



Rekonfigurierbare CPUs

Softe Hardware

Dr. Klaus Schlüter • Kaiserslautern? Traditionsreiche Uni-Idylle sieht anders aus. Doch originelle Geister finden die Atmosphäre anregend. Professor Reiner Hartenstein vom Fachbereich für Informatik an der Universität hat hier ein neues Rechnerprinzip ausgetüftelt – die Sonne geht auf über der Pfalz.

Solange ich tat, was alle tun, hatte ich nur Freunde. Auf Widerstand stieß ich erst, als ich etwas Neues zu bieten hatte, etwas, das im Lehrgebäude der Informatik noch nicht verankert war.“ So faßt Professor Reiner Hartenstein, Inhaber des Lehrstuhls für Rechnerstrukturen der Universität Kaiserslautern, eine Erfahrung zusammen, die auch schon andere Vordenker gemacht haben.

Albert Einstein brachte es auf den Punkt: „Eine wirklich gute Idee erkennt man daran, daß ihre Verwirklichung von vornherein ausgeschlossen erscheint.“

Hartenstein verfolgt ein ehrgeiziges Ziel: Er möchte ein Rechnerprinzip etablieren, welches das von-Neumann-Konzept auf den Kopf stellt. Seine Xputer sequenzieren nicht die Befehle, sondern die Daten. An die Stelle des Befehlszählers tritt ein

Datenzähler. Xputer lösen daher Aufgaben mit regelmäßigen Datenabhängigkeiten besonders flott: bis zu 2000mal schneller als konventionelle Computer, wie Messungen ergaben. Die meisten numerischen Probleme in den Naturwissenschaften fallen in diese Kategorie. Und in der elektronischen Signalverarbeitung könnten Xputer teure Digital Signal Processors (DSP) ersetzen. Einziges Handicap: Die im

Labor gefertigten Prototypen waren bisher zu klein, um damit umfangreiche Probleme angehen zu können.

■ Das Prinzip

Wie die meisten genialen Ideen haben auch die Xputer etwas bestechend Einfaches. Aber, um es mit den Worten des Physik-Nobelpreisträgers Richard Feynman zu sagen: „Hört mal, Jungs! Wenn ich Euch in einer

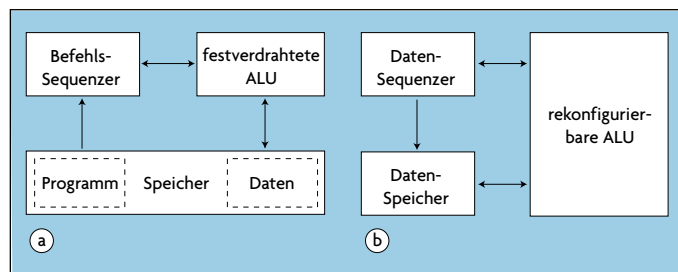


Bild 1. Unterschiede zwischen Computern (a) und Xputern (b): Sowohl die ALU als auch der Sequenzierer spiegeln die jeweiligen Paradigmen wider.

Minute sagen könnte, was ich getan habe, dann wäre das sicher nicht den Nobelpreis wert.“ Also, krepeln wir die Ärmel hoch und versuchen zu verstehen, was Xputer so einzigartig macht.

Da ist, neben dem Daten-Sequenzierer, auch noch eine „reconfigurable“ Arithmetic Logic Unit (rALU, Bild 1). Sie kann, im Unterschied zu den festverdrahteten ALUs herkömmlicher CPUs, zur Laufzeit programmiert werden: eine softe Hardware, sozusagen. Ihre Bausteine sind – wie Hartenstein sie nennt – „reconfigurable“ Application Specific Integrated Circuits, zu denen die Field Programmable Gate Arrays (FPGA) ebenso zählen wie andere Programmable Logic Devices (PLD). Für die Möglichkeit, die Hardware umprogrammieren zu können, ist leider ein heute noch hoher Preis zu entrichten: Die softe Hardware verbraucht mehr Silizium als die festverdrahtete. Folge: Sowohl die Kosten als auch der Energieverbrauch der Chips steigen.

Aber: Bei der nach der Jahrtausendwende zu erwartenden 0,1-Mikron-Technologie spielt das vielleicht gar keine Rolle mehr. Höchste Zeit also, das neue Computerparadigma unter die Leute zu bringen. „Wir brauchen eine Generalmobilisierung des Technik-Bewußtseins“, sagt Hartenstein und hat dabei nicht nur seine Xputer im Blick, wie er im Interview „Made in Germany – da ist der Wurm drin“ erklärt.

Xputer verfügen über eine Reihe von Feldregistern, die Scan Fenster oder Scan Cache genannt werden (Bild 2). Sie sind größenverstellbar und können wie ein adressierbares Fenster über den zweidimensionalen Datenspeicher gelegt werden. Der Scan Cache wird entlang eines Pfades über den Datenspeicher bewegt, der Scan-Muster oder Scan Pattern heißt (Bild 3). Den Datenaustausch zwischen Scan Cache und Datenspeicher regelt ein Daten-Sequenzierer, der ein festverdrahtetes Repertoire von Daten-Adreßfolgen bereitstellt.

Xputer im Internet

Homepage, englisch <http://xputers.informatik.uni-kl.de>
Homepage, deutsch http://xputers.informatik.uni-kl.de/index_d.html
Homepage für langsame Verbindungen http://xputers.informatik.uni-kl.de/index_low_speed.html
Mitarbeiter, englisch <http://xputers.informatik.uni-kl.de/staff/staff.html>
Mitarbeiter, deutsch http://xputers.informatik.uni-kl.de/staff/staff_dt.html
Forschungsarbeit der Arbeitsgruppe http://xputers.informatik.uni-kl.de/index_academic.html
Xputer http://xputers.informatik.uni-kl.de/xputer/index_xputer.html
Xputer-Animationen http://xputers.informatik.uni-kl.de/xputer/intro_to_xputers.html
FAQ <http://xputers.informatik.uni-kl.de/FAQ-Pages/FAQaFQA.html>
Hardware/Software Co-Design http://xputers.informatik.uni-kl.de/CoDe-X/index_codex.html
MoPL-3 <http://xputers.informatik.uni-kl.de/~wolfgang/index.html>
VLSI-CAD-Gruppe http://xputers.informatik.uni-kl.de/vlsi/index_vlsi_cad.html

Ein einfaches Beispiel demonstriert das Zusammenspiel der Komponenten. Bild 4a beschreibt einen Algorithmus, wie er in einer Hochsprache formuliert sein könnte. Bild 4b zeigt den dazugehörigen Datenabhängigkeitsgraphen, aus dem der Compiler (Xpiler) eine Speicherkartierung ableitet (Bild 4c). Er spezifiziert Scan-Cache-Format und rALU-Teilnetze (Bild 4d), die sich wie Verbundoperatoren über das aktive Scan-Fenster legen. Schließlich legt der Xpiler auch das Scan Pattern fest (Bild 4e). Am Ende des Scan Pattern

steht eine Markette, ein sogenanntes Tagged Control Word (TCW). Es ist als einzige Anleihe an Neumann unverzichtbar. Als „klassische“ Kontrollmechanismen sind nur bedingte Verzweigungen nötig, denn der Xpiler kennt noch andere unkonventionelle Entscheidungsmechanismen – keine Befehle, sondern Verbundoperatoren, Scan Caches oder Scan Patterns. Da TCWs nur sehr selten vorkommen, übertrifft die Datenkomponente den Kontrollfluß bei weitem. Der große Vorteil gegenüber der von-Neumann-Architektur: Der Compi-

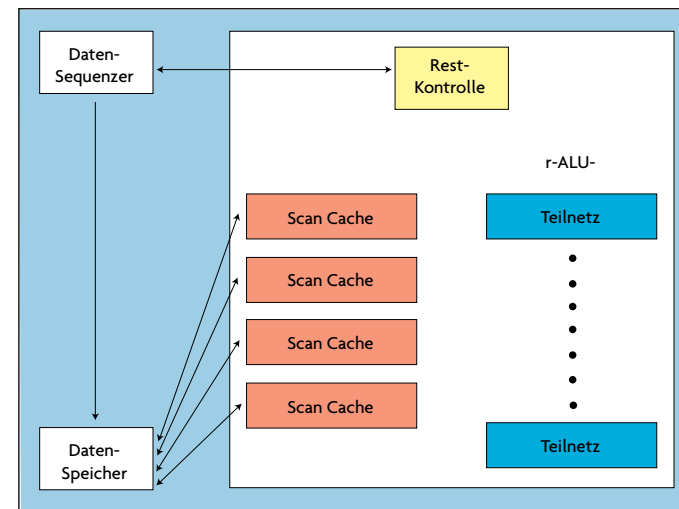


Bild 2. Datenaustausch zwischen Scan Cache und Datenspeicher: Der Daten-Sequenzierer stellt ein festverdrahtetes, aber hoch parametrisiertes Repertoire von Daten-Adreßfolgen bereit.

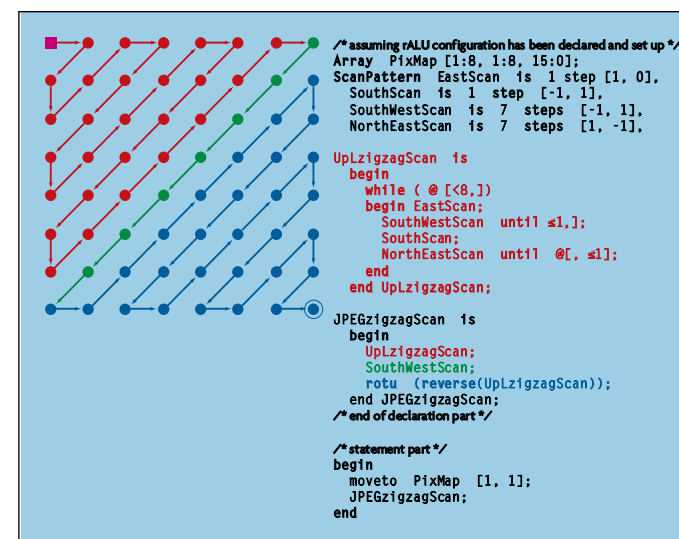


Bild 3. Scan Pattern für den JPEG-Algorithmus: Selbst komplexe Datenzugriffspfade sind in MoPL einfach zu beschreiben.

ler muß nur wenige Speicheradressen berechnen.

Die Software

Wie sind nun Xputer zu programmieren? Auch darüber hat sich die Gruppe rund um Hartenstein Gedanken gemacht und eine eigene Programmiersprache für den Xputer und seinen Host entwickelt. Holprig

ist allenfalls ihr Name: q-X-C – q wie Quelle, X wie Xputer und C wie ANSI-C. Das X heißt auch Map oriented Programming Language (MoPL) und ist ein Supplement von ANSI-C. Es spiegelt die Datenfluß-getriebene Hardware wider und setzt eine detaillierte Kenntnis des Xputer-Paradigmas voraus.

Vor dem eigentlichen Übersetzungsschritt (Bild 5) trennt ein

X-C-Filter und Optimierer den Code für Host und Xputer. Diejenigen C-Konstrukte, welche der Xputer nicht beschleunigen kann, sind in einer Untermenge der Sprache C, Host C (hC), ausgewiesen, alle anderen in Accelerator C (aC). Reinen MoPL-Code kann nur der Xputer ausführen. Der X-C-Filter und Optimierer ergänzt den hC-Code um Synchronisations- und

Kommunikations-Routinen, die dann ein GNU-C-Compiler in Host Code übersetzt. Den aC-Code strukturiert und partitioniert ein sogenannter aC-Compiler, bevor er ihn in Generic Address Generator (GAG) Code und Arithmetic and Logic Expressions for Xputers (ALE-X) übersetzt. Somit realisiert der aC-Compiler den Übergang vom Kontrollfluß- zum Datenfluß-Paradigma. Den MoPL-Code übersetzt ein MoPL-Compiler in GAG- und ALE-X-Code. Ein GAG- sowie ein ALE-X-Linker führen sequentielle und strukturelle Codes in der richtigen Reihenfolge zusammen.

Listing 1 beschreibt einen Weichzeichner-Algorithmus in q-X-C. Der X-C-Filter und Optimierer zerlegt das Programm in einen aC- und einen hC-Code (Listing 2 und 3). Aus dem aC-Code erzeugt der aC-Compiler sowohl den GAG-Code als auch den ALE-X-Code, deren Abdruck wir uns hier gespart haben. Dafür hat Jürgen Becker, Doktorand bei Hartenstein, den Weichzeichner für DOS auch in MoPL programmiert und ausführlich kommentiert (Listing 4). Bild 6 zeigt das Resultat seiner Bemühungen.

Die Realisierung

Hartensteins neues Paradigma steht dem Neumanns diametral gegenüber. Doch warum interessiert sich eigentlich hierzulande die Industrie so wenig dafür? Schließlich sind Xputer ebenso universell wie Computer. Und genau wie bei diesen sind auch unterschiedliche Architekturen realisierbar. Daß rekonfigurierbare ALUs ganz neue Möglichkeiten bieten, haben englische Forscher jedenfalls erkannt. Am Department of Electrical and Electronic Engineering des Imperial College in London hat eine Gruppe um Peter Cheung zusammen mit den Hewlett Packard Labs und der Firma Xilinx ein PCI-Bus-Board entwickelt, das dynamisch rekonfigurierbare FPGAs

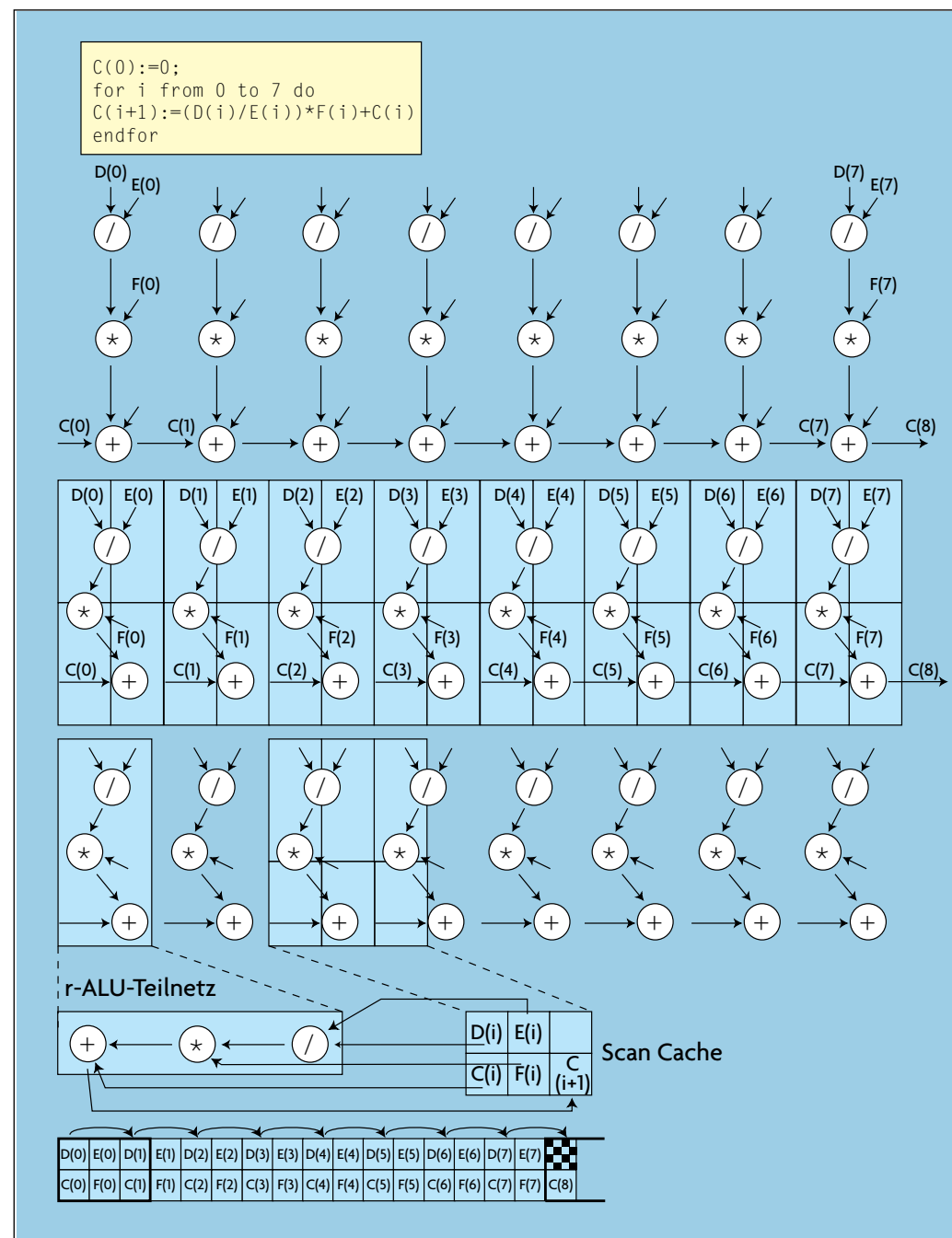


Bild 4. Zusammenspiel aller Xputer-Komponenten: Der Compiler analysiert einen vorgegebenen Algorithmus, in (a) als Textprozedur und in (b) als Datenabhängigkeitsgraph dargestellt. Er generiert einen Belegungsplan des Speichers (c) und spezifiziert Scan-Cache-Format und rALU-Teilnetze (d) sowie das Scan Pattern (e).

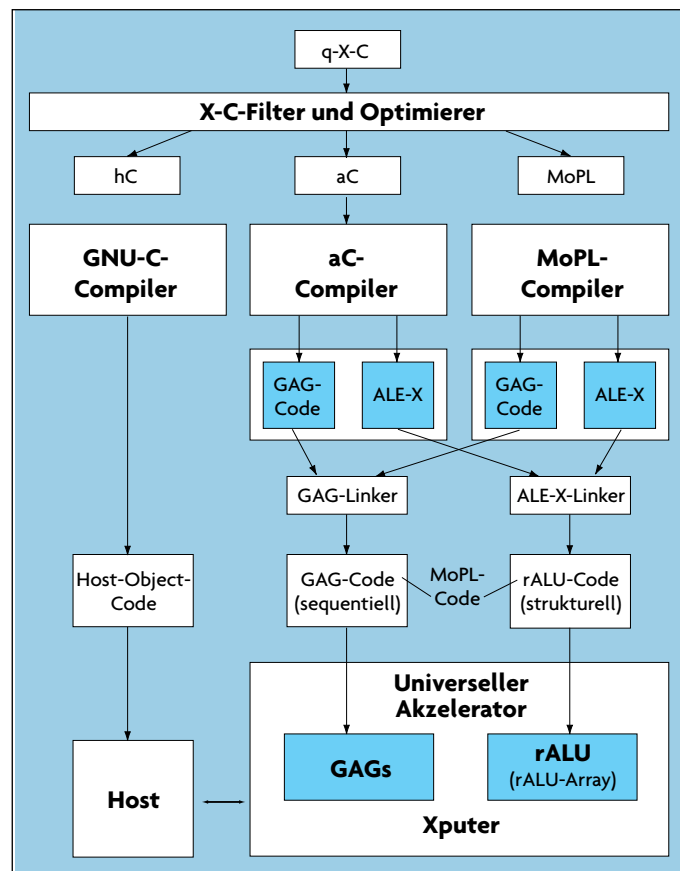


Bild 5. Co-Compiler von Host und Xputer: Die verglichen mit Computern stark verkürzte Laufzeit macht den etwas aufwendigeren Compiler-Gang bei weitem wett.

enthält. Den ersten Xputer, die MoM-1, schuf Hartensteins Gruppe in den 80er Jahren. Ihr Name steht für Map oriented Machine, first generation. „Map“ deutet die zweidimensionale Speicherorganisation an. Neben dem ersten Data Sequencer entwickelten die Forscher ein Dynamically Programmable Logic Array (DPLA), dessen Leistungsdaten damals die kommerzieller FPGAs angeblich übertroffen haben. Die MoM-1 war überragend schnell. Bei einem Design Rule Check mit einem der Bildverarbeitung entlehnten Pattern-Matching-Verfahren erzielte sie bei 800 Referenzmustern ein Speed Up von 2300 gegenüber einer VAX 11/750. Mit anderen Worten: Sie war mehr als 2000mal schneller als ein für damalige Verhältnisse rasanter Super-Minicomputer. „Natürlich waren die Fachkollegen skeptisch angesichts einer sol-

chen Beschleunigung“, erinnert sich Hartenstein. „Ein ähnlich überzeugendes Ergebnis haben wir seitdem nicht mehr erzielt. Nur um zehn oder hundert, das waren sonst übliche Akzelerations-Werte.“ Um die Ergebnisse zu erklären, verglichen die Forscher die Operations-Mechanismen von Computern und Xputern. Dabei stellten sie fest, daß Computer besonders viel Zeit zum Adressieren des Speichers benötigen: bis zu 90 Prozent der Gesamt-rechenzeit nämlich. Xputer hingegen sind darin recht sparsam und müssen nur wenige Adressen explizit berechnen.

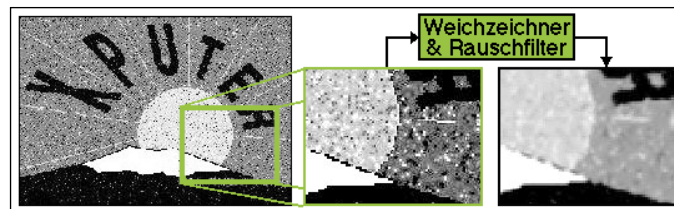


Bild 6. Weichzeichnen mit dem Xputer: Das Resultat der Eingabe (a) ist ein geglättetes Bild (b).

Zur weiteren Beschleunigung trägt die Mikro-Parallelität der rALUs bei: die Software-zu-Hardware- oder besser die Software-zu-„Configware“-Migration. Außerdem findet eine Laufzeit-zu-Compile-Zeit-Migration statt, sprich: Compilieren dauert bei Xputern länger als bei Computern; dafür sind Xputer in der Ausführung schneller. Im Gegensatz zur MoM-1 war der zweite in Kaiserslautern realisierte Xputer skalierbar, die MoM-2: Bis zu 15 rALUs konnten parallel betrieben werden. Allerdings erschien den Forschern die Architektur nicht einfach und nicht elegant genug. Eine rALU aus kommerziellen Bausteinen beanspruchte einerseits viel Platz und war andererseits sehr teuer. Deshalb entwickelte Reiner Kress, ein ehemaliger Doktorand von Hartenstein, die MoM-3 mit dem nach ihm benannten, skalierbaren rALU-Array. Der Pfiff dabei: Das dazugehörige Compiler-Backend, das sogenannte Data Path Synthesis System (DPSS), macht alle Routing- und Placement-Algorithmen obsolet, an denen sich schon Heerscharen von Forschern die Zähne ausgebissen haben.

Die Zukunft

Den Glauben an einen totalen Sieg über Neumann hat Hartenstein längst aufgegeben: „Das wäre Größenwahnsinn! Es gibt zuviel Software, die neu geschrieben werden müßte.“ Aber: Obwohl sich die Performance der Neumann-Rechner alle 18 Monate verdoppelt, stoßen sie bei zahlreichen zeitkritischen Anwendungen immer wieder an Grenzen. Intel

hat einen Weg aus der Krise aufgezeigt: Die Hardware wird komplexer, Stichwort MMX. Damit sind die Maschinen zwar immer noch universell programmierbar, aber die Anzahl der Maschinen-Befehle wächst. Da zudem viele der festverdrahteten Kommandos nur selten benutzt werden, liegt ein Teil des Siliziums ständig brach. Auch das wäre nicht schlimm, würde dabei keine Energie verschwendet. Ob Intel – und all die anderen, die versuchen gleichzuziehen – da auf dem richtigen Dampfer ist? Es wäre doch wieder mal an der Zeit, etwas tatsächlich Neues auszuprobieren und Xputer zu kommerzialisieren. Doch wer sollte das tun? Wo sind die Firmen, die innovative Ideen zu marktreifen Produkten machen? Im Ausland? Wir Deutschen sind gebildet wie nie zuvor. Viele von uns haben ein Hochschuldiplom. Wir – „Weltmeister in Sachen Komplexität“ – hätten das Zeug, wieder etwas zu bewegen. Aber wir tun es nicht! *sk*

Literatur

- [1] H. Grünbacher, R. W. Hartenstein: Field-Programmable Gate Arrays: Architectures and Tools for Rapid Prototyping. Springer Verlag, Berlin, 1992.
- [2] R. W. Hartenstein, M. Z. Servit: Field-Programmable Logic. Architectures, Synthesis and Applications. Springer Verlag, Berlin, 1994.
- [3] R. W. Hartenstein: Field-Programmable Logic. Smart Applications, New Paradigma and Compilers. Springer Verlag, Berlin 1996.
- [4] R. W. Hartenstein, J. Becker: A Two-level Co-Design Framework for Xputer-based data-driven reconfigurable Accelerators. IEEE 1997 HICCS-30, Wailea, Maui, Hawaii, Januar 1997.
- [5] R. W. Hartenstein: Standort Deutschland: Wozu noch Mikro-Chips? IT Press, Bruchsal, Chicago, 1994.
- [6] R. W. Hartenstein: Null Bock auf High Tech. IT Press, Bruchsal, Chicago, 1996.
- [7] K. Schlüter: Xputer – Innovation für den Multimedia-Markt. Funkschau 5/92, S. 76 ff.

Listing 1. Der Weichzeichner als q-X-C-Programm

```

1 /* Dateiname: Weichzeichner.c
2 Dieses Programm filtert Rauschen aus einem
3 Bild aus. Es liest das Bild ein und gibt es
4 geglättet wieder aus. */
5
6 #include <stdio.h>
7 #include <stdlib.h>
8 #include <string.h>
9 #include „gif-gray-load.h“
10 #include „gifsave.h“
11 static unsigned char global_color_map_out
12 ↪ = 0;
13 static unsigned char print_image = 0;
14 int image[640][480];
15 /*-----*/
16 void FIR()
17 {
18     int x,y;
19     int i,j;
20     int sum;
21     sum = 0;
22     for(x=0; x<640-2; x++)
23     { for(y=0; y<480-2; y++)
24     {
25         for( i=0; i<2; i++)
26         for( j=0; j<2; j++)
27             { sum += image[x+i][y+j]; }
28         image[x][y] = sum / 3;
29     }
30     }
31 }
32 /*-----*/
33 int main(int argc, char * argv[])
34 {
35     FILE * fp;
36     int i, t;
37     if((fp = fopen(argv[1],“r”) == NULL)
38     {
39         printf(„# error # cannot open %s\n“,
40 ↪ argv[1]);
41         exit(1);
42     }
43     GIFRead_gray(argv[1], image);
44     FIR();
45     GIF_Create(„Weichzeichner.gif“,640,480,
46 ↪ 2,1);
47     GIF_Close();
48     return 1;
49 }
    
```

Listing 2. aC-Code des Weichzeichners

```

1 int image[640][480];
2 int x,y;
3 main()
4 { for(x=0; x<640-2; x = x + 1)
5   { for(y=0; y<480-2; y = y + 1)
6     {
7         image[x][y] = ( image[x+0][y+0] +
8             image[x+0][y+1] +
9             image[x+1][y+0] +
10            image[x+1][y+1] ) / 3;
11     }
12 }
13 }
    
```

Listing 3. hC-Code des Weichzeichners

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #include „gif-gray-load.h“
5 #include „gifsave.h“
6 static unsigned char global_color_map_out
7 ↪ = 0;
8 static unsigned char print_image = 0;
9 int image[640][480];
10 int main(int argc, char * argv[])
11 {
12     FILE * fp;
13     int i, t;
14     if((fp = fopen(argv[1],“r”) == NULL)
15     {
16         printf(„# error # cannot open %s\n“,
17 argv[1]);
18         exit(1);
19     }
20     GIFRead_gray(argv[1], image);
21     FIR();
22     GIF_Create(„Weichzeichner.gif“,640,480,
23 ↪ 2,1);
24     GIF_Close();
25     return 1;
26 }
    
```

Listing 4. MoPL-Code des Weichzeichners

```

1 /* MoPL Programm des ausgefilterten aC-
2 Programmsegments */
    
```

```

3 mopl weichzeichner (ref int image
4 ↪ [640][480]);
5 /* Deklaration der erforderlichen
6 2-dimensionalen Datenbereiche */
7 array image [1:640, 1:480] of int;
8 /* Deklaration der erforderlichen 2 Basis-
9 Scan-Patterns, welche im Hauptprogramm zu
10 einem Video Scan ueber das gesamte Bild
11 verschachtelt werden */
12 scanpattern
13 /* Scan Pattern ueber eine Reihe */
14 reihenscan is 638 steps [1,0];
15 /* Scan Pattern ueber eine Spalte */
16 spaltenscan is 478 steps [0,1];
17 /* Deklaration des erforderlichen
18 Scan Windows der Groesse 2x2 */
19 window weichzeichnerW is
20 image_window [1:2, 1:2] of int;
21 /* Deklaration der erforderlichen rekonfigu-
22 rierbaren ALU zur Manipulation
23 (Weichzeichnen) der Bilddaten */
24 rALU subnet weichzeichner of weichzeichnerW
25 ↪ is
26 image_window[1,1] = (image_window[1,1] +
27 image_window[1,2] +
28 image_window[2,1] +
29 image_window[2,2]) / 3;
30 /* Beginn Hauptprogramm */
31 begin
32 /* Spezifikation der zu benutzenden Scan
33 Window Gruppe */
34 with weichzeichnerW do
35     /* Beginn des Aktivierungs- und
36     Ausfuehrungsteils */
37     begin
38         /* Aktivieren der entsprechenden
39         rekonfigurierbaren ALU */
40         activate weichzeichner;
41         /* Bewegen des Scan Windows in
42         Initialposition [1,1] */
43         move image_window to image [1,1];
44         /* Beginn des Video Scan Patterns, das
45         durch Verschachteln der beiden oben
46         deklarierten Basis-Scan-Patterns
47         entsteht */
48         spaltenscan (reihenscan [image_win
49 ↪ dow];) [image_window];
50     /* Ende des Aktivierungs- und
51     Ausfuehrungsteils */
52     end;
53 /* Ende Hauptprogramm */
54 end;
    
```

„Made in Germany – da ist der Wurm drin!“

Dr.-Ing. Reiner Hartenstein ist Professor am Fachbereich für Informatik der Universität Kaiserslautern. Dort leitet er eine Forschergruppe für Rechnerstrukturen und Mikrochip-Entwurf. Im Laufe seines Lebens hat er ein Dutzend Bücher geschrieben und mehr als 300 Fachaufsätze veröffentlicht. Seine Ausbildung erfuhr er als Mitarbeiter von Karl Steinbuch an der Universität Karlsruhe.

Hartenstein war 1981 Gastprofessor am Electrical Engineering and Computer Science Department der University of California in Berkeley. Dort zog ihn die von Carver Mead losgetretene Massenbewegung „Professoren auf die Schulbank“ in ihren Bann. Sie wollte das Chipdesign auch an den Hochschulen etablieren.

Zurück in Deutschland, versuchte Hartenstein, die Mikroelektronik auch in den Lehrplänen hiesiger Hochschulen zu verankern. Beim Bundesministerium für Forschung und Technologie (BMFT) stieß er



„Das ist die freiwillige Selbstentmannung des deutschen Michels, stillschweigend abgesegnet durch ein Parlament von High-Tech-Eunuchen, der Schildbürgerstreich des Jahrhunderts!“

dabei zunächst auf wenig Gegenliebe. Seine Bemühungen gipfelten – wie er selbst gerne betont – zunächst in einem „Rausschmiß erster Klasse“ durch den zuständigen Referatsleiter im BMFT und dessen Stammgutachter. Dennoch gelang es ihm 1983, auch die deutschen Professoren „auf die Schulbank“ zurückzubringen. Das BMFT unter Leitung von Dr. Heinz Riesenhuber bewilligte 35 Millionen Mark und später weitere 15 Millionen Mark für das Multi-University-E.I.S. Project (E.I.S. steht für „Entwurf Integrierter Schaltungen“), dem nationalen Vorläufer des „Eurochip Project“, heute „Europractice“ genannt.

Hartenstein ist der Fachwelt bekannt durch seine Beiträge zum Computer Aided Design (CAD) for Very Large Scale Integration (VLSI) in Telecommunications Project (CVT) und zum CAD for VLSI Systems Project (CVS), zwei ESPRIT-Programme der Europäischen Gemein-

schaft. Lange vor VHDL und Verilog entwickelte er die Hardware-Beschreibungssprache KARL-3, die in den 80er Jahren auch international erfolgreich war. Seit Ende der 80er Jahre arbeitet Hartenstein mit seinem Team an einem neuen Rechner-Paradigma, dem Xputer. Mit Professor Hartenstein sprach DOS-Redakteur Dr. Klaus Schlüter.

mc-extra: Ihre beiden Bücher „Standort Deutschland: Wozu noch Mikro-Chips?“ und „Null Bock auf High Tech“ haben beide einen schwarzen Einband. Sind Sie ein Schwarzseher?

Hartenstein: Den Schwarzseher schmerzt es am meisten, wenn er recht behält – so wie mein Lehrer Karl Steinbuch mit seinen Bestsellern. «Wenn wir nicht umkehren auf dem Marsch in eine Gesellschaft, wo High-Tech-Unwissen in ist», so warnte er, «dann müssen wir im Jahr 2000 den Gürtel enger schnallen.» Seine Prognose ist, wie wir heute wissen, fünf Jahre früher

„Made in Germany – da ist der Wurm drin!“ (Fortsetzung)

eingetreten. Aber noch immer therapieren wir nur die Symptome. Und vielleicht begreifen wir ja erst dann die tieferen Ursachen der steigenden Arbeitslosigkeit, wenn das zu einer Umkehr nötige Geld vollends fehlt.

mc-extra: Sie werfen den Deutschen Technikfeindlichkeit und ein rückständiges Bewußtsein vor. Sie zitieren Heinz Erhard: «Made in Germany - da ist der Wurm drin». Worin sehen Sie die Ursachen für den Niedergang der deutschen Wirtschaft?

Hartenstein: Unser Weg in die Krise begann mit der Jobkiller-Diskussion. Dabei ist der Mikroprozessor gar nicht der Jobkiller, als der er gemeinhin bezeichnet wird. Stephen Roache, einer der führenden Volkswirtschaftler der US-amerikanischen Investment-Bank Morgan Stanley, bezeichnete das Ergebnis einer über 15 Jahre laufenden Studie über Computeranwendungen in der US-amerikanischen Industrie als Productivity Paradox: Es stellte sich heraus, daß selbst durch massive Investitionen in Computer – die Firmen gaben im Schnitt ein Viertel ihres Umsatzes für Software, Hardware und Support aus – keine spürbaren Produktivitätssteigerungen erzielt werden konnten. Paradox ist – so empfinde ich es jedenfalls –, daß wir die hierzulande so verhaßten Mikrochips massenhaft importieren. Sie sind in vielen Konsumartikeln eingebaut: im Auto, im Fernseher und im Fotoapparat ebenso wie in der Stereoanlage. Wir haben viele Arbeitsplätze wegdiskutiert, die eine hohe Qualifikation voraussetzen. Und das, obwohl wir Deutschen – Weltmeister in Sachen Komplexität – die besten Chip-Designer hätten werden können.

mc-extra: Vor über zwei Jahren haben Sie die Pleite von Grundig vorausgesagt. Ihr Argument: Eine Unterhaltungselektronik-Firma sei langfristig nicht haltbar, wenn sie keine hundertprozentige Mikroelektronik-Kompetenz besitze. Was ist zu tun, um den Verlust weiterer Arbeitsplätze abzuwenden?

Hartenstein: Der Motor der Wirtschaft hat keine PS, sondern MIPS. Er heißt Mikrochip und veredelt nahezu alle Produkte. Wir müssen diese enorme Wertschöpfung wieder verstärkt in Eigenregie erbringen, statt sie überwiegend und teuer zu importieren. Das setzt unter anderem Steuerentlastungen für alle in dieser Branche erzielten Einkünfte voraus. Ganz wichtig: Wir müssen die Unternehmen

wieder dazu ermuntern, in Deutschland zu investieren. Die Globalisierung ist kein Unheil, sondern eine Chance.

mc-extra: Sie reden der Globalisierung das Wort? Worin sehen Sie deren Chancen?



„Wir haben viele Arbeitsplätze wegdiskutiert, die eine hohe Qualifikation voraussetzen. Und das, obwohl wir Deutschen – Weltmeister in Sachen Komplexität – die besten Chip-Designer hätten werden können.“

Hartenstein: Daß in Folge der Globalisierung Arbeitsplätze hier vernichtet und anderswo eingerichtet werden, ist nur ein Teil der Wahrheit. Tatsächlich steigen die Löhne in den Billiglöhnländern schneller als bei uns. Langfristig werden die Gehälter daher unser Niveau erreichen. Die neu gewonnene Kaufkraft macht die vermeintlichen Entwicklungsländer zu attraktiven Absatzmärkten

der Zukunft. Damit wir jedoch nicht weg vom Fenster sind, bis es soweit ist, dürfen wir die Hände nicht in den Schoß legen. Statt die Altindustrien zu subventionieren, müssen wir die Zukunftsindustrien steuerlich begünstigen – und zwar subito!

mc-extra: Im Jahre 1994 soll das BMFT die Streichung aller Fördergelder für Hardware-Projekte beschlossen haben. Was ist an der Behauptung?

Hartenstein: Das stimmt. Das ist die freiwillige Selbstenstimmung des deutschen Michels, stillschweigend abgesegnet durch ein Parlament von High-Tech-Eunuchen, der Schildbürgerstreich des Jahrhunderts!

mc-extra: Hoppla! Was ist denn passiert?

Hartenstein: 1994 beschloß das BMFT klammheimlich, die Hardware-Förderung zu beenden – nur «gut unterrichtete Kreise» hatten davon Kenntnis. Frei nach dem Motto: «Wir Deutschen sind schlauer als die anderen, wir machen nur die Blaupausen (die Software).» Einige Professoren, darunter auch ich, formulierten daraufhin eine Protest-Resolution – auf einem Treffen in Dagstuhl, um genau zu sein. Später hat das BMFT dann alles dementiert – klammheimlich, versteht sich. Omerta heißt, glaube ich, so ein Vorgehen bei der Mafia: Jeder, der etwas gesehen hat, schweigt, um sich selbst zu schützen.

mc-extra: Was meinen Sie, wenn Sie von Kommunikationsfilz reden, von Motivation statt Hierarchie?

Hartenstein: High-Tech „per ordre di muf-ti“, mit High-Tech-Laien an der Spitze – das funktioniert nicht. Das hat doch Edzard Reuter, der ehemalige Chef von Daimler-Benz, vorgeführt. Das wohl bekannteste Beispiel in der Geschichte des Computers ist aber wohl die Firma Fairchild Semiconductor im Silicon Valley. In den 60er Jahren entwickelte die Firma die IC-Fertigungstechnik. Da die Muttergesellschaft Fairchild Cameras and Instruments ihren Hauptsitz an der amerikanischen Ostküste hatte, war eine Hierarchisierung der Geschäftsabläufe nicht zu vermeiden. Bewährte Mitarbeiter fühlten sich durch zunehmende Gängelung dazu ermuntert, eigene Firmen zu gründen: darunter waren Andy Grove, Gordon Moore und Robert Noyce zum Beispiel, die Väter von Intel.

Wir brauchen keine neuen Hierarchien – wir brauchen gut funktionierende Teams motivierter Mitarbeiter. Die Kommunikation muß stimmen, der Gedankenaustausch fließen. Die Probleme, vor denen wir heute stehen, sind komplex. Sie setzen die Fähigkeit zu vernetztem Denken voraus. Dem PC als Telekommunikationsmedium kommt da eine zentrale Rolle zu. Er bringt alle an einen Tisch, die zur Lösung eines Problems beitragen können. Manager, die es gern militärisch haben, empfinden es mitunter als Statusverlust, wenn sie ihre Mitarbeiter am Bildschirm nicht exerzieren lassen können.

mc-extra: Auf der Jahrespressekonferenz von Compaq im Frühjahr in München gaben die Firmenbosse Eckhard Pfeifer, Kurt Dobitsch und Andreas Barth zu, daß sie nur noch die «horizontale», aber nicht mehr die «vertikale» Entwicklung forcieren wollen. Mit anderen Worten: Das Geld wird in vorhandene Technologien und nicht in neue investiert, die Fertigung wird rationalisiert und – so sehe ich das jedenfalls – Arbeitsplätze gehen verloren. Gilt die Wachstumsmaxime eigentlich nur noch für den Produkti-

onsfaktor Kapital: Hauptsache, die Geldmenge wächst, egal auf wessen Kosten?

Hartenstein: Ich kenne die drei Herren nicht persönlich. Fast die gesamte PC-Branche entwickelt so gut wie nichts mehr selbst, schon gar keine Mikrochips. Es gibt Ausnahmen: Siemens Nixdorf beispielsweise. Allenfalls die Boards werden noch in Eigenregie gefertigt. Ich glaube, daß der Begriff «Schraubenzieher-Firmen» die Sache ganz gut trifft. Langfristig wird sich dieses Vorgehen jedoch als kurzsichtig erweisen. Das horizontale Denken mag zwar im einen oder anderen Gewerbe lukrative Gewinne abwerfen, in der High-Tech-Branche ist jedoch nach wie vor die Vertikale gefragt.

mc-extra: Welchen Stellenwert hat der Produktionsfaktor Bildung in unserer Gesellschaft?

Hartenstein: Der wichtigste Produktionsfaktor Deutschlands ist die Qualifikation der Bürger. Sie fängt in der Schule an, bei den Leminhalten. Mikroelektronik, Informatik und Technik überhaupt sollten Priorität bekommen – die Länder in Fernost machen das vor. Aber schauen Sie mal, was hierzulande getan wird. Statt im Geschichtsunterricht auf die Historie von Innovationen einzugehen, lernen die Kinder immer noch die des Gallischen Krieges ...

mc-extra: ... und 50 lateinische Vokabeln für den Begriff „Krieg führen“: bellum facere, pugnare, ... ! Wenn Sie von Fernost sprechen, meinen Sie sicher Korea und Taiwan. In Japan hat Kultur einen anderen Stellenwert.

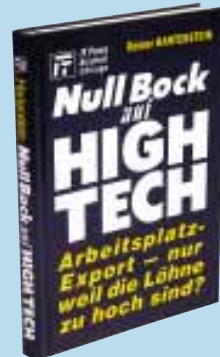
Hartenstein: Das stimmt. Dennoch, was wir brauchen, ist ein gemeinsamer Nenner in der Gesellschaft, ein gemeinsames Ziel. Wir müssen den Begriff High-Tech mit neuem Leben füllen.

mc-extra: In Ihrem Buch „Null Bock auf High Tech“ fordern Sie einen neuen Nationalismus. Ein Radikaler sind Sie doch allenfalls in Ihrer Wissenschaft, oder?

Hartenstein: Der Mensch braucht Ziele, eine Orientierung – vor Führern und Rätenfängern sollten wir uns allerdings hüten. Unser nationales Ziel muß die Schaffung qualifizierter Arbeitsplätze sein. Wir brauchen eine Infrastruktur, die High-Tech-Begabungen zu Höchstleistungen animiert. Wir brauchen keine neuen Hierarchien, sondern begeisterte Team-Arbeiter. Der Wettbewerb – Krieg, wenn Sie so wollen – geht heute nicht mehr um Territorien, sondern um Märkte. Und technische Bildung ist die wichtigste Waffe in dieser Auseinandersetzung.



„Unser nationales Ziel muß die Schaffung qualifizierter Arbeitsplätze sein.“



„Unser Abstieg ist nur zu bremsen, wenn wir das Wesen des weltweiten Umbruchs der Wirtschaft begreifen.“