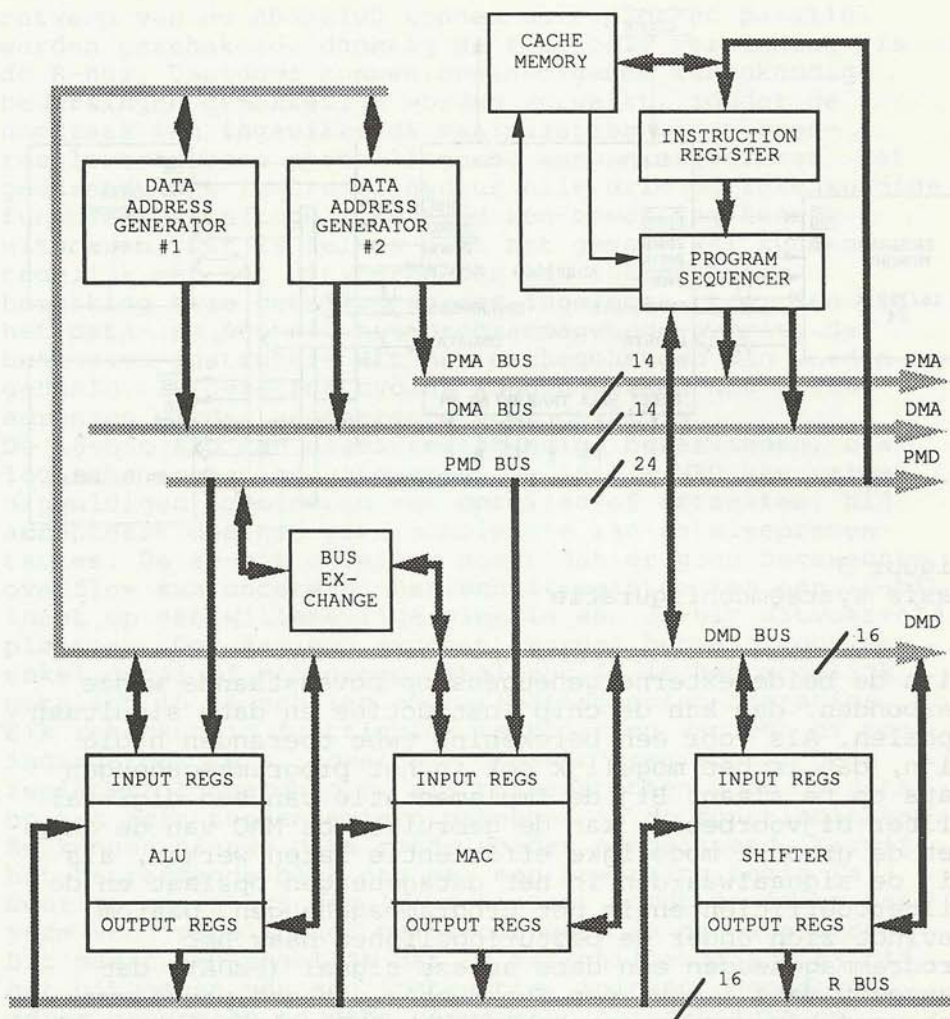
Tabel 1
Beknopt overzicht bekende DSP's

Dit is zeker het geval voor een hardware-systeem, dat men met eenvoudige middelen binnen een laboratorium zelf wil ontwerpen. Het zelf ontwerpen verlangt daarbij het beschikbaar zijn van een gebruikersvriendelijk software ontwikkel- en simulatie-pakket bij de gekozen DSP.

3 De ADSP2100

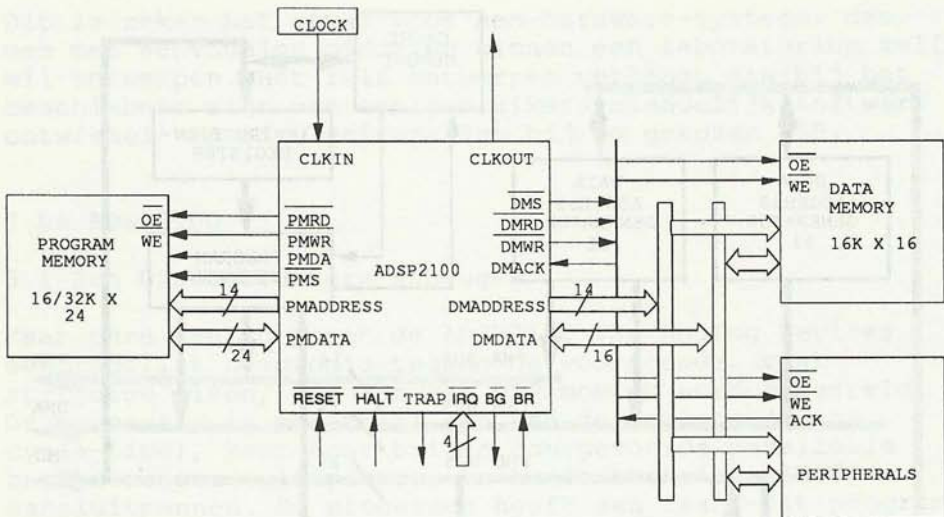
3.1 Een DSP met extern geheugen

Naar onze mening vormt de ADSP2100 van Analog Devices een redelijk compromis tussen de voornoemde, vaak strijdige eisen, die aan een DSP moeten worden gesteld. De processor is momenteel één van de snelste (80 ns cycle-time), kent naar buiten doorgevoerde parallele bussen zonder multiplexer, en heeft toch 'maar' 100 aansluitpennen. De processor heeft een 16x24-bit program cache, waardoor de instructiebus in vele gevallen ook voor data in- en output gebruikt kan worden. Daarnaast ondersteunt de fabrikant de processor met plezierig bruikbare ontwikkelprogrammatuur, die draait op een IBM-PC of een DEC-VAX. Deze programmatuur bestaat uit een system builder, assembler, linker, simulator, (P)ROM splitter en C compiler. De assembler kent niet de gebruikelijke mnemonic codes, maar de syntaxis laat het wel toe codes te schrijven in een 'algebraïsche' notatie. Dit houdt de programma's redelijk leesbaar. De ontwerpers van Analog Devices gingen er kennelijk vanuit, dat het plaatsen van zowel het programma-, als het datageheugen buiten de chip niet langer bezwaarlijk zou zijn, mits dit gepaard zou gaan met een toename van de verwerkingscapaciteit; daarbij mocht de accesstijd tot het (nu externe) geheugen niet toenemen. De hogere verwerkingscapaciteit is bereikt door in de vrij gekomen ruimte op de chip extra schakelingen voor besturing en signaalverwerking op te nemen. Naast de gebruikelijke MAC, een ALU en een schuifregister, beschikt de ADSP2100 over twee onafhankelijke data-adresgeneratoren (DAG's). Zij dragen bij tot het verhogen van de verwerkings-snelheid, omdat het bijwerken van adressen binnen een buffer nu parallel kan gebeuren met andere, met name rekenkundige, bewerkingen. De besturingseenheid met cachegeheugen zorgt er voor, dat korte programmalussen kunnen worden uitgevoerd zonder onnodige overhead cycli. Bovendien kent de interne opbouw van de ADSP2100 - zie figuur 2 - vijf bussen, waardoor de interne informatie-overdracht snel kan geschieden. Complexe instructies kunnen zo in één verwerkingsslag worden uitgevoerd. Vier bussen zijn extern doorgetrokken. De uitvoer-registers van de rekenkundige functieblokken zijn via een R-bus (result-bus) met elkaar verbonden. Zo kunnen de resultaten van een berekening direct voor een



Figuur 2
Interne architectuur van de ADSP2100

volgende verwerkingsslag in hetzelfde of in een ander functieblok worden gebruikt, zonder dat daarvoor een tussenopslag in het externe geheugen nodig is. Door de interne bussen voor data en instructies naar buiten door te trekken, wordt de hoeveelheid beschikbaar geheugen niet bepaald door de interne structuur, maar kan deze aan de toepassing worden aangepast. In de basisconfiguratie - zie figuur 3 - is de ADSP2100 verbonden met twee externe geheugens, elk met zijn eigen adres-, data- en besturingslijnen. Zodoende kan de DSP 16K x 24-bit programmeergeheugen en 16K x 16-bit data-geheugen adresseren. Wel moeten deze geheugens snel zijn!



Figuur 3
Basis systeemconfiguratie

Zijn de beide externe geheugens op bovenstaande wijze verbonden, dan kan de chip instructies en data simultaan ophalen. Als voor een berekening twee operanden nodig zijn, dan is het mogelijk ook in het programmeergeheugen data op te slaan. Bij de implementatie van een digitaal filter bijvoorbeeld, kan de gebruiker de MAC van de chip met de grootst mogelijke efficiëntie laten werken, als hij de signaalwaarden in het datageheugen opslaat en de filtercoëfficiënten in het programmeergeheugen. Daarom bevindt zich onder de besturingslijnen naar het programmeergeheugen een data access signal (PMDA), dat aangeeft data i.p.v. een instructie uit het programmeergeheugen kan worden opgehaald. Het PMDA-sigitaal kan ook worden gebruikt als 15e adresbit. Data-overdracht met het datageheugen is onderworpen aan een handshake-protocol, om ook langzame I/O-chips in staat te stellen data aan te leveren. Verder is de processor voorzien van busrequest- en busgrant-lijnen, welke toepassing in een multiprocessor-omgeving mogelijk maken. Er zijn vier interruptlijnen (IRQ) voor het synchroniseren van de programmatuur en voor het snel verwerken van speciale gebeurtenissen. De IRQ-afhandeling heeft zero-overhead dankzij een set schaduw-registers, die automatisch in werking kunnen treden als een IRQ afgehandeld wordt.

3.2 Rekenkundige functieblokken

Drie rekenkundige functieblokken (ALU, MAC en schuif-register) zorgen voor het bewerken van de data. In het

ontwerp van de ADSP2100 konden deze blokken parallel worden geschakeld, dankzij de flexibele verbinding via de R-bus. Daardoor kunnen opeenvolgende rekenkundige bewerkingen gemakkelijk worden verwerkt, zonder de noodzaak van ingewikkelde manipulaties met tussenresultaten. Deze opstelling zou kunnen suggereren, dat gedurende één instructiecyclus alle drie de rekenkundige functieblokken tegelijkertijd een bewerking kunnen uitvoeren. Dit is helaas niet het geval. Wel kunnen tegelijk met het uitvoeren van een rekenkundige bewerking twee getallen worden ingelezen, t.w. één uit het data- en één uit het programmeergeheugen, mits de betrokken instructie uit het cachegeheugen kan worden gehaald. In dezelfde cyclus kunnen tevens nog nieuwe adressen worden gegenereerd door de DAG's.

De 16-bit ALU kan naast rekenkundige bewerkingen, ook logische operaties uitvoeren. De 16-bit MAC kan vermenvuldigen combineren met optellen of aftrekken; hij accepteert daarbij elke combinatie van getalrepresentaties. De 40-bit opteller zorgt dat er geen ongewenste overflow kan ontstaan. Het schuifregister kan een 16-bit input op een willekeurige plek in een 32-bit uitvoerveld plaatsen. Ook kan een exponent worden bepaald voor een enkel getal of een groep getallen. Zo is het mogelijk getallen in te schalen voor floating-point operaties. Elk rekenkundig functieblok is voorzien van één of meer ingangsregisters, zowel als "result"-registers. Deze registers fungeren o.m. als tussenstation voor de data op hun reis tussen extern geheugen en de functieblokken. Zo kunnen nieuwe data reeds worden aangeleverd, terwijl het betreffende blok nog met een bewerking bezig is. Door de registers gevuld te houden, wordt in feite een vorm van "pipelining" geïntroduceerd in de datastroom. Dit maakt het mogelijk dat de verschillende acties bij het uitvoeren van een instructie, inlezen, decoderen, adres genereren en data verwerken, gelijktijdig op kunnen treden.

3.3 Overzicht van de instructieset

De instructies van de ADSP2100 kunnen worden ingedeeld in vier basis categorieën - zie tabel 2 - . De set is toegesneden op specifieke DSP-algoritmes. Alle instructies kunnen rechtstreeks 16-bit bewerkingen uitvoeren in één machinecyclus. In de instructieset vinden we geen instructies die meerdere cycli in beslag nemen. Wel zijn er "multifunction operations", die ons in staat stellen in één instructiecyclus tot vijf verschillende operaties tegelijk uit te voeren, waaronder één rekenkundige bewerking mag zijn.

Bij de programmabesturing vinden we naast bijvoorbeeld JUMP en CALL ook een DO...UNTIL loop instructie, met

Lees- en schrijfoopdrachten ("move instructions")

```

register <-> register
register <-> datageheugen
register <-> programmegeheugen
direct wegschrijven van een getal -> register

```

Rekenkundige bewerkingen, al dan niet gecombineerd met lees- en schrijfoopdrachten ("multifunction operations")

```

ALU-, MAC-, of schuifoperatie,
    afhankelijk van de inhoud van de statusregisters
ALU-, MAC-, of schuifoperatie,
    gecombineerd met register <-> register
ALU-, MAC-, of schuifoperatie,
    gecombineerd met register <-> datageheugen
ALU-, MAC-, of schuifoperatie,
    gecombineerd met register <-> programmegeheugen
ALU-, of MAC-operatie,
    gecombineerd met datageheugen -> register, en
    programmegeheugen -> register

```

Programmabesturing ("program flow control instructions")

jump	alle deze
subroutine call	instructies al
return	dan niet
na uitvoeren subroutine, of	afhankelijk
afhandelen interrupt	van de actuele
trap	inhoud van de
do...until	statusregisters

Diverse andere instructies

```

verzadiging van de accumulator
stack-control
mode-control
no operation ("NOP-operations")

```

Tabel 2

Beknopt overzicht instructieset ADSP2100

daarnaast de mogelijkheid om de meeste rekenkundige bewerkingen onder voorwaarde te laten verrichten. Al met al is de instructieset redelijk flexibel. Het is echter wel zaak een goed inzicht te hebben in de opbouw van de ADSP2100, teneinde te kunnen begrijpen waarom bepaalde instructies wel of niet mogelijk zijn. Welke registers wel of niet als een bepaalde operand kunnen optreden, wordt bepaald door de interne verbindingen in de processor.

4 Programmable gate arrays (PGA's)

4.1 De LCA's van Xilinx

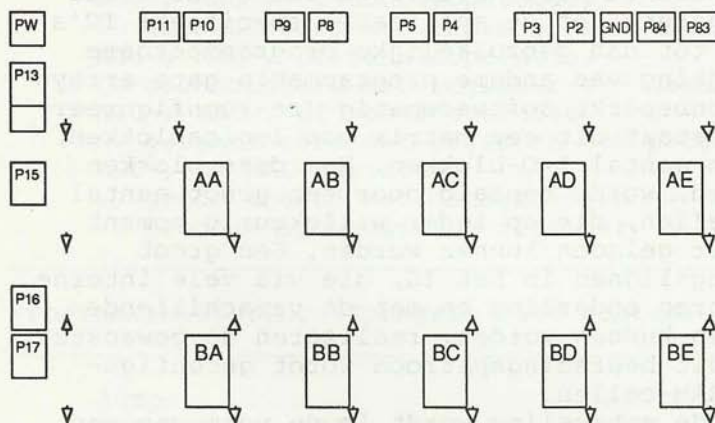
In 1985 lanceerde Xilinx de Logic Cell Array (LCA). Met dit IC-concept werd de kloof overbrugd tussen de vaste gate arrays, behorend tot de applicatie specifieke IC's (ASIC's), en de tot dan gebruikelijke programmeerbare logica. In afwijking van andere programmable gate arrays blijft een LCA onbeperkt softwarematig (re-)configureerbaar. Een LCA bestaat uit een matrix van logicablokken, omringd door een aantal I/O-blokken. Hoe deze blokken zijn gedefinieerd, wordt bepaald door een groot aantal statische RAM-cellen, die op ieder willekeurig moment door de gebruiker geladen kunnen worden. Een groot aantal verbindingslijnen in het IC, die via vele interne schakeltransistoren onderling en met de verschillende blokken verbonden kunnen worden, realiseren de gewenste bedrading. Ook dit bedradingspatroon wordt geconfigureerd door de SRAM-cellen.

Het ontwerp van de schakeling wordt in de vorm van een bitpatroon opgeslagen in bijvoorbeeld een EPROM. Bij inschakelen van de hardware wordt het ontwerp in de vorm van een configuratie-bitstroom via een down-load verbinding vanuit het opslagmedium in de LCA geladen. Dit komt neer op het laden van SRAM-cellen, en duurt diensgevolge slechts een tiental milliseconden. Ook herprogrammeren van de LCA is dus in een dergelijk tijdsbestek mogelijk. Daardoor wordt het gebruik van LCA's in een laboratoriumomgeving aantrekkelijk, omdat het mogelijk is de inhoud van het IC, en daarmee het functioneren van een stuk hardware in een opstelling, zelfs gedurende een experiment snel te kunnen wijzigen. De gebruikersvriendelijke ontwikkelsoftware, maakt het ontwerpen van schakelingen met een LCA vrij eenvoudig.

4.2 LCA-chips

Xilinx produceert thans drie verschillende LCA-families, t.w. de XC2000, 3000 en 4000 serie, alle in CMOS (zie Meijer 1987 en Brinkman 1990).

De XC2064 en de XC2018 hebben 64, resp. 100 definieerbare logicablokken met elk vier logische ingangen. De logicablokken zijn gerangschikt in een matrix, omringd door 58 I/O-blokken bij de XC2064 en 74 bij de XC2018. De XC2000-familie is verkrijgbaar in vier uitvoeringen voor schakelfrequenties van resp. 33, 50, 70 en 100 MHz. De XC3000-familie kent een vijftal telgen, waarbij het aantal logicablokken loopt van 64 tot 320, en het aantal I/O-blokken van 64 tot 144. Als voorbeeld staat in figuur 4 een gedeelte van de matrixstructuur van de XC3020PC84 afgebeeld. De logicablokken van de XC3000's hebben vijf logische ingangen en bovendien een andere interne opbouw. De XC3000's zijn verkrijgbaar in uitvoeringen tot 125 MHz. De LCA's uit de XC4000-familie zijn uitgebreider; zij blijven hier buiten beschouwing.

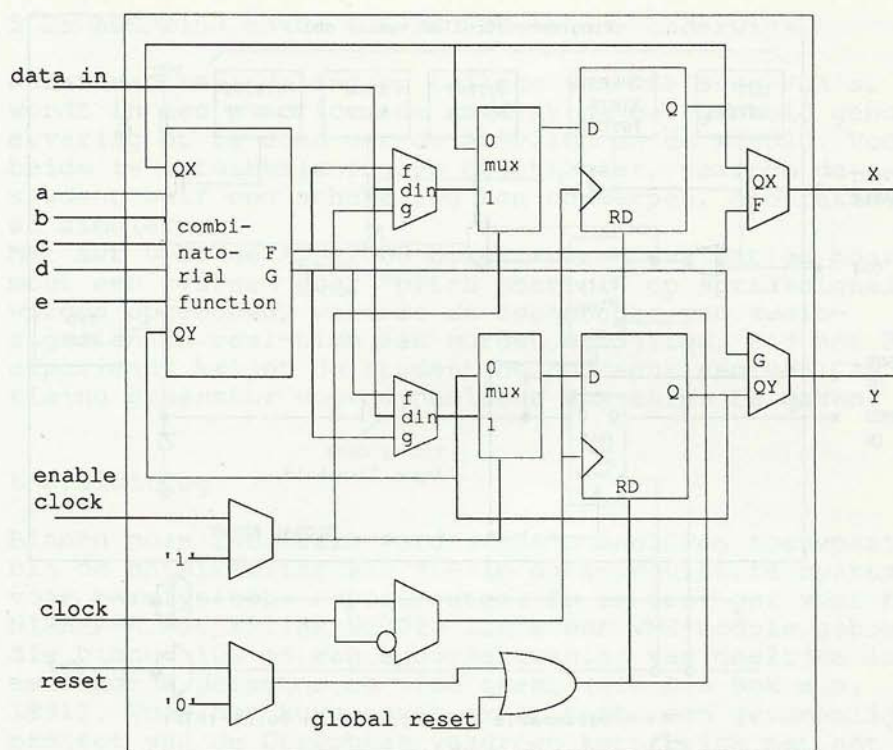


Figuur 4

Detail van de matrixstructuur van een LCA

4.3 Logicablokken

De logicablokken, Configurable Logic Blocks (CLB's), vormen de kern van een LCA. Elk blok kent een aantal logische ingangen, een klokingang, een aantal besturingslijnen en twee uitgangen - zie figuur 5 -. De logische ingangen zijn verbonden met een blok combinatorische logica. Dit blok is configureerbaar door het specificeren van de logische functie met een booleaanse vergelijking, een waarheidstabel, of een karnaugh-diagram. Zijn niet alle ingangsvariabelen nodig voor het specificeren van de functie, dan is het mogelijk twee functies F en G tegelijk in het logicablock op te nemen. Elk blok herkent en genereert zowel positieve als negatieve logica, waardoor het gebruik van interne inverters overbodig wordt.

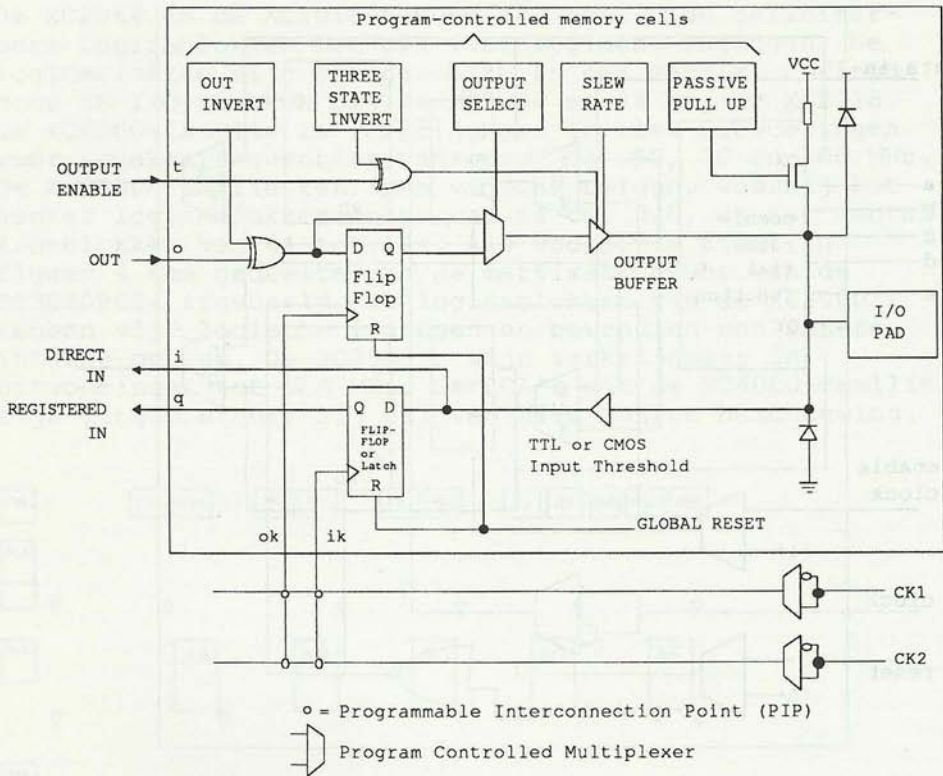


Figuur 5
CLB van een LCA uit de XC3000-familie

Het is mogelijk een flipflop-uitgang Q binnen het CLB terug te koppelen als variabele in de functies F en G van het blok combinatorische logica. De twee uitgangen X en Y van een logicablok kunnen onafhankelijk van elkaar geprogrammeerd worden, zodat ze de uitgang F of G, dan wel de uitgang Q van een flipflop representeren.

4.4 I/O-blokken

Een I/O-blok heeft een externe aansluitpen, die naar keuze configureerbaar is als signaalingang, -uitgang, of als bidirectionele pen. Het signaalniveau is instelbaar voor TTL of CMOS; bij gebruik als uitgang kan de pen de status van een hoge impedantie krijgen ten behoeve van tristate-toepassingen. Als ingang heeft elk I/O-blok binnen de XC2000-familie een flipflop om binnenkomende signalen desgewenst te synchroniseren. Een flipflop kan ook dienen als lees-/schrijfregister of als onderdeel van een schuifregister. De flipflops van de I/O-blokken aan dezelfde kant van de chip zijn op één kloklijn aangesloten. Zo kent het IC vier verschillende kloklijnen die elk een intern of extern kloksignaal kunnen voeren.



Figuur 6
I/O-blok van een LCA uit de XC3000-familie

De XC3000-familie kent een uitgebreidere opbouw van de I/O-blokken in vergelijking met de XC2000-familie, waaronder twee flipflops i.p.v. één; - zie figuur 6 - .

4.5 Bedradingspatroon

Het configureerbare bedradingspatroon kent een orthogonaal stelsel van bedradingselementen die drie soorten verbindingen kunnen realiseren, t.w. via rechtstreekse verbindingslijnen tussen de blokken, via verticale en horizontale lijnen tussen rijen en kolommen van de blokken, en via over het gehele IC-kristal doorlopende verticale en horizontale "lange lijnen". De rechtstreekse verbindingen zijn kort en snel, maar ook de lange lijnen kunnen signalen met een geringe tijdsvertraging over juist grote afstanden binnen de chip transporteren. Lijnen tussen de rijen en de kolommen van de blokken worden aan elkaar gekoppeld via apart configureerbare schakelmatrixes. De matrixes, die een reeks bidirectionele schakelaars bevatten, koppelen deze lijnen tot een orthogonaal verbidingsnetwerk.

5 De ADSP2100 en de LCA XC3020 in ons onderwijs

Naast een behandeling op college van DSP's en PGA's, wordt in een practicum de student de gelegenheid geboden ervaring op te doen met de ADSP2100 en de XC3020. Voor beide is ontwikkelsoftware beschikbaar, waarmee de student zelf een schakeling kan ontwerpen, doorrekenen en simuleren.

Met het voor de ADSP2100 beschikbare "evaluation board" moet een systeem voor "pitch control" op spraaksignalen worden opgebouwd, waarmee de toonhoogte van audio-signalen in real-time kan worden gewijzigd. Bij het PGA-experiment krijgt de student de opdracht een RAS/CAS timing generator voor dynamische RAM-chips te maken.

6 Afsluiting

Binnen onze faculteit worden PGA's en DSP's toegepast bij de ontwikkeling van snelle data-acquisitie systemen voor kernfysische experimenten. Zo is door ons voor het NIKHEF-K met Xilinx XC3020 LCA's een VME-module gebouwd, die binnen 100 ns een spoorherkenning van deeltjes in een hadron detector op zich neemt (zie Den Bok e.a. 1991). Voor het Huygensvat-experiment, een gezamenlijk project van de Utrechtse vakgroep kernfysica met het Kernfysisch Versneller Instituut van de RUG, is door ons een real-time pulsvorm analysator in ontwikkeling, waarin zowel de XC3020 LCA, als de ADSP2101 (een DSP, afgeleid van de ADSP2100) een centrale rol spelen. Zoals reeds eerder betoogd, verwachten wij een toename in de toepassing van intelligente modules, gebaseerd op DSP's en PGA's, in systemen die veel data moeten verwerken of die real-time moeten reageren op data vanuit meerdere kanalen. Reden, waarom wij binnen onze faculteit rond dit soort componenten een stuk onderwijs opgebouwd hebben voor studenten in de richting experimentele natuurkunde met als specialisatie of als keuzevak Fysische Informatica; daarnaast verzorgen wij sinds najaar 1990 eveneens een cursus in het kader van het Post Academisch Onderwijs Natuurwetenschappen.

Gebruikte literatuur

Brinkman, M. (1990) Evolutie van de Logic Cell Array.

In: Elektronica 90/17, 43-53.

Den Bok, H.W., J.L. Viisschers, A.J. Borgers & W. Lourens (1991) Track recognition with an associative pattern memory. In: Nuclear Instruments and Methods in Physics Research A300, 107-114.

Meijer, J.C. (1987) Ontwerpen met Logic Cell Arrays. In: Elektronica 87/17, 49-61.