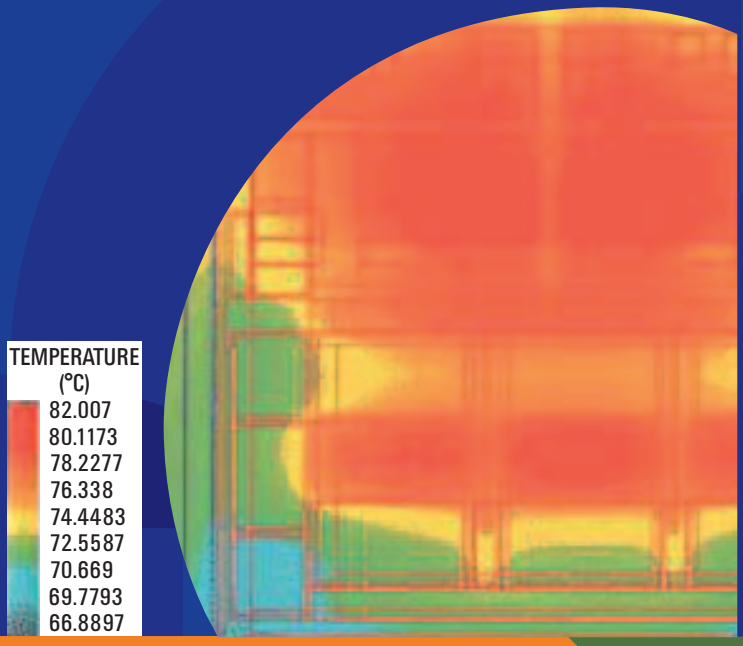
Universiteit Twente
de ondernemende universiteit



Energie in Computer Architectuur
less is more

door Prof.dr.ir. G.J.M. Smit

Energie in Computer Architectuur
less is more

Rede uitgesproken bij
het aanvaarden van het ambt
van hoogleraar

Computer Architectures for Embedded Systems

aan de Faculteit Elektrotechniek,
Wiskunde en Informatica
van de Universiteit Twente
op donderdag 22 november 2007
door

Prof.dr.ir. G.J.M. Smit

Mijnheer de Rector Magnificus, Dames en Heren,

Inleiding

Voor mijn oratie heb ik de titel “Energie in Computer Architectuur - less is more” gekozen. In de komende 45 minuten zal ik proberen duidelijk te maken waarom naar mijn mening deze titel bij de leerstoel Computer Architectuur voor Embedded Systemen (CAES) past. Ik zal ingaan op de problemen, uitdagingen en mogelijke oplossingen van dit vakgebied.
Om dit allemaal in zijn perspectief te zien zal ik eerst door middel van een stukje historie iets vertellen over het belang van energie in computer architectuur.
Ik zal ingaan op de wetmatigheden die er gelden om te komen tot efficiëntere computer architecturen. Ik zal beschrijven hoe we op dit moment al werken aan oplossingen en hoe die de komende jaren nog verder uitgebouwd kunnen worden. We zullen zien dat energie van een onbelangrijk zijeffect bij computer architecturen is uitgegroeid tot een echte barrière voor toekomstige systemen. Daarna zal ik mijn visie geven op het onderwijs.

Probleem

De naam van mijn leerstoel luidt “Computer Architecture for Embedded Systems”. Voor degene die niet bekend zijn met de term embedded systemen even een korte uitleg. Embedded systemen zijn apparaten die er aan de buitenkant niet uitzien als een computer (ze hebben bijvoorbeeld geen beeldscherm of toetsenbord); maar in hun inwendige wel degelijk computer technologie (ICT technologie) bevatten. U kunt bijvoorbeeld denken aan een TV, videorecorder, airbag in een auto, MP3 speler, wasmachine, CV ketel etc.
Mijn leerstoel richt zich op computer architecturen voor dergelijke systemen; in het bijzonder richt de groep zich op embedded systemen waarbij energie van belang is.
Energie is een bijzonder fenomeen. Je realiseert je vaak pas dat iets energie verbruikt als je er niet genoeg meer van hebt. In computersystemen is energieverbruik dan ook lang een onbetekenend ontwerp criterium geweest.
In de laatste 10 jaar is daar verandering in gekomen en is ook in onderzoek energie een belangrijke factor geworden.
Voordat we op het onderzoek ingaan even wat basis informatie, wellicht om

uw natuurkunde kennis wat op te frissen. Energie (in Joules) is de integraal van het vermogen over de tijd. Als het vermogen constant is dan is de energie vermogen maal tijd. Vermogen en energie worden vaak verward. Meer vermogen hoeft niet per definitie tot meer energieverbruik te leiden. Een vermogen van 1 W gedurende een seconde leidt tot minder energie verbruik dan een vermogen van 0.5 W gedurende 4 seconden. In mijn presentatie zal ik het met name over energie hebben.
Ik onderscheid drie redenen waarom onderzoek gedaan wordt naar het energieverbruik van computers:

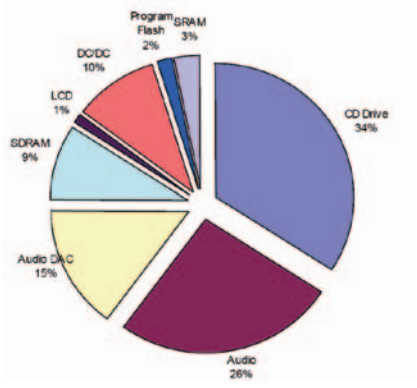
1) voor systemen die voor de energievoorziening afhankelijk zijn van een batterij,
2) voor systemen die problemen hebben met hun warmtehuishouding en
3) uit milieu overwegingen.

Doordat computers kleiner worden komen er ook steeds meer apparaten op de markt die op een batterij werken; denk aan MP3 spelers, mobiele telefoons, elektronische agenda's. Het ligt natuurlijk voor de hand dat energiezuinige computers hier van groot belang zijn. Toen Paul Havinga en ik 10 jaar geleden aan energiezuinige systemen begonnen te werken, was dit het hoofddoel van ons onderzoek. We waren toen pioniers op dit gebied; voor veel onderzoekers was energie toen nog een marginale zaak.
Als voorbeeld neem ik de iPod nano. In de praktijk is het maximale toelaatbare vermogen van een chip zonder geforceerde koeling ongeveer 5 Watt. De iPod nano heeft een batterij met een energie inhoud van 1.2 Wuur. Bij een opname van 5 Watt zou de iPod dus slechts 15 minuten kunnen spelen. Bovendien is meer dan 5 W 's zomers niet erg comfortabel in je broekzak. De specificatie zegt dat de iPod nano 14 uur muziek zou moeten leveren zonder tussentijds op te laden. Dit betekent dat er zuinig met energie moet worden omgesprongen. Dus het gemiddelde vermogensbudget van het hele systeem is 1.2/14 = 85mW. De iPod maakt gebruikt van een dual core ARM7 processor die op maximaal 80 MHz draaien (dus geen GHz). Om MP3 te decoderen hoeven ze niet eens op volle snelheid te lopen. Ze kunnen zo binnen het gemiddelde energie budget van 85mW blijven.
Naast batterij gevoede apparaten zien we de laatste jaren steeds grotere vraag naar energie-efficiëntie in verband met koeling en recentelijk komen daar ook milieu overwegingen bij. Doordat computers nu eenmaal warmte produceren, en zeker als je veel rekenkracht samen brengt op een klein oppervlak is er koeling nodig. Een hogere temperatuur zorgt ook voor een lagere betrouwbaarheid. Zonder koeling overleeft een krachtige processor het niet. Meestal zorgen ventilatoren voor de koeling van processoren of

video kaarten, waardoor computers vaak lawaai maken. Maar het kan ook met stille waterkoeling.

Figuur 1: Koeling van een computer

Tenslotte zal het duidelijk zijn dat de miljoenen PCs, servers en Internet routers die er op de wereld staan een flinke energie rekening en dus CO_2 uitstoot veroorzaken. Voor beheerders van serverparken is de energie rekening voor het server gebruik en de benodigde airconditioning aanzienlijk. De groei in elektriciteitsverbruik in Nederland wordt onder meer toegeschreven aan de groei in het computergebruik.
Zie hier het probleem: computers gebruiken steeds meer energie. De vraag is dus: kunnen we hier iets aan doen? Helaas is er niet een eenduidig antwoord. Het energieverbruik wordt namelijk niet alleen veroorzaakt door de processor, maar ook door het scherm, disk, IO etc. Als voorbeeld zien we in de volgende figuur de verdeling van het energieverbruik van een multimediaspeler.



Figuur 2: Energie van een audio speler

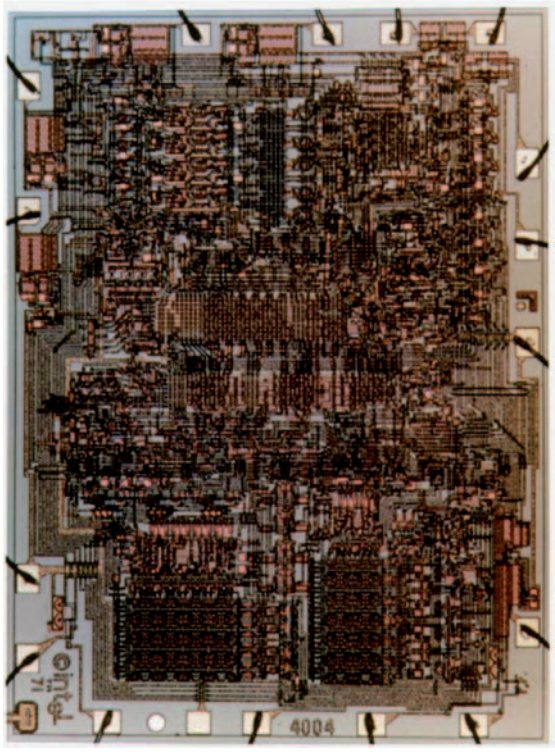
Historie

Om een inzicht te krijgen in het energieverbruik van computers ga ik met u een paar jaar terug in de geschiedenis. In 1946 kwam de ENIAC (Electronic Numerical Integrator And Computer) gereed, een van de eerste elektronische computers. Volgens huidige begrippen een kolossaal apparaat, het apparaat woog 30 ton. De ENIAC rekende met ongeveer 18.000 buizen, en kon 5000 optellingen en 300 vermenigvuldigingen per seconde uitvoeren. Ter vergelijk de huidige processoren halen gemakkelijk meer dan 1 miljard operaties per seconde. Een aardig detail is dat in 1955 de ENIAC onherstelbaar beschadigd werd door een blikseminslag. Te veel energie is niet gezond!
Het energie verbruik van de ENIAC wordt geschat op 140kWuur. Energie verbruik was toen nog helemaal geen ontwerp criterium.

Figuur 3: De ENIAC uit 1946

Een belangrijke doorbraak kwam in 1947 toen door Bardeen, Brattain en Shockley bij Bell Labs in Murray Hill in de VS de transistor uitgevonden werd. Door de komst van de transistor ging het energieverbruik van computers

drastisch omlaag; maar wat nog veel belangrijker was ze konden in grote aantallen goedkoop geproduceerd worden. In 1961 brachten Texas Instruments en Fairchild het eerste geïntegreerde circuit (IC) op de markt (twee flip-flops voor de opslag van 2 bits).
In 1968 opende een nieuw bedrijf zijn poorten Intel. In eerste instantie maakte Intel geheugen IC's; maar op aanvraag van een Japans bedrijf voor zakrekenmachines Busicom maakte Intel een 4-bits processor. Het Japanse bedrijf ging failliet en Intel nam uit het faillissement de rechten voor $60.000 over van de processor onder de naam van 4004. Als ooit nog eens een buitenaards wezen de ruimtesonde Pioneer 10 inspecteert dan zullen ze daar een Intel 4004 processor aantreffen. Ik vraag me af wat voor indruk dat zal maken.



Figuur 4: Intel 4004

In 1972 (het jaar waarin ik met mijn studie elektrotechniek begon) kwam Intel met de 8008 op de markt. Door tegenvallende prestaties werd deze snel opgevolgd door de 8080, een 8-bits processor met een 2 MHz klok. Dat was de microprocessor waar ik voor het eerst als student-assistent in aanraking ben gekomen, in het lab van de vakgroep digitale techniek op de 9e verdieping van het E&F gebouw, in de groep van Prof. Gerrit Blaauw. We werkten met een 8080 met 12 kB aan geheugen!
Een paar jaar daarvoor in 1965 formuleerde Gordon Moore [1], een van de grondleggers van Intel, zijn inmiddels zeer bekende wet van Moore. Gordon Moore voorspelde dat elke 18 maanden het aantal transistoren op een chip zou verdubbelen. Deze wet heeft de halfgeleider industrie sinds die tijd gedomineerd. Zo kunnen we vandaag computers ontwerpen met meer dan 500 miljoen transistoren op een chip.
De wet van Moore leidde tot het jaar 2000 tot een performance winst van 50% per jaar. Daarna vlakt de curve af. Door de komst van bipolaire transistoren en later MOS transistoren dachten we van het energieprobleem af te zijn. Maar niets is minder waar, de meest recente processoren van Intel hebben een energieverbruik van 130W. En dat op een oppervlak van postzegel: 2 bij 2 cm. u begrijpt dat daar flink wat koeling voor nodig is. Tot voor enkele jaren werden processoren ontworpen om zo snel mogelijk berekeningen uit te kunnen voeren; energie was bij het ontwerp van deze processoren van secundair belang. Jarenlang hielden Intel en AMD een wedloop om de snelste processor; volgens Intel de processor met de hoogste klokfrequentie. Pas toen draagbare computers op de markt kwamen realiseerde men zich weer dat computers energie verbruiken. De eerste laptops hielden er namelijk al na 1 uur mee op, omdat de batterij leeg was. Dit probleem werd groter naarmate de systemen kleiner werden: omdat dan de batterijen ook kleiner moesten worden. Dit gaf de aanzet tot de eerste generatie energiezuinige processoren. Ook kwamen er toen nieuwe performance indicatoren; in plaats van Mega Operations per Second (MOPS) werd steeds belangrijker het aantal MOPS per Watt (de computationele efficiency). Door de lage kostprijs van computers ontstonden er nieuwe mogelijkheden.
Rond 1990 ontstond Internet zoals we dat nu kennen. Via Internet werden eerst enkele, en later miljoenen computers met elkaar verbonden. Door Internet ontstonden geheel nieuwe fenomenen die voor die tijd nog niet bestonden bijvoorbeeld de zoekmachine Google, Internet winkels, marktplaats.nl, Internet encyclopedie Wikipedia, etc. Energie lijkt geen probleem te zijn; maar dat is maar schijn.
Als voorbeeld noem ik het bedrijf Google. Het is niet goed te achterhalen wat

de energie-rekening van Google is: maar dat het veel is is zeker. Geschat wordt dat er bij Google ongeveer 0.5 miljoen servers staan. Als we aannemen dat een gemiddelde server 400W energie dissipeert dan komen we op een totaal energie verbruik van 200MW gelijk aan de energie productie van een kleine kerncentrale. Daar bovenop komt nog het energieverbruik van het Internet verkeer. Aardig detail is: Google koos onlangs als vestigingsplaats voor zijn nieuwe server park een staat in de VS niet vanwege de goede infrastructuur of goedkope arbeidskrachten, maar omdat ze daar een lage energie prijs konden bedingen. In die staat stonden tot voor kort energieverslindende aluminium fabrieken, die zijn verplaatst naar Azië. Deze worden nu vervangen door energieverslindende servers. Voor veel server bedrijven is de energienota dan ook hoger dan de kosten voor arbeidsloon [14].

De volgende ontwikkeling dient zich al aan. Computers zijn inmiddels zo klein geworden dat ze in allerlei voorwerpen ingebed kunnen worden. Sinds een aantal jaren wordt hier veel onderzoek aan gedaan onder exotische namen als: ambient intelligence, pervasive computing, smart dust en ubiquitous computing. Het zijn zeer kleine computers die draadloos met elkaar kunnen communiceren.

Figuur 5: Smart dust

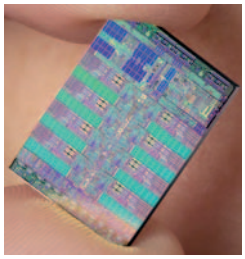
In de recentelijk opgerichte leerstoel Pervasive Systems, die voor een deel ontstaan is uit de leerstoel CAES, wordt onderzoek gedaan aan dit soort systemen. Voor pervasive systemen is energie van het allergrootste belang: omdat deze systemen liefst jaren moeten draaien op een of meer kleine batterijen. Nog beter zou het zijn als deze systemen hun energie uit de omgeving kunnen halen; ook wel energie harvesting of energie scavenging

genoemd. Het bijzondere is dat dit soort systemen, en daarom noem ik ze hier, is dat deze van meet af aan ontworpen zijn voor een extreem laag energieverbruik. Een nieuw fenomeen doet met dit soort systemen zijn intrede: 'stand-by' stroom. Dit is van cruciaal belangrijk; omdat dit soort apparaten 24 uur per dag 7 dagen per week moeten werken. U zult bij dit soort systemen tevergeefs zoeken naar een aan/uit knop. Het systeem werkt zolang er energie voorradig is. Om zo'n lange levensduur te halen moeten de systemen uiterst efficiënt werken. 99% van hun tijd slapen de processoren in de stand-by mode. Als ze slapen gebruiken ze enkele μA stroom, maar ze blijven wel alert. U zou kunnen zeggen dat deze processoren erg lui zijn: want ze verslapen het grootste deel van de tijd.

Waar staan we nu wat betreft het energieverbruik van processoren?

Fabrikanten van processoren lopen nu tegen harde energie grenzen aan. 130W vermogensdissipatie is wel ongeveer het maximaal haalbare. Bij een voedingsspanning van 1V lopen er zo stromen van 130A naar de processor. Er zijn echter nog meer grenzen waar ontwerpers van dit soort architecturen tegen aan lopen. Patterson [3][4] noemt naast de zojuist genoemde energiebarrière, de geheugenbarrière, en de ILP (Instructie Level Parallellisme) barrière. De geheugenbarrière heeft te maken met het verschil in snelheid tussen processor en geheugen. Om dit snelheidverschil te overbruggen worden moderne processoren met grote cachegeheugens uitgerust. Op dit moment wordt meer dan 90% van het chipoppervlak van een Pentium processor al door cachegeheugen ingenomen. Cachegeheugens hebben echter hun beperkingen. De ILP barrière heeft te maken met het probleem dat het steeds moeilijker wordt om voldoende parallellisme in sequentiële programma's te ontdekken om de executie units aan het werk te houden. De enige mogelijkheid die computerarchitecten hebben om meer rekenkracht te krijgen is om meer processoren op een chip parallel in te zetten. Dit soort processoren wordt meestal cores genoemd. Intel en met hun andere bedrijven brengen chips op de markt met meerdere cores: bijvoorbeeld begin dit jaar heeft Intel de quand-core Pentium op de markt gebracht (vier Pentiums op een chip). Andere ontwikkelingen zijn bijvoorbeeld de Intel research processor met 80 cores [8] of de Cell processor met 8 cores voor de Play Station 3 [7].

In het FP7 project CRISP, dat in januari 2008 van start gaat, werkt de leerstoel aan multi-core computerarchitecturen voor embedded systemen van de toekomst. We werken aan heterogene dynamisch herconfigureerbare System-on-Chip (SoC) architecturen met tientallen cores. Om dit op een energiezui-



Figuur 6: Cell processor

nige manier te doen moeten we echter wel bepaalde spelregels in acht nemen. Maar hierover later meer.
Met de komst van multi-core architecturen staan we nu voor een belangrijke splitsing en worden belangrijke trends doorbroken in computerarchitectuur land. Ik noem er hier vier.

1) Voorheen waren transistoren duur en was energie geen probleem: nu is energie belangrijk en zijn transistoren vrijwel gratis. Het loont nu om transistoren uit te zetten als je deze niet nodig hebt. Sterker nog als alle 500M transistoren tegelijk zouden gaan schakelen dan zou de chip verbranden door de warmte.
2) Voorheen was de interne werking van uni-processoren voorspelbaar en betrouwbaar; beneden de 65 nm technologie krijgen we te maken met soft-errors en statistisch gedrag van componenten.
3) Voorheen had het weinig zin om een applicatie te parallelliseren, want na 18 maanden was er toch wel een 2 keer zo snelle processor. Doordat de performance winst in multi-core architecturen zit, zijn we nu echter gedwongen om te parallelliseren om performance winst te halen.
4) Voorheen leverde wachten op een processor met een snellere klok automatisch performance winst. De grens aan de kloksnelheid van uni-processoren is inmiddels bereikt.

Het is duidelijk dat we de komende jaren naar tientallen, honderden cores op een chip gaan. De centrale vraag is hoe we die honderden processoren kunnen programmeren? Met het programmeren van parallelle architecturen hebben we de afgelopen 25 jaar slechte ervaringen opgedaan; maar we hebben nu geen keus meer. Daar liggen grote uitdagingen waar we ook in de leerstoel CAES aan werken.

Mogelijkheden

Welke mogelijkheden hebben we als computerarchitect om het energieverbruik omlaag te brengen? De meeste huidige processoren worden in CMOS technologie gemaakt. Het vermogen kan voor deze context benaderd worden door de som van het statische vermogen en het dynamische vermogen.
Het statische vermogen is het vermogen van een circuit ook al doet de schakeling niets. Het statische vermogen wordt o.a. veroorzaakt door stand-by stroomverbruik. Dit was tot voor kort verwaarloosbaar, maar is in de nieuwe technologieën zeker niet meer te verwaarlozen. De statische stroom is namelijk o.a. afhankelijk van het aantal transistoren op een chip: en die is flink gegroeid de laatste jaren.
Voor het dynamisch vermogen geldt voor elke circuit de formule:

$$P_{dyn} \sim \alpha f C V^2$$

waarbij f de frequentie van de schakeling is, C de totale capaciteit van de bedrading en input capaciteiten van afnemende poorten, V de voedingsspanning, en α de activiteit van de schakeling. Het totale dynamische vermogen is dan de som van het dynamisch vermogen van alle circuits.
U ziet dus als de activiteit nul is (α = nul) het dynamische vermogen nul is. Het vermogen is kwadratisch afhankelijk van de voedingsspanning. Door de voedingsspanning omlaag te brengen neemt het dynamisch vermogen af, maar helaas neemt de vertraging dan ook toe.
Wat kunnen we verder zoal doen om het energieverbruik terug te brengen? Ik zal enkele mogelijkheden met u doorlopen:

1) holistische benadering (de energie wet van Amdahl);
2) locality of reference;
3) adaptiviteit (luiheid principe);
4) efficiënte herconfigureerbare architecturen.

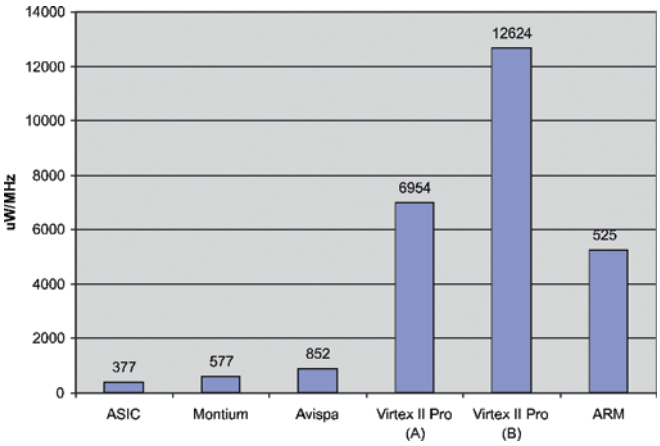
Ad 1) Holistische benadering: Met holistische benadering bedoel ik dat het energieverbruik in zijn volle context gezien moet worden. Het heeft weinig zin om speciale voorzieningen in een processor te stoppen voor het energieverbruik als de bovenliggende softwarelagen (bijvoorbeeld compilers en operating systemen) er geen gebruik van kunnen maken. Dit staat ook wel bekend als de energiewet van Amdahl. Bijvoorbeeld: als de processor van een laptop 50% van de energie gebruikt, en we zouden de energieconsumptie van de processor van een laptop op nul kunnen krijgen zonder de rest aan te passen, dan kan de laptop nog steeds maar twee keer zo lang mee.
De consequentie van de holistische benadering is dat wij in onze groep niet alleen met computerarchitectuur, maar ook met applicaties, chip design,

compilers en operating systems in aanraking komen. Het is onmogelijk om al deze expertises in een groep te hebben. Vandaar dat wij veel samenwerken met andere leerstoelen binnen de UT (bijvoorbeeld ICD, PS, DIES, SAS en DWMP binnen EWI) en onderzoeksgroepen en bedrijven buiten de UT (bijvoorbeeld. TU/e, TUD, NXP, Thales, Philips Medical Systems, Océ).
Ad 2) Locality of reference: Transporteren van data kost energie. Hoe groter de afstand die overbrugd moet worden hoe meer energie het kost. Communicatie binnen een chip over een korte afstand is het goedkoopst, off-chip communicatie is al duurder, maar een byte draadloos versturen is erg duur. Op dit moment is off-chip communicatie energietechnisch 1000 keer zo duur dan on-chip communicatie en draadloze communicatie weer 1000 keer duurder.
Zo is het draadloos versturen van een byte equivalent met het executeren van 2800 instructies op een energiezuinige processor [18]. Omdat transport duur is moeten we dit zoveel mogelijk vermijden. Dit wordt aangeduid met locality of reference. Locality of reference is een uitermate belangrijk middel om energiezuinig te zijn. Dit heeft als consequentie dat veel gebruikte data zo dicht mogelijk bij de processor, zo mogelijk in een on-chip geheugen, geplaatst moet worden. Voor sensorsystemen betekent dit dat processing zo dicht mogelijk bij sensoren plaats moet vinden. Locality of reference is niet alleen belangrijk voor het energieverbruik maar ook voor de performance van het systeem. Voordat we data van elders ophalen moeten we ons afvragen of we die data niet kunnen (her)berekenen. Want processing wordt relatief gezien steeds goedkoper en off-chip communicatie steeds duurder. In de in de groep ontwikkelde architecturen zoals de Montium is het locality of reference principe nadrukkelijk gebruikt [9]. De Montium Tile processor bevat naast de processor ook geheugen voor data en instructies. Als het hoofdbestanddeel van de communicaties zich binnen de core afspeelt kan een energie-zuinig systeem gebouwd worden. Bijvoorbeeld: voor een 1024pFFT spelen meer dan 90% van de geheugenreferenties voor een Montium zich binnen de core grens af.

Algorithm	#Interne geheugen refs			#Externe geheugen refs		
	Read	Write	Total	Read	Write	Total
1024p FFT	30720	20480	51200	2048	2048	4096
200 tap FIR	400	5	405	1	1	2
SISO alg. (N softbits)	10*N	8*N	18*N	2*N	N	3*N

Tabel 1: Interne en externe geheugen referenties

Ad 3) Adaptiviteit: Het adaptiviteits (of luiheids) principe houdt in dat de processor niet meer werk doet dan strikt noodzakelijk: de processor past zich aan aan de vraag van de applicatie. Bijvoorbeeld: als de processor klaar is met rekenen gaat deze zo snel mogelijk naar een low-power mode (bijvoorbeeld stand-by mode). Veel systemen worden momenteel ontworpen voor het worstcase scenario. Helaas is worstcase ook vaak energietechnisch het duurst. Dus is het beter een systeem voor worstcase te ontwerpen en te optimaliseren voor de typical case.
Het snel omschakelen naar een low-power (slaap) mode is voor veel processoren problematisch. Zeker de wat oudere processoren hebben die mogelijkheid niet: ze lopen altijd op volle snelheid. Een soort raceauto die altijd vol gas rijdt. Moderne processoren hebben Dynamic Voltage en Frequency Scaling mogelijkheden: bijvoorbeeld de speed-step technologie van Intel. Het kost echter wel extra energie en tijd om processoren om te schakelen (bijvoorbeeld omdat de PLL eerst stabiel moet werken en omdat de status van berekeningen opgeslagen moet worden). Een complicerende factor is dat applicaties niet altijd even goed voorspelbaar zijn; waardoor niet altijd goed bepaald kan worden wanneer de processor in de stand-by stand gebracht kan worden of wakker gemaakt moet worden.
Zoals zojuist al aangegeven is, is voorspelbaarheid van groot belang [12]. Als applicaties en architectuurconcepten een voorspelbaar gedrag vertonen, dan kan daar heel veel energiewinst mee behaald worden. In de groep wordt gewerkt aan architecturen waar technieken die leiden tot onvoorspelbaar gedrag, bijvoorbeeld speculatieve executie van instructies, cache geheugens en bussen, die veel toegepast worden in high-end PC's, zoveel mogelijk worden vermeden. In onze aanpak gebruiken we cores zonder speculatieve instructies, met een Scatchpad geheugen in plaats van cachegeheugen en een Network-on-Chip in plaats van een bus zodat garanties gegeven kunnen worden voor minimale doorvoer en maximale vertraging.
Ad 4) Tot slot, in de groep wordt gewerkt aan efficiënte architecturen. Hiervoor wordt o.a. gebruik gemaakt van herconfigureerbare architecturen. Dat zijn architecturen waarbij de instructieset aangepast wordt aan de behoefte van de applicatie. In de afgelopen jaren hebben we aangetoond dat deze aanpak nuttig is voor het energieverbruik.



Tabel 2: Energie verbruik van een FFT butterfly

Deze architecturen hebben een lage control overhead en komen qua efficiëntie in de buurt van ASICs. In bovenstaande figuur wordt het energieverbruik van een Montium vergeleken met andere processoren, o.a. een FPGA en een ARM processor. Ik ben ervan overtuigd dat de toekomst is aan heterogene multi-core herconfigureerbare architecturen, waarbij de communicatie via een Netwerk-on-Chip (NoC) gaat. Dit soort systemen zal op den duur de generieke architectuur worden niet alleen voor embedded systemen maar ook voor processoren in PC's. Het zal ASICs, FPGAs en energiehongerige processoren vervangen. De CRISP architectuur is een voorbeeld van zo'n architectuur. Het bestaat uit tientallen herconfigureerbare cores en twee ARM processoren. Waarom is zo'n oplossing gunstig voor het energieverbruik voor toekomstige systemen?
Ik zal een paar punten noemen:

1) Voor elke core afzonderlijk kan de spanning en frequentie naar behoefte aangepast worden
2) Cores die niet gebruikt worden kunnen uitgezet worden wat statische energie bespaart
3) De architectuur is opgebouwd uit een beperkt aantal relatief eenvoudige bouwblokken, die goed geverifieerd en getest kunnen worden
4) De cores bevatten naast een processor ook geheugen zodat locality of reference inherent ondersteund wordt.
5) Het systeem heeft een voorspelbaar gedrag en compositionaliteit

wordt goed ondersteund.

6) Het systeem kan de betrouwbaarheid en yield verbeteren, omdat defecte cores uitgezet kunnen worden en andere hun werk kunnen overnemen.

Voorbeeld 1

Hierna zal ik wat dieper ingaan op een tweetal onderwerpen, waaraan de groep de komende jaren zal gaan werken. Het eerste voorbeeld gaat over zogenaamde streaming applicaties and multi-core architecturen.
Streaming applicaties zijn toepassingen die gekenmerkt worden door een regelmatig communicatie en rekenpatroon. Voorbeelden daarvan zijn multimedia systemen zoals TV's, camera's, DVD-spelers, MP3-spelers, maar ook high-performance systemen zoals medische beeldbewerking, phased array antenne systemen zoals gebruikt in radio astronomie en radar systemen en beeldbewerkingsprocessen voor kopieerapparaten.
Dit soort systemen kunnen gemodelleerd worden als een data-flow graaf, bijvoorbeeld een Kahn process graaf, en hebben de volgende kenmerken:

- De communicatie tussen de processen heeft een grote doorvoer en de processing voor ieder data element is lokaal.
- De data elementen stromen van proces naar proces en veroorzaken een repeterend gedrag. De tijd tussen twee data elementen is applicatie afhankelijk bijvoorbeeld 4 μs voor HiperLAN/2 of 20 ms voor een video frame.
- Ook de grootte van de data elementen is applicatie afhankelijk bijvoorbeeld 64 32-bits woorden voor HiperLAN/2 OFDM symbool of 8x8 24 bits waardes voor een video macro blok.
- Het repeterende gedrag veroorzaakt een zekere voorspelbaarheid waar de systeemarchitect zijn voordeel mee kan doen.
- Voor de meeste applicaties geldt dat er strikte timing garanties opgelegd zijn bijvoorbeeld het aantal elementen per seconde dat minimaal verwerkt moet worden.
- Tenslotte, als een streaming applicatie eenmaal opgestart is, dan zal deze gedurende een langere periode blijven lopen. Bijvoorbeeld: als een MP3 applicatie wordt opgestart dan zal deze minstens enkele minuten blijven draaien.

Recentelijk heeft de leerstoel CAES het initiatief genomen om samen met de TU/e, TUD, LIACS, UvA, NXP, Thales, Océ en Philips Medical Systems onder-

zoeksvoorstellen in te dienen rond het thema adaptive streaming applicaties. In januari 2008 gaan we samen met Recore, NXP, Thales, Atmel en Universiteit van Tampere in Finland van start met het EU FP7 CRISP project. In deze projecten komen de eerder genoemde energie principes terug. Ik zal hier iets verder ingaan op het gebruik van adaptiviteit. In de meeste hedendaagse systemen wordt er een voorspelling gemaakt voor de belasting van een core voor de komende periode, zodat de spanning en frequentie van de core ingesteld kan worden. Deze systemen zijn dus gebaseerd op het gedrag in het verleden. Maar zoals ook voor de belegger geldt: berekeningen gebaseerd op het gedrag uit het verleden zijn geen garantie voor de toekomst. Voor toepassingen waarbij het gedrag soms sterk wijzigt in de tijd, bijvoorbeeld een scène wisseling in video, werkt deze aanpak niet goed.
In streaming applicaties hebben processoren niet altijd werk te doen. Dus in ons voorstel wordt aan elke core-processor een klein beetje logica (een soort nano-processor) toegevoegd die een soort wekker functie heeft. Als het proces dat op de core-processor moet draaien alle input beschikbaar heeft en er is voldoende ruimte voor het resultaat, dan wordt de core-processor wakker gemaakt.
Deze rekent dan een iteratie van de applicatie uit en gaat vervolgens weer slapen totdat de nano-processor hem weer wakker maakt. Dit soort technieken zijn alleen mogelijk als:

- We voldoende kennis hebben over het gedrag van de streaming applicatie;
- We over technieken beschikken om cores snel aan en uit te kunnen zetten: liefst de voedingspanning uitzetten en de klok stil zetten, maar de inhoud van geheugens wel bewaren. Voor deze aanpak wordt gebruik gemaakt van zogenaamde ‘slaaptransistoren’. Dat zijn transistoren die een deel van de chip kunnen uitzetten. U zou het kunnen beschouwen als dynamische clock-gating op kleine schaal.
- De core processor moet ook zeer snel (binnen enkele clock cycli) van standby naar actief kunnen omschakelen.

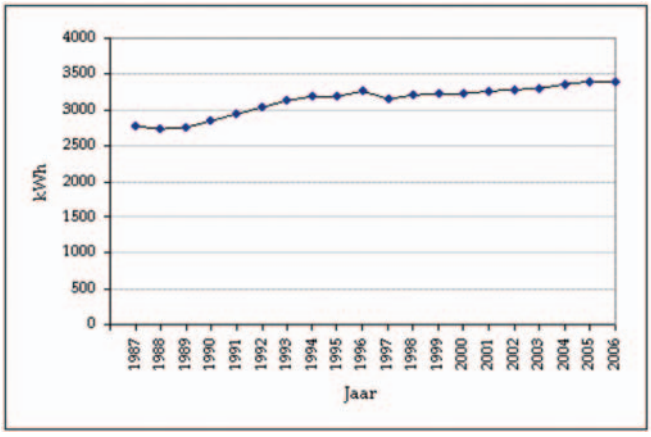
We zien dus dat we logica (transistoren) inzetten om energie te besparen. Ook hier zien we weer dat we kennis van veel disciplines moeten combineren. Voor adaptieve streaming applicaties werkt dit zeer efficiënt. Voor streaming applicaties weten we veelal wel het verwachte worstcase gedrag van de componenten. Dus kunnen we het systeem zo op design-time dimensioneren voor het verwachte worstcase geval en via adaptiviteit kunnen we op run-time het energieverbruik omlaag brengen.
Kenmerkend aan onze aanpak is dat we het afbeelden van streaming appli-

caties naar een multi-core architectuur dynamisch, op run-time, doen, terwijl verreweg de meeste ontwerpers nog steeds deze afbeelding op design-time doen[13]. Dit houdt in dat we op design-time de kernels naar een of meer cores compileren en annoteren met performance getallen (bijvoorbeeld energie verbruik en executie tijd). Deze informatie kan dan door het run-time systeem gebruikt worden om de optimale mapping te vinden. Dit kan alleen efficiënt als alle systeemcomponenten deterministische eigenschappen hebben. Het uitstellen van de afbeelding tot run-time heeft een aantal interessante voordelen bijvoorbeeld.

1) Op run-time weten we de exacte QoS eisen van het systeem;
2) We weten dan welke combinatie van applicaties draaien;
3) We kunnen de dynamiek van applicaties maximaal benutten;
4) We kunnen rekening houden met defecte componenten in ons systeem, wat gunstige is voor de yield en de betrouwbaarheid.

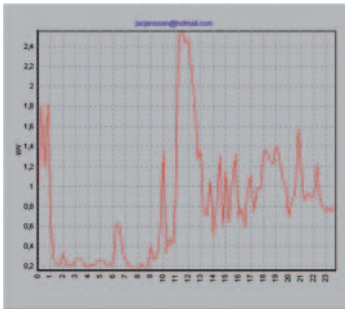
Voorbeeld 2

In het begin van mijn oratie heb ik gesteld dat computers mede de oorzaak zijn van een stijgend elektriciteitsverbruik. Met ICT technologie kunnen we hier echter ook wat aan doen. Dit lijkt een contradictie, maar ik zal proberen uit te leggen dat dit niet zo is.
Op de volgende figuur zien we het gemiddelde elektriciteitsverbruik van een Nederlands gezin op jaarbasis.



Figuur 7: Elektriciteitsverbruik van een Nederlands gezin op jaarbasis

Als we het elektriciteitsverbruik van een dag beschouwen dan blijkt dat zelfs 's nachts het verbruik ongeveer 200 Watt is. Het blijkt dat 10 tot 15 % van alle elektrisch vermogen wordt opgenomen door de zogenaamde stille energievreters: stand-by van TV, radio, netadapters voor kabel of ADSL modems, opladers voor telefoons etc.
Verwacht wordt dat in de toekomst steeds meer energie decentraal wordt opgewekt, bijvoorbeeld via zonnecellen, brandstof cellen en micro-WKKs. Als voorbeeld noem ik de micro-WKKs (micro Warmte Kracht Koppeling). Dat zijn apparaten die de CV ketels in gewone huizen op den duur vervangen door een apparaat waar gas in gaat en naast warmte (voor de verwarming en warm water) ook elektriciteit levert. De verwachting is dat er over 5 tot 10 jaar een miljoen van dit soort systemen in Nederland geïnstalleerd zijn [15][16].
De opgewekte elektriciteit kunt u in huis gebruiken, maar wat u niet nodig hebt kunt u exporteren naar het elektriciteitsnet. Een probleem is dat een micro-WKK slechts 1 kW aan elektrisch vermogen levert. Dit zou bijna voldoende zijn om een Nederlands huis van energie te voorzien, ware het niet dat het energieverbruik van een huis niet constant is.



Figuur 8: Elektriciteitsverbruik van een dag (4 mei 2002)

Pieken worden veroorzaakt doordat meerdere apparaten tegelijk aanstaan; bijvoorbeeld de wasmachine en de vaatwasmachine. In het recentelijk gehonoreerde STW project SFEER gaan we slimme schedulingstechnieken inzetten om er voor zorgen dat dit niet optreedt zodat deze pieken zoveel mogelijk worden vermeden. Dit staat bekend als 'peakshaving'. In dit onderzoek wordt samengewerkt met Essent, Gasterra, E.ON UK, Homa Software, Eaton and Salland Electronics [17]

Ook hier liggen flinke uitdagingen. Onlangs hebben we een SmartSiP voorstel ingediend waarin wordt voorgesteld om elk wandcontactdoos in het huis uit te rusten met een kleine processor. Gezamenlijk kunnen alle wandcontactdozen ervoor zorgen dat zoveel mogelijk piekbelastingen voorkomen worden en dat de micro-WKK aanspringt wanneer er voldoende vraag is naar elektriciteit. Als bijkomstigheid kunnen we nu ook apparaten uitschakelen die onnodig in de stand-by stand staan. Als dit systeem op een grote schaal wordt toegepast kan een aanzienlijke energie besparing en dus CO_2 reductie bereikt worden.
Dit soort slimme wandcontactdozen moeten een uitermate lage stand-by stroom hebben, anders spannen we het paard achter de wagen. Ook voor dit systeem geldt weer: kijk naar het totale systeem, gebruik adaptiviteit, en snel aan en uit schakelen.

Uidagingen

In projecten zoals hierboven genoemd willen we de CAES groep in Nederland, binnen en buiten Europa op de kaart zetten. Waar liggen de grote uitdagingen waar we aan gaan werken:

1) ontwikkelen van efficiënte systemen,
2) het specificeren van streaming applicaties,
3) het afbeelden van streaming applicaties op multi-core architecturen,
4) het detecteren en corrigeren/anticiperen op statistische fouten van toekomstige chips.

Efficiënte computer architecturen zijn nodig voor efficiënte streaming applicaties zoals eerder genoemd. Efficiënte architecturen zijn ook nodig voor de kleinste computers zoals bijvoorbeeld gebruikt voor draadloze sensor nodes, de eerder genoemde slimme wandcontactdozen, en nog wat verder in autonome objecten of elektronisch klei [19]. Dat zijn kleine componenten die door het activeren van magneten elkaar kunnen aantrekken en dus kunnen bewegen. Voor deze toepassingen is energie een primair ontwerpcriterium. Voor zwaardere systemen met een budget van miljarden transistoren gelden wat andere regels: hier is een flinke overhead in transistoren acceptabel om chips te realiseren met 100% yield met een voldoende performance en flexibiliteit. Dit zal ook hard nodig zijn om het hoofd te bieden aan de toekomstige statistische fout modellen. De kunst zal zijn om met onbetrouwbare componenten toch betrouwbare en voorspelbare systemen te maken. Voor desk-top processoren kunnen we, voor wat betreft de energie consump-

tie, veel leren van processoren voor sensor nodes. Ik denk bijvoorbeeld aan het snel aan/uit schakelen van niet gebruikte onderdelen en adaptief regelen van spanningen en frequenties.
Specificeren van streaming applicaties zal op een hoger niveau moeten gebeuren zonder de efficiëntie uit het oog te verliezen. In de toekomst zullen compilers applicaties niet meer vertalen naar instructies, maar compileren naar voorgedefinieerde kernels die op een of meerdere cores efficiënt uitgevoerd kunnen worden. Zo zullen we streaming applicaties kunnen beschrijven als een data-flow graaf. Hierbij kunnen we aansluiten bij ontwikkelingen zoals StreamIt en S-NET [5][6]. Het afbeelden van applicaties op multi-core architecturen zal ook onze onderzoeksagenda bepalen. Als we in staat zijn de afbeelding efficiënt op run-time uit te voeren dan zijn we in staat systemen te realiseren die adaptief zijn; betrouwbaar en toch energiezuinig werken.

Onderwijs

De leerstoel is betrokken bij bachelor en master onderwijs binnen Technische Informatica en Elektrotechniek. De leerstoel is nauw betrokken bij de landelijke 3TU master Embedded Systems. Wij bieden vakken aan die op het grensvlak van informatica, elektrotechniek en wiskunde liggen. Wat dat betreft zijn we een echte EWI leerstoel. Wij hebben de afgelopen jaren vele afstudeerders afgeleverd. Het afnemende veld, de industrie zowel als kennisinstellingen, staat te springen om dit soort mensen. Als ik de kwaliteit van onze afstudeerders vergelijk met die van toonaangevende universiteiten in ons omringende landen dan kunnen zij zich zeker met hen meten. Helaas is het aantal eerstejaars studenten laag. Wat doen wij fout? Als iemand suggesties heeft laat het ons weten. Is ons imago wel goed? Weet een VWO student wat informatica is? Of denkt deze nog steeds dat informatica een soort combinatie van Word/ Power-Point en Excel en spelletjes is. Ik denk dat daar nog veel zendingswerk te verrichten is.
Ik hoop dat ik u duidelijk heb gemaakt dat informatica en elektrotechniek (en zeker embedded systemen) veel meer is dan Word, Power-Point en Excel. Toegegeven ontwerpen van embedded systemen is geen gemakkelijk vak; maar wel een leuk en uitdagend vak. Ik denk dat de sterkte van onze faculteit de goede samenwerking tussen wiskunde, elektrotechniek en informatica is. We moeten onze sterkte nog veel beter verkopen. Ik hoop daar mijn steentje aan bij te dragen. Mijn mening is dat we een technische faculteit

zijn en dat ook moeten blijven. De voorgestelde nieuwe studierichting Creative Technologie kan een bijdrage leveren aan de holistische visie die ik eerder geschetst heb. Maar ik merk uit dagelijkse contacten met bedrijven dat de arbeidsmarkt grote behoefte heeft aan afgestudeerden met een brede technische opleiding zoals we die nu afleveren.
Dat betekent dat we moeten blijven investeren in kennis van studenten: studenten "hands-on experience" geven op een breed terrein van moderne multi-core computer architecturen via systeem software, software engineering, DSP algoritmes, chip design, compiler back-ends, real-time operating systems, gedistribueerde systemen tot parallelle applicaties. En dat alles in een embedded systemen dus resource zuinige context.

Dankwoord

Tot slot: zonder de hartverwarmende steun van de mensen uit de leerstoel CAES, de vaste staf (Paul, Bert, Andre, Jan en Hans) de secretaresses Marlous en Nicole, en de AIO's en ex-AIO's, had ik hier vandaag niet gestaan. Het is dan ook een voorrecht om leiding te geven aan deze fantastische groep.
Ik ben er trots op dat ons onderzoek niet alleen tot AIO plekken maar ook tot bedrijvigheid in onze regio leidt (met name Recore Systems en HOMA Software) en voor wat betreft sensornetwerken Ambient Systems. De resultaten van ons onderzoek verdwijnen niet in de boekenkast, maar gaan naar spin-off bedrijven. Ik hoop en verwacht dat nog meer van onze AIO's dit goede voorbeeld zullen volgen.
Bij deze woorden van dank wil ik ook mijn voorganger professor Krol betrekken, wiens taak ik heb overgenomen. Thijs bedankt zonder jouw kennis en steun was de groep er in deze vorm niet geweest. Je hebt een goed lopende trein aan mij overgedragen en ik zal proberen deze trein gaande te houden.
Ik bedank de collega's bij informatica, elektrotechniek en wiskunde in Twente en de andere universiteiten en de bedrijven voor de samenwerking.
Ook wil ik STW en NWO bedanken voor de prettige manier waarmee wij door hun begeleid worden. Ik hoop dat we nog veel projecten samen kunnen uitvoeren.
Tenslotte een paar privé woorden. Zonder de stimulering van mijn vader, moeder, broers en zussen was ik niet zover gekomen. Het is jammer dat mijn ouders deze oratie niet meer kunnen meemaken. Precies vandaag zou mijn moeder 96 jaar zijn geworden. Het was met name mijn vader die mij stimuleerde om na de MULO naar de HBS te gaan, wat ik eigenlijk helemaal

niet zag zitten, omdat ik na 5 jaar MULO de middelbare school wel gezien had.
Eefje, Maarten en Jasper ik hoop dat jullie niet te veel geleden hebben van mijn energie activiteiten. Vaak 's avonds en in het weekend papers lezen en projectvoorstellen schrijven is niet altijd gezellig.
En lest best Alda. Je bent je er niet altijd van bewust, maar je staat voor de waarden die er uiteindelijk alleen maar toe doen. In veel belangrijke beslissingen in mijn leven was jij betrokken. Zo ben jij er eigenlijk ook mede de oorzaak van dat ik 10 jaar geleden in de energie gedoken ben.
Ik hoop dat je me nog vaak de goede kant op stuurt.

Samenvattend: Energie in computerarchitectuur is een hot issue of misschien beter een cool issue.

Ik heb gezegd.

Referenties

[1] Gordon E. Moore, “Cramming more components onto integrated circuits”, Electronics, Volume 38, Number 8, April 19, 1965

[2] William Dally et al. "Stream Processors: Programmability with Efficiency" ACM Queue, March 2004, pp. 52-62.

[3] D. Patterson, Arvind, K. Asanovic, D. Chiou, J. Hoe, C. Kozyrakis, S. Lu, M. Oskin, J. Rabaey, J. Wawrzynek, "RAMP: Research Accelerator for Multiple Processors," Technical Record of the 18th Hot Chips Symposium, Palo Alto, CA, August 2006

[4] Krste Asanovic, etal “The Landscape of Parallel Computing Research: A View from Berkeley” , EECS Department University of California, BerkeleyTechnical Report No. UCB/EECS-2006-183, December 18, 2006

[5] William Thies, Michal Karczmarek, Saman Amarasinghe, “StreamIt: A Language for Streaming Applications”, proceedings 11th International Conference ETAPS 2002, April 8-12, 2002

[6] Clemens Grelck, Sven-Bodo Scholz, Alex Shafarenko, “S-Net: A Typed Stream Processing Language”, Proceedings of the 18th International Symposium on Implementation and Application of Functional Laguages (IFL’06) September 4-6, 2006, pp. 81-97.

[7] James A. Kahle, et al. “Introduction to the cell multiprocessor”. IBM Journal of Research and Development, 49(4/5):589–604 2005.

[8] Sriram Vangal, et al, “An 80-tile 1.28 Tflops network-on-chip in 65nm CMOS”, In Proceedings of the International Solid State Circuits Conference, February 2007.

[9] P.M. Heysters, “Coarse-Grained Reconfigurable Processors – Flexibility meets Efficiency”, PhD. thesis, University of Twente, 2005, ISBN 90-365-2076-2.

[10] Smit, G.J.M. and Kokkeler, A.B.J. and Wolkotte, P.T. and Hölzenspies, P.K.F. and van de Burgwal, M.D. and Heysters, P.M. “The Chameleon Architecture for Streaming DSP Applications”. EURASIP Journal on Embedded Systems, 2007

[11] Rauwerda G.K., Heysters P.M. and Smit G.J.M., “Towards Software Defined Radios using Coarse-Grained Reconfigurable Hardware”, accepted for IEEE Transactions on VLSI, 2008

[12] Wolkotte, P.T. and Hölzenspies, P.K.F. and Smit, G.J.M. “Fast, Accurate and Detailed NoC Simulations”. In: Proceedings of the 1st ACM/IEEE International Symposium on Networks-on-Chip, 6-9 May 2007, Princeton, NJ, USA. pp. 323-332.

[13] Philip Hölzenspies, J. Kuper, J. Hurink, G.J.M. Smit, “Run-time Spatial Mapping of Streaming Applications to a Heterogeneous Multi-Processor System-on-Chip (MPSoC)”, accepted for DATE 2008.

[14] Joris Polman “Internet sloopt het klimaat” Spits, 13 september 2007.

[15] Our Energy Challenge – Power from the people; Microgeneration Strategy, Department of Trade and Industry, London, UK, March 2006

[16] Marktontwikkeling Micro – en mini- warmtekracht in Nederland tot 2020, Smart Power Foundation, april 2006.
[17] V.Bakker, A. Molderink, G.J.M. Smit, J. Hurink: "Algorithm Design for next generation energy solutions using micro-CHP", abstracts collection for Siren 2007
[18] van Hoesel, L.F.W. and Havinga, P.J.M. "Design Aspects of An Energy-Efficient, Lightweight Medium Access Control Protocol for Wireless Sensor Networks", Technical Report TR-CTIT-06-47 University of Twente, Enschede. ISSN 1381-3625, 2006
[19] http://www.cs.cmu.edu/~claytronics/

Universiteit Twente
de ondernemende universiteit