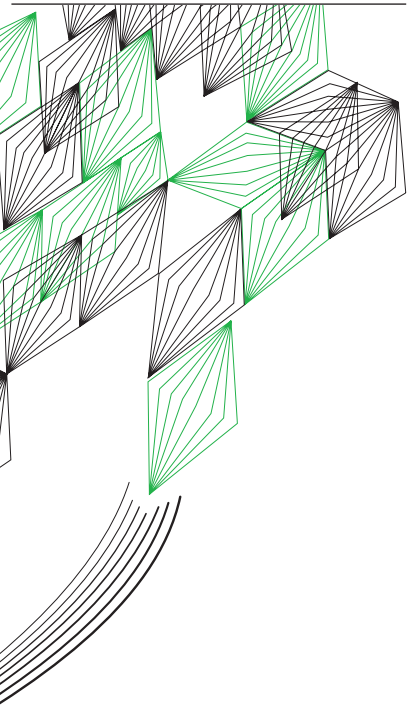
AFSCHEID PROF.DR.IR. GERARD J.M. SMIT
VRIJDAG 9 JUNI 2017



09 06 2017

UNIVERSITY OF TWENTE.

ENERGIE EN COMPUTER ARCHITECTUUR

- VAN PICOJOULE NAAR GIGAJOULE -

AFSCHEIDSREDE PROF. DR. IR. GERARD J.M. SMIT

9 JUNI 2017

COLOFON

June 2017

ENERGIE EN COMPUTER ARCHITECTUUR - VAN PICOJOULE NAAR GIGAJOULE -

Mijnheer de rector, collega's, vrienden en familie. Met deze afscheidsrede wil ik graag terug kijken op de afgelopen jaren bij de Universiteit Twente, en de laatste jaren als hoogleraar CAES (Computer Architecturen voor Embedded Systems). Er is in die jaren in dit werkveld ontzettend veel veranderd, te veel om in 45 minuten aan de orde te stellen. Ik zal me in dit college moeten beperken tot slechts twee onderwerpen. Het eerste deel zal gaan over de ontwikkeling van efficiënte computer architecturen, in het bijzonder herconfigureerbare architecturen. Dit verklaart de subtitel picoJoule (= 10^{-12} Joule) in de titel. Even voor de energie-leken onder ons, Joule is de eenheid van energie en dat is weer gelijk aan 1 Ws. B.v. een lamp van 100 W die 1 minuut brandt verbruikt 100 x 60 = 6000 Joule = 6kJoule. De dames in de zaal zijn wellicht beter bekend met kilo calorieën. Maar die kun je in elkaar uitdrukken: 1 kCal is gelijk aan 0,24 kJoule. In het tweede deel van mijn rede zal ik ingaan op het gebruik van embedded systemen voor het besturen van energie stromen op wijkniveau of stadniveau de zg Smart Grids, vandaar Giga Joule (= 10^{9} Joule) in de subtitel.

Herconfigureerbare architecturen.
Tegenwoordig weet iedereen wat een computer is. Dat was in 1972, toen ik begon als student elektrotechniek, wel anders. Er bestond toen nog geen PC, geen internet, geen GPS, geen Microsoft, geen Apple, en geen mobiele telefoon, laat staan Smartphones. Je schreef nog brieven, of gebruikte dit apparaat om op afstand met elkaar te praten. Jongeren hebben tegenwoordig geen idee wat dit is of wat je hiermee kunt.

Figure 1 Telefoon

Het vakgebied elektrotechniek en informatica heeft sinds 1972 een stormachtige ontwikkeling doorgemaakt. Figuur 2 is een veelzeggend plaatje over de ontwikkeling van elektrotechniek en informatica in 8 jaar tijd.
Mijn onderzoeksgebied is embedded systemen. Een embedded systeem is een computer die onlosmakelijk verbonden is met een apparaat. De volgende figuur geeft een overzicht van de verschillende embedded systemen waar ik in de loop der tijd aan heb gewerkt.

Figure 2: St Pietersplein met 8 jaar tijdsverschil

Zo'n embedded apparaat kan variëren van een eenvoudige halsband voor een koe, tot een complete printer. De uitdaging van een embedded systeem is dat we bij het ontwerpen te maken hebben met fysieke beperkingen bv. de grootte van een apparaat, of de beschikbare energie, of de toegestane rekentijd. Hoe kleiner een apparaat, hoe korter de rekentijd, hoe minder energie je ter beschikking hebt, hoe complexer het apparaat, des te interessanter wordt het probleem voor een embedded systemen ontwerper.

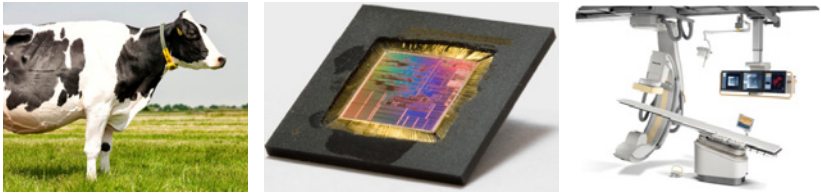


Figure 3: Voorbeelden van embedded systemen

Als onderzoeker heb ik gemerkt is het vaak handig om de grenzen op te zoeken. In het recent goedgekeurde STW perspectief onderzoeksvoorstel ZERO doen we bijvoorbeeld onderzoek aan energie-autonome embedded systemen, systemen die in hun eigen energie moeten voorzien: een combinatie van elektrotechniek, wiskunde en informatica.
Een normale computer, zoals die in een PC of laptop zit, is een alleskunner, in het Engels een 'general purpose' computer. Je zou kunnen zeggen one-size-fits-all. Ze zijn heel flexibel, je kunt er allerlei mogelijke toepassingen op draaien: van reken problemen, bv. de belastingaangifte, tekstverwerking zoals MSWord, tot video telefoneren zoals Skype. Het nadeel van een alleskunner is, dat ze wel alles kunnen maar niet erg efficiënt zijn. Voor een normale PC die zijn energie uit het lichtnet haalt is dit niet direct van belang, maar er zijn binnen het domein van embedded systemen ook toepassingen waar de standaard PC lang niet genoeg rekenkracht voor heeft of niet efficiënt genoeg is. Voorbeelden zijn te vinden in radioastronomie, medische beeldbewerking en toekomstige autonoom rijdende auto's.
Waarom is een alleskunner als een PC zo inefficiënt? Daar is niet één duidelijke oorzaak voor aan te geven, maar dit is een combinatie van factoren. Ik zal slechts op één belangrijk punt ingaan: de kosten van communicatie.
Een basisbewerking van een computer noemen we een instructie. Een bijvoorbeeld van zo'n instructie is een vermenigvuldig instructie die twee getallen met elkaar kan vermenigvuldigen. Alle instructies die een computer uitvoert kosten een klein beetje tijd en energie.

Add			Add			Cache	(64 bit)
8 bit	0.03 pJ		16 bit	0.4 pJ		8kB	10 pj
16 bit	0.1 pJ		32 bit	0.9 pJ		32 kB	20 pJ
Mult			Mult			1 MB	100pJ
8 bit	0.2 pJ		16 bit	1.1 pJ		DRAM	1.3-2.6nJ
16 bit	3.1 pJ		32 bit	3.7 pJ			

Instruction energy breakdown

I cache	RF access	control	operation	
25 pJ	6 pJ	43 pJ	1 pJ	Total 70 pJ

Figure 4: Energie kosten van een instructie in 45 nm

Dit plaatje laat het energie-verbruik van een instructie zien: in totaal 70pJ. Het laat twee opvallende dingen zien: 1) de meeste energie zit niet in rekenen (slechts 1 pJ), en 2) veel energie zit in overhead o.a. in control en communicatie met het geheugen. Communicatie is over het algemeen heel duur qua performance, dwz. kosten in tijd en kosten in energie verbruik. We moeten dus op zoek naar oplossingen om het geheugengebruik te minimaliseren in plaats van het aantal berekeningen minimaliseren. U zou natuurlijk kunnen zeggen, dat zijn picoJoules dus dat stelt niet veel voor. Maar bedenk wel

dat hedendaagse computers dit soort instructies in een verbazingwekkend tempo van plm. 1 miljard instructies per seconde kunnen uitvoeren.
1 Miljard maal iets kleins kan toch tot veel energieverbruik leiden. De volgende tabel laat zien dat vooral communicatie over afstand duur is. Draadloze dataoverdracht is helemaal duur qua energie verbruik.

Operation (8-bit operand)	Energy/Op (45 nm)	Cost (vs. ALU)
ALU operation	0.05 pJ	1X
Move 10 mm on-chip	2.4 pJ	50X
Load from on-chip SRAM	2.5 pJ	50X
Send to off-chip DRAM	320 pJ	6,400X

* Data from J. Brunhaver, W. Dally, M. Horowitz, Stanford University

Figure 5: Energiekosten van transport

Welke mogelijkheden hebben we om de performance van computers te verbeteren? Stel we nemen als voorbeeld een toepassing die op een PC 10 uur rekentijd vergt, maar de klus moet in 1 uur klaar zijn. Wat zijn dan de opties? Je kunt 10 PCs nemen en als dat kan de taak netjes over alle PCs verdelen en ze parallel laten rekenen. Is een oplossing als je de mogelijkheid, de ruimte, het geld, en de energie hebt, maar wat indien dit embedded systeem net zo groot is als een mobiele telefoon? Daar passen niet 10 PCs in, en je hebt ook niet genoeg energie voor 10 PCs. Dan is het eenvoudig parallelliseren van het probleem niet altijd de beste oplossing. Ook wachten op een snellere chips heeft geen zin, want processor kernen worden vandaag de dag niet meer sneller, hooguit alleen wat zuiniger.
Een heel andere mogelijkheid is om voor elke toepassing een specifieke computer architectuur te ontwikkelen, zeg maar een maatpak computer: een computer die specifiek is ontwikkeld voor één bepaalde toepassing.
In het Engels heet dat een ASIC: Application Specific Integrated Circuit. ASICs zijn qua rekentijd en energieverbruik uitermate efficiënt, maar helaas niet flexibel en ook erg kostbaar om te ontwikkelen. Het ontwikkelen van een chip kan wel in de orde van een paar miljoen Euro liggen. We zijn daarom binnen embedded systemen voortdurend op zoek naar computer architecturen die flexibel en tegelijk efficiënt zijn. Een van de mogelijkheden is een z.g. her-configureerbare architectuur.
Een ASIC kan ongeveer 1000 keer zo efficiënt werken als een PC. Dus als we dat zouden vertalen naar rekentijd dan hoef je in plaats van 1 uur slechts 3,6 seconden te wachten op het antwoord. Herconfigureerbare architecturen, die een beetje flexibel zijn, zijn zo'n 100 keer efficiënter dan een PC, dus je bent in 6 minuten klaar. De combinatie van een PC met verschillende applicatie specifieke herconfigureerbare acceleratoren (een accelerator per toepassingsdomein) noemen we een heterogeen systeem, en dit is precies de trend in computer architectuur voor de komende jaren.

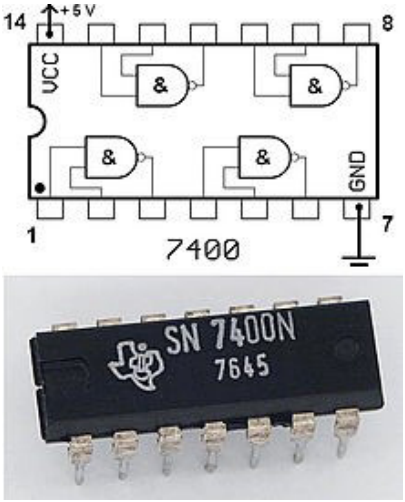


Figure 6: TTL poort

Om iets over herconfigureerbare architecturen te vertellen moet ik eerst wat terug gaan in de tijd. Toen ik in 1972 als student op de Technische Hogeschool Twente elektrotechniek studeerde kwam ik voor het eerst in aanraking met digitale techniek, de basis van alle computer architecturen. We ontwierpen de schakelingen toen nog met losse poorten, in die tijd was TTL de meest gangbare technologie. Eigenlijk ontwierpen we toen uitsluitend Applicatie Specifieke Systemen. Het aantal transistoren op een chip was toen nog zeer beperkt. De eerste transistor zag pas in 1948 het levenslicht. In 1960 ontwikkelde Fairchild het eerste IC (Integrated Circuit), tegenwoordig zouden we dat een chip noemen. Een bedrijf als Intel, een bedrijf dat groot is geworden met het maken van processoren op een chip, ontstond pas rond 1970.

We ontwierpen in 1972 dus digitale schakelingen met losse TTL poorten. Dit is een voorbeeld van een 7400 TTL IC, zo'n IC bevatte 4 NAND poorten. Voor de leken onder ons even weer wat uitleg over NAND poorten. Computers werken met bits, een signaal die de waarde 0 of 1 heeft, daarbij kun je nul associëren met 0V (of uit) en 1 met 5V (of aan). Met 1 bit heb je 2 combinaties 0 of 1, met 2 bits heb je 4 combinaties: namelijk 00 01 10 11, met 3 bits 8 combinaties, met 4 bits 16 combinaties etc.

Met logische poorten kunnen we bits aan elkaar combineren. Bekende logische poorten zijn de EN poort en de OF poort en de NIET poort. De functie van een poort kan weergegeven worden door een waarheidstabel. In deze tabel staat de functie van een poort of schakeling, het geeft de waarde van de uitgang voor alle mogelijke ingangswaardes. Op de lagere school noemden we dat een tafel, bv de tafel van 5.

A	B	Output
0	0	1
0	1	1
1	0	1
1	1	0

Figure 7: Waarheidstabel NAND poort

Zie hier een voorbeeld van de waarheidstabel van een NAND poort. Een NAND poort is een EN poort met een NIET poort erachter gezet.
Deze poorten kunnen vervolgens weer met transistoren gemaakt worden, de basis bouwstenen van de hedendaagse digitale elektrotechniek. Een NAND is een bijzondere poort, het blijkt namelijk dat je elke willekeurige digitale schakeling met uitsluitend NAND poorten kunt realiseren. Een NAND noemen we een universele bouwsteen.
Door NAND poorten te combineren kan elke willekeurige digitale schakeling, dus ook een digitale computer ontworpen worden. We maken een onderscheid in combinatorische schakelingen, waarbij de uitgang alleen afhankelijk is van de huidige waarde van de ingangen, en sequentiële schakelingen, die ook geheugen bevatten. In een geheugen kan de toestand van een signaal onthouden worden. Een bit kan dan opgeslagen worden in een zg. FlipFlop, waarvan de D FF de meest gebruikte is.
Een belangrijke eigenschap, die ik verderop gebruik, is dat elke gewenste digitale combinatorische schakeling met een niveau van EN poorten en een tweede niveau van OF poorten gerealiseerd kan worden.
Ontwerpen van digitale schakelingen is nu de kunst om met zo weinig mogelijk poorten en geheugen elementen een schakeling te realiseren die aan de eisen, bv qua performance, tijdseisen, of energie eisen voldoet.
Digitaal ontwerpen is afwegingen maken over ruimte (dwz aantal poorten en geheugens of wel chip oppervlakte) en tijd. In de loop der tijd is daar ook energie bijgekomen als ontwerp parameter. Deze afweging is voor elke toepassing weer anders, omdat er telkens andere eisen qua kostprijs, energie en tijd zijn.
Toen ik in 1972 studeerde was digitaal ontwerpen nog allemaal handwerk, tegenwoordig zijn daar uitstekende computer programma's voor om vanaf een specificatie van het gewenste gedrag een schakeling te synthetiseren.

Zoals eerder gezegd, werden rond 1970 digitale schakelingen ontworpen met losse poorten, omdat toentertijd de IC technologie nog niet beschikbaar was om veel poorten op een chip te zetten. Dit leidde vaak tot een hele printplaat vol ICs om een schakeling te realiseren. Die schakeling

had dan één specifieke functie (zeg maar een specifieke invulling van de waarheidstabel). Als de functie dus de waarheidstabel aangepast moest worden, kon je als je pech had de schakeling (dus printplaat) weggooien en een nieuwe gaan ontwerpen. Rond 1980 was de technologie zover dat er meer poorten op een chip geïntegreerd konden worden.

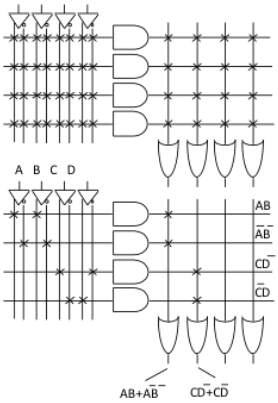


Figure 8: PLA struktuur

Met de ontwikkeling van zg PLAs, Programmable Logic Array veranderde er veel. PLAs maken gebruik van de eigenschap dat elke digitale schakeling met een twee niveau schakeling gemaakt kan worden: dus een EN niveau en een OF niveau. De eerste PLAs waren eenmalig programmeerbaar. Alle EN poorten waren verbonden met alle ingangen. Door nu de verbindingen in de chip letterlijk op te blazen (specifieke verbindingen letterlijk doorbranden), kun je elke gewenste combinatie van ingangen op de EN poort aansluiten. Dit proces noemen we programmeren van een PLA. Het type wat ik veelvuldig gebruikt heb zijn de zg PAL componenten, deze zijn wat eenvoudiger dan PLAs. Ze hebben namelijk OF poorten met een vast aantal ingangen. Ook deze waren weer eenmalig te programmeren, dwz dat je eenmalig een digitale functie in de PAL kunt stoppen: dus 'programmeerbare' digitale hardware. Zo was de PAL16R8 een veel gebruikte PAL. Een PAL met 16 ingangen en 8 uitgangen.

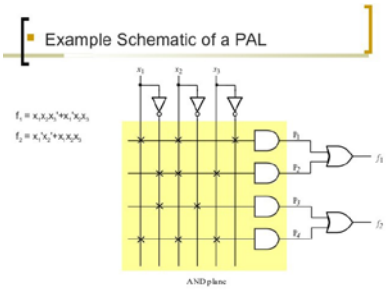


Figure 9: PAL struktuur

Met de PALs konden we al vrij complexe schakelingen maken. Met de komst PALs veranderde ook veel in het ontwerpproces van digitale functies. Waren de schakeling met losse poorten nog met de hand te ontwerpen, met de komst van PALs werd dit langzamerhand ondoenlijk. Er kwamen software programma's om de schakelingen te ontwerpen. Dit was ook de springplank voor zg hardware beschrijvingstalen en

hardware synthese tools zoals VHDL en Verilog. Wat echter hardware ontwerpen moeilijk maakt, en dat zien veel informatici vaak over het hoofd, is dat alle poorten parallel werken, iets wat voor ons mensen moeilijk te bevatten is. In de loop der tijd is dus digitale hardware ontwerpen programmeren geworden, maar dan wel in een exotische taal als VHDL.

Het nadeel van de PAL technologie was dat deze eenmalig te programmeren was, bij elke verandering in de functie kon de chip weggegooid worden. Je pakte dan een lege PAL en programmeerde de nieuwe functie erin. Dat veranderde met de komst van FPGAs, Field Programmable Gate Arrays rond 1990. Zoals eerder genoemd kunnen we de functie van een logische of digitale schakeling beschrijven met een waarheidstabel. Deze waarheidstabel kunnen we in een klein geheugentje zetten: de waarheidstabel van een 4 input EN poort ziet er zo uit.

I4	I3	I2	I1	AND 2	AND 3	AND4	OR2	OR3	OR4	XOR2	XOR3	XOR 4	EQ3	EQ 4	
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	
0	0	0	1	0	0	0	1	1	1	1	1	1	0	0	
0	0	1	0	0	0	0	1	1	1	1	0	1	0	0	
0	0	1	1	1	0	0	1	1	1	0	0	0	0	0	
0	1	0	0	0	0	0	0	1	1	0	1	1	0	0	
0	1	0	1	0	0	0	1	1	1	1	0	0	0	0	
0	1	1	0	0	0	0	1	1	1	1	0	0	0	0	...
0	1	1	1	1	1	0	1	1	1	0	1	1	1	0	...
1	0	0	0	0	0	0	0	0	1	0	0	1	1	0	...
1	0	0	1	0	0	0	1	1	1	1	1	0	0	0	
1	0	1	0	0	0	0	1	1	1	1	1	0	0	0	
1	0	1	1	1	0	0	1	1	1	0	0	1	0	0	
1	1	0	0	0	0	0	0	1	1	0	1	0	0	0	
1	1	0	1	0	0	0	1	1	1	1	1	1	0	0	
1	1	1	0	0	0	0	1	1	1	1	0	1	0	0	
1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	

Figure 10: Waarheidstabel in een geheugen

Als we nu een geheugen hebben met 16 posities van 1 bit, en we vullen dit geheugen met deze waardes, dan hebben we een 4 input EN poort gemaakt. Als we nu de inhoud van het geheugen aanpassen, dan hebben we b.v. een 4 input OF poort. Met een geheugen van 16 posities kunnen we 2^16 verschillende 4 input poorten maken (dat zijn er meer dan 65.000). Door nu zo'n geheugen als RAM uit te voeren (geheugen dat zowel gelezen als geschreven kan worden) kunnen we door de inhoud van het geheugen te veranderen de logische functie aanpassen en hebben we dus her-programmeerbare logica. We hoeven dan het IC niet weg te gooien als de functie aangepast moet worden, maar kunnen deze her-programmeren.

De eerder genoemde FPGA heeft CLBs (Configurable Logic Blocks) als basisblokken, die bestaan uit twee geheugens voor de combinatorische functie en 2 FlipFlops voor de sequentiële functie. Zo'n CLB van het bedrijf Xilinx ziet er sterk vereenvoudigd zo uit: 2 geheugens voor de logische functie en twee FlipFlops.

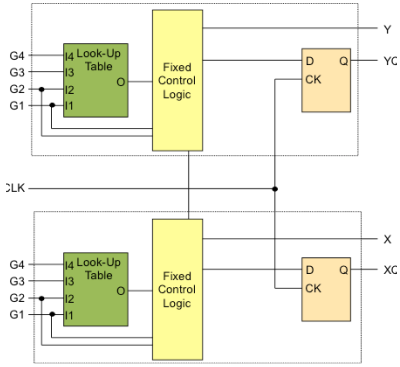


Figure 11: CLB struktuur

Door nu een soort dambord te maken van CLBs, waarbij de in- en uitgangen van de CLBs verbonden zijn, kunnen groteren schakelingen gerealiseerd worden. Hier een sterk vereenvoudigd voorbeeld van een Xilinx FPGA. Dit ziet er nog betrekkelijk eenvoudig uit, maar in de praktijk zijn dat honderden verbindingen. Merk wel op dat voor een 4 input digitale functie we nu dus een geheugen hebben van 16 posities, en dat kost heel wat meer transitoren dan een directe implementatie van de functie met transistoren, maar we creëren zo wel flexibiliteit. Ook hier geldt weer flexibiliteit kost chip oppervlakte en extra energie en tijd.
In de beginjaren werden FPGAs gebruikt voor de vervanging van losse poorten. In de loop van de jaren ontwikkeld Xilinx steeds grotere FPGAs met duizenden CLBs waar complete systemen op gerealiseerd kunnen worden. Zoals net opgemerkt levert een logische functie gerealiseerd met CLBs niet altijd de meest efficiënte oplossing qua chip oppervlakte en energie verbruik, daarom zijn er in de loop der tijd ook steeds meer complexere componenten aan de FPGAs toegevoegd: vermenigvuldigers, geheugens en zelfs complete processoren.
Een voorbeeld is de recente Zinq Ultra Scale FPGA van Xilinx met maar liefst 7 ARM processoren aan boord, 3.000 vermenigvuldigers, 50 MB geheugen, 300.000 CLBs en allerlei snelle communicatie links. Deze chip heeft maar liefst 2577 aansluit pinnen, heel wat meer dan de 16 uit 1970. Het programmeerbare deel van de chip is equivalent met 600.000 logische poorten, heel wat meer dan 5 van de 7400 uit de TTL technologie van 1972. U kunt zich voorstellen dat hiermee heel complexe systemen gerealiseerd kunnen worden. Het realiseren van een toepassingen voor zo'n chip is bovendien navenant ingewikkeld. Een van de huidige uitdagingen is dan

ook, om uitgaande van de specificatie van een complex probleem een ontwerp te maken voor zo'n FPGA die gebruik maakt van de uitgebreide mogelijkheden van de moderne FPGAs. In de CAES groep werken we hier hard aan mee, o.a. met de tool Cλash die uitgaande van een hoog-niveau specificatie in Haskell een VHDL of Verilog specificatie kan generen. Met deze VHDL/Verilog specificatie kan met standaard tools een FPGA geprogrammeerd worden. Deze Cλash tool wordt nu naar de markt gebracht door het start-up bedrijf Qbay Logic van collega Jan Kuper en ex-promovendus Christiaan Baaij. De complexiteit en mogelijkheden van on-chip processoren en de programmeerbare logica levert heel wat nieuwe uitdagingen op en nog genoeg werk voor toekomstige promovendi.
Een ander pad die we in de loop der jaren hebben ontwikkeld is niet de eerder genoemde CLBs als basisbouwstenen te gebruiken, maar super eenvoudige processoren, die heel efficiënt werken. Een CLB werkt met bits als elementaire data eenheid, terwijl processoren met woorden van 8, 16 of 32 bits werken. Een voorbeeld die ik wil noemen is de ontwikkeling van de Montium processor die binnen CAES is ontwikkeld, en door het start-up bedrijf Recore verder is ontwikkeld. Bij Recore, opgericht door drie ex-promovendi Paul Heysters, Lodewijk Smit en Gerard Rauwerda, werken inmiddels vele ex-afstudeerders en ex-promovendi. De Montium werkte met 16 bits woorden, en had naast rekenfuncties ook geheugen aan boord. Hier ziet U een blokschema van de Montium, met rekenfuncties (b.v. vermenigvuldigen en optellen).

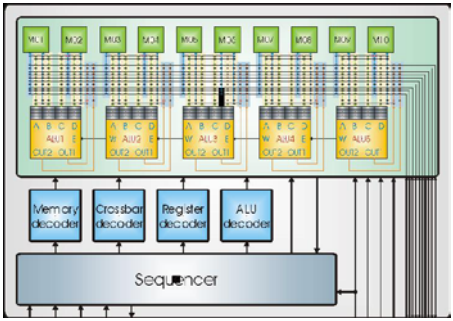


Figure 12: Montium processor kern

Met name door het samenbrengen van geheugen en processing kunnen de communicatielijnen kort zijn (want communicatie over een lange afstand is duur) kan de Montium heel efficiënt werken in zijn toepassingsdomein. De Montium is specifiek bedoeld voor Digital Signal Processing. Hier een foto van een chip met 4 Montiums, die we samen met Recore gerealiseerd hebben, waarop we digitale radio en video (DAB en DMB) gerealiseerd hebben. Ook deze chip is her-programmeerbaar, door het programma in de Montiums aan te passen kunnen we de functie van de chip aanpassen,

zonder dat we de chip hoeven weg te gooien. Ook voor deze chip gold weer een groot deel van de complexiteit zit in het applicatie development proces.
Heel wat uitdaging en promotiewerk zit in het programmeren van dit soort complexe heterogene herconfigureerbare architecturen. Niet alleen bij Recore maar ook in nauwe samenwerking met NXP o.a. via het onderzoek van Marco Bekooij en zijn promovendi.
Nu zult U wellicht denken dat inmiddels alle problemen wel zijn opgelost: dit is verre van waar: er is altijd de wens van een beter efficiëntie (meer operaties per Watt ofwel 1pJ per instructie), promovendi van o.a. collega Andre Kokkeler werken aan approximate computing (misschien heb je niet overal evenveel precisie nodig), en Marco Bekooij met zijn promovendi werkt aan betere tools om Real-Time toepassingen te modelleren en efficiënter op complexe heterogene architecturen af te beelden, en liggen nog genoeg uitdagingen t.a.v. betrouwbaarheid en veroudering van ICs waar Hans Kerkhoff en zijn promovendi aan werken. Er zijn voldoende nieuwe uitdagingen te vinden binnen embedded systemen waar jonge onderzoekers hun tanden op stuk kunnen bijten.

Ik ga nu over op het tweede onderwerp Smart Grids.
Rond 2005 heb ik samen met Johann Hurink een nieuw onderzoeksgebied aangeboord, dat van energie management van energie stromen op wijk niveau. Deze samenwerking was voor CAES een nieuw onderwerp maar ook heel vruchtbaar en ik denk ook voor DMMP. Eigenlijk ben ik daar bij toeval ingerold door een contact met Simon Kolin, maar wat is toeval? Het onderwerp energietransitie spreekt vele mensen aan, en daar kun je bij verjaardagsfeestjes een goede discussie over voeren want iedereen heeft daar wel een mening over, meer dan over efficiënte computer architecturen heb ik geleerd.
De laatste jaren heb ik mij daarbij weer gericht op de extremen, hoe kunnen we 100% duurzame gebouwen / wijken / steden realiseren. Het is ons gaandeweg steeds meer duidelijk geworden dat energietransitie veel meer is dan vermogenselektronica. De bijdrage van informatica en wiskunde, en zeker de combinatie is daar van elementair belang. Ook in dit gebied is de samenwerking en integratie van kennis uit verschillende disciplines (van technische tot maatschappelijke) van groot belang. Dit onderwerp levert voor de UT (bij uitstek High Tech Human Touch) en als ondernemende universiteit grote kansen.

Helaas kan ik ook nu maar één voorbeeld wat nader toelichten. Ik wil een scenario uitleggen waarbij een blok van 16 huizen (het mogen er ook 160 zijn) het hele jaar door op een duurzame manier van energie kan worden voorzien. Alle huizen krijgen 15 m^2 zonnepanelen op het dak en de huizen zijn redelijk goed geïsoleerd (volgens de laatste bouwvoorschriften). Verder hebben de huizen een accu van plm. 2 kWh (pakweg 2 tot 3 auto accu's).

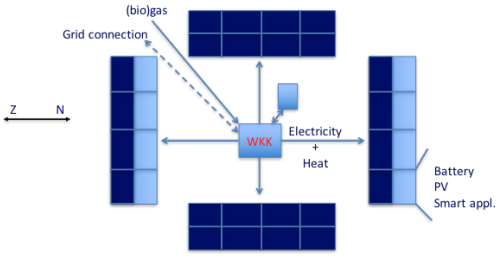


Figure 13: 16 huizen concept

Met 15 m^2 zonnepanelen en de accu's kunnen de huizen in de zomer voldoende energie produceren om hun eigen verbruik te kunnen afdekken. In de winter ligt dat wat anders: op de eerste plaats omdat de zonnepanelen dan niet zoveel opleveren en omdat er ook een warmtevraag is in de winter. In ons concept hebben we centraal in de wijk een WKK (Warmte Kracht Koppeling) systeem. Een WKK is in zijn eenvoudige vorm een motor die op gas (in ons geval bio-gas) loopt die warmte en elektriciteit levert. De WKK is warmte-gedreven, dwz. deze zal alleen lopen als er een warmtevraag is. De warmte wordt met een klein lokaal warmtenet naar de huizen gebracht en de elektriciteit wordt ook aan de huizen geleverd om het tekort aan zonne-energie aan te vullen. We hebben hier uitgebreide simulaties aan gedaan, waaruit blijkt dat het concept realistisch is. We zijn nu in gesprek met partijen om dit in de praktijk te gaan uitvoeren.
Ik zal een paar eerste resultaten laten zien.

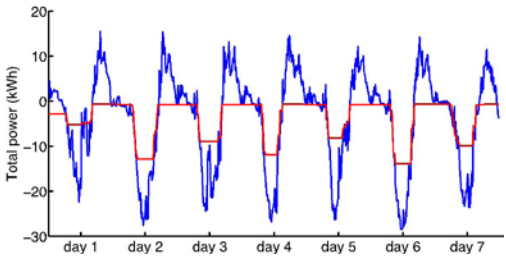


Figure 14: Totaal vermogen in de zomer

Deze figuur laat een week in de zomer zien met mooi weer. De blauwe lijn geeft het vermogen dat door de wijktransformator gaat als we geen ICT en accu's gebruiken. Het typische gedrag is dat overdag (als de zon schijnt) elektriciteit wordt terug geleverd,

maar in de avond/nacht wordt weer elektriciteit terug gehaald uit het net. Dit staat ook wel bekend als nul-op-de-meter. Dit levert heel grote pieken op voor het elektriciteitsnet, en als je dit met veel meer dan 16 huizen doet, zal de netwerkbeheerder nieuwe kabels moeten gaan trekken, omdat het elektriciteitsnetwerk de piekbelasting aan moeten kunnen. De rode lijn geeft aan wat er gebeurt als we wel ICT en accu's gebruiken. Het gedrag is veel minder grillig, maar we zien wel dat we altijd energie over hebben. Daarmee zouden we in de zomer ook nog een elektrische auto kunnen opladen.

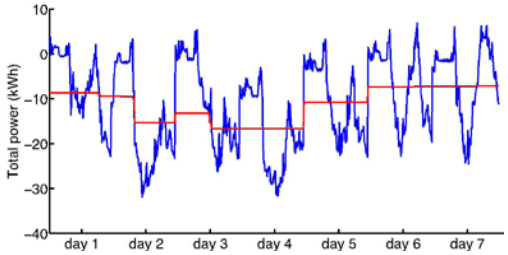


Figure 15: Totaal vermogen in de winter

In de winter is er ook een overschot, maar nu omdat de WKK warmte moet produceren en dus ook elektriciteit. Ook hier is het resultaat met ICT besturing aanzienlijk beter. Ook nu hebben we in de winter weer een flink overschot. De moeilijkste week was een week in oktober met relatief hoge temperaturen (dus de WKK loopt niet omdat er geen warmtevraag is) en weinig zon (dus de zonnepanelen leveren niet veel op). In het ICT gecontroleerde scenario is het systeem vrijwel in balans, maar zonder ICT zien we weer een piekerig patroon.

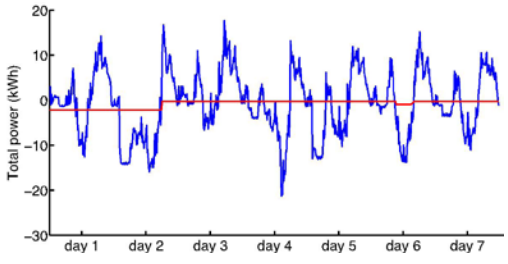


Figure 16: Totaal vermogen in de herfst

Onze conclusie is, dat met zonnepanelen en een beetje bio-gas (misschien uit mest vergisting) de wijk 100% op duurzame energie kan werken. Bovendien dit concept schaalt naar grotere eenheden met meer huizen. Ik denk dat dit soort simpele concepten, die we op de UT ontwikkeld hebben, een belangrijke bijdrage kan leveren aan de energietransitie in de wereld.

Een belangrijk uitgangspunt is dat de mensen zo zuinig mogelijk omgaan met de beschikbare energie, dat de flexibiliteit die in het gebruik zit optimaal wordt benut en dat we alle beschikbare vormen van energie benutten.
We werken nauw samen met het start-up bedrijf Ipsum van Peter de Bie, waar inmiddels 4 ex-promovendi van CAES werken. Dit bedrijf ontwikkelt algoritmes om uitgaande van data van de elektriciteitsmeter in de meterkast inzicht te geven in het energieverbruik van huishoudens en dus ook inzicht te geven in de flexibiliteit van huishoudens. Met dit inzicht kunnen ze de bewoners aanbevelingen geven hoe zij hun verbruik kunnen verminderen, of beter kunnen optimaliseren. Maar dit kan ook inzicht geven in de flexibiliteit van huizen die we op wijkniveau weer kunnen gebruiken.

DANKWOORD

In de afgelopen jaren ben ik indiener of mede indiener geweest van vele subsidieprojecten bij de EU FP6, FP7, H2020, en de nationale subsidieverstrekkers zoals STW (nu TTW), NWO, RVO of rechtstreeks van bedrijven. Ik wil alle subsidieverstrekkers van harte bedanken voor het gestelde vertrouwen. Ik heb daarbij veel support gekregen van bedrijven, waarvoor ik hun hartelijk wil danken. Zoals in mijn rede aangegeven, heb ik met vele start-up bedrijven samengewerkt, of bijgedragen aan hun totstandkoming. Ik heb dat met veel plezier gedaan, en ik wil dat in de toekomst nog graag blijven doen. Het enthousiasme van dit soort bedrijven werkt aanstekelijk, en is ook belangrijk voor de maatschappelijke relevantie van het onderzoek van de UT. Bovendien zijn deze bedrijven een belangrijke inspiratiebron voor nieuwe ideeën en nieuwe samenwerkingsprojecten.
Daarnaast wil ik alle universitaire collega's in den lande en Europa hartelijk bedanken voor de prettige samenwerking bij het opzetten van de programma's. Ik denk bv met genoegen terug aan de goede samenwerking met de TU Eindhoven en TU Delft bij de recent gehonoreerde STW perspectief programma's Robust design of Cyber Physical Systems en ZERO - over energie autonome systemen. Zonder deze prettige samenwerking was dit nooit tot stand gekomen, en was mijn werk een stuk minder aantrekkelijk.
Ik wil de collega's binnen EWI en de UT bedanken voor de samenwerking. Met Paul Havinga heb ik in de begin jaren veel projecten samen gedaan. Met Bram Nauta hebben we in de loop der jaren een vruchtbare brug geslagen tussen de analoge wereld van ICD en de digitale wereld van CAES. Met Johann Hurink is de samenwerking zo nauw, dat Johann bijna een onderdeel van de CAES groep is geworden.
Ik wil de ondersteunende diensten van de faculteit EWI bedanken en het management van de faculteit, in het bijzonder de decaan, bedanken die het mogelijk hebben gemaakt dat ik dit werk kon doen. Ik was het niet altijd eens met alle beslissingen, daarom ik wil nog een suggestie aan de decaan geven. Wees niet bang om eens met een out-of-the-box voorstel in zee te gaan. Zonder dat was ik nooit aan energie management voor Smart Grids begonnen (waar we nu nationaal en internationaal voorop lopen), en hadden we niet aan Hermite communicatie gewerkt, met een cum-laude promotie van Wim Korevaar tot gevolg. We leiden veel hele

slimme en getalenteerde promovendi op, die een voor een de UT verlaten. Er moet toch een mogelijkheid zijn om een paar aan ons te binden, zonder dat ze tegelijk door drie hoepels van een tenuretrack procedure hoeven te springen. Laat ze werken in een inspirerende omgeving met medewerkers, AIO's, post-docs en studenten, geef ze de ruimte, daag ze uit en er komen gegarandeerd fantastische en vooral onverwachte resultaten uit. Zoals 1500 (proef-)ballonnen in mijn kamer na terugkeer van de vakantie. Jonge mensen de kans en de ruimte geven is in het kort ook ongeveer de aanpak die ik bij CAES heb gevolgd.

Ik heb mogen werken in een geweldige groep met hardwerkende en plezierige collega's, AIOs en post-docs. Bert Molenkamp als steun en toeverlaat in het onderwijs en bij de master Embedded Systems, Andre Kokkeler als harde en gedegen werker en een belangrijke schakel met de ICD groep, Marco Bekooij als ons interface met NXP, Sabih Gerez als chip designer, Hans Kerkhoff die op het gebied van betrouwbaarheid de laatste jaren zeer succesvol was binnen de groep en Jan Kuper als gepensioneerde Cλash/Haskell goochelaar. Marco Gerards is recentelijk hieraan toegevoegd die wiskunde met informatica combineert in het Smart Grid gebied. En natuurlijk niet de vergeten de ondersteuning van Bert, Marlous en Nicole. Marlous, zonder jouw beschermende vleugels was mijn agenda helemaal een ratjetoe geweest, en liepen allerlei processen, zoals de organisatie van de succesvolle Energy-open workshop van een paar weken terug en de organisatie van mijn afscheidsrede, zeker niet zo soepel.

Een hele belangrijke bijdrage is geleverd door de vele master studenten, promovendi en post-docs. Zonder hun oorspronkelijke, inspirerende en frisse ideeën was dit vak lang niet zo leuk. Ik hoop dat er snel een opvolger voor mij gevonden wordt, zodat de groep de komende jaren weer vooruit kan.

Ik wil mijn broers en zussen bedanken voor hun betrokkenheid. Onze broer Jan is helaas niet meer onder ons, maar zal van boven deze afscheidsrede waarschijnlijk fijntjes observeren.

Tenslotte, Eefje, Jacob, Maarten, Danique en Jasper, ik hoop dat jullie niet teveel tekort heb gedaan, met mijn enthousiasme voor mijn werk. Ik hoop dat ik een beetje van het enthousiasme aan jullie heb overgedragen.

En lest best Alda, zonder jouw steun had ik dit allemaal niet kunnen doen. Je hebt vaak, overigens terecht gemopperd, alweer werk, leg dat proefschrift toch eens weg. Onderzoek in computer architectuur is het vinden van de juiste balans tussen tijd en ruimte. Dit geldt ook voor een relatie, ik hoop dat de balans de komende tijd wat rechtgetrokken wordt.

Ik heb gezegd.

UNIVERSITEIT TWENTE.
POSTBUS 217
7500 AE ENSCHEDE