

Prof. Dr.-Ing. Reiner Hartenstein

Der Mikroprozessor im nächsten Jahrtausend

Als diskreter Baustein hat er keine Zukunft mehr

Nach 30 Jahren Weiterentwicklung neigt sich die Ära des Mikroprozessors dem Ende zu: Schon heute nimmt er im Vergleich zu den Cache-Speichern nur noch einen Bruchteil der Chipfläche ein. Die zunehmende Spezialisierung für bestimmte Anwendungen führt zu einer unübersichtlichen Ausweitung der Architekturen und beseitigt dennoch nicht den systembedingten Nachteil des Mikroprozessors: sein unveränderlicher Instructionssatz. Die Zukunft gehört der rekonfigurierbaren Hardware.

Mainframes sind die Dampflokomotiven der Digitaltechnik. Obwohl solche Museums-Eisenbahnen auch heute noch in Betrieb sind, ziehe ich den Begriff „Computer“ absichtlich etwas enger: Ich befaße mich hier nur mit Mikroprozessoren. Nicht der Mainframe, sondern der Mikroprozessor ist die größte kulturelle Errungenschaft der Menschheit seit Erfindung der Schriftzeichen. Nur als Mikroprozessor wird der Computer allgegenwärtig als Basis des Informations- und Kommunikations-Zeitalters.

Die Tatsache, daß der Mikroprozessor von allen Halbleiterprodukten die größte Aufmerksamkeit in der Öffentlichkeit genießt, verschleiert, daß die Silizium-Industrie ungeheuer vielschichtig ist bezüglich Infrastrukturen und Methoden. Stets existiert eine Vielfalt veralteter, aktueller und emporkommender Technologien nebeneinander. Makimotos Wellen-Modell in *Bild 1* faßt Geschichte und Zukunft dieser Hauptentwicklungslinien in der Silizium-Anwen-

dung zusammen [1]. Makimoto unterscheidet, ähnlich wie Kondratieff-Zyklen, drei Wellen der langfristigen Entwicklung, die ich nennen möchte: die Steinzeit des Siliziums, die Bronzezeit, die wir jetzt noch erleben, und die Zeit der rekonfigurierbaren Hardware, die bereits begonnen hat.

Nick Tredennick faßt die Paradigmenwechsel der Hardware-Plattformen auf seine Art zusammen [2, 3]. Erste historische Computer wie Digital Differential Analyzer und andere waren anwendungsspezifisch oder auf ein bestimmtes Anwendungsgebiet festgelegt: Algorithmen und Ressourcen waren unveränderlich. Diese Steinzeit wird dominiert durch festverdrahtete Plattformen und beendet durch die erste Entwurfs-Krise – verursacht durch eine explodierende Typenvielfalt immer komplexer werdender ICs. Die Bezeichnung „Steinzeit“ erinnert an Lithographie für ICs, insbesondere an die maßhaltigen „rubylith“-Filme, aus denen vor Jahrzehnten das Layout ausgeschnitten und mit einer Reprokamera als Maske abfotografiert wurde. Digitalisierte Weiterentwicklungen solcher „steinzeitlicher“ Methoden sind übrigens noch heute eine Basis der Hardware-Synthese. Doch im Rahmen dieses Aufsatzes ist für weitere Details hierzu leider kein Raum.

Der Mikroprozessor als Kulturrevolution

In den siebziger Jahren begann mit der Einführung des sog. Von-Neumann-Paradigmas eine neue Epoche. Sie erlaubt variable Algorithmen bei unveränderlichen Ressourcen. Derzeit beobachten wir noch diese, nun niedergehende Bronzezeit, gekennzeichnet durch die Dominanz prozeduraler Denkweisen, die um 1977 begann und auf der Einführung des Mikrocontrollers beruht. Die Zeitdifferenz zur Erfindung des Mikroprozessors um 1969 deutet die Diffusionszeit bis zum Durchbruch an. Damit das Produkt Ab-



Zeitalter*	Steinzeit	Bronzezeit	Configware-Zeit
Kommentar	explodierende Typenvielfalt an ICs: am Ende die erste Entwurfs-Krise	Vielfalt an Akzeleratoren: Symptome der Grenzen des Mikroprozessors am Ende die zweite Entwurfs-Krise	programmierbare Akzeleratoren: „grobkörnig“, da FPGAs zu ineffizient am Ende als Retter aus der Technologie-Krise
Makimotos Wellen-Modell	standardisiert Standard Transistoren, Nand, Nor... 1957 (TTL) spezialisiert	Speicher, Mikro-Prozessor 1977 prozedurale Programmierung	FPGAs „feinkörnig“ rekonfigurierbar (universell**) 1997 strukturelle Programmierung **)aber ineffizient
Tredennicks Paradigmen	fest verdrahtet (wie auch Prä-von-Neumann-Computer): Paradigma: Algorithmen:fest Ressourcen:fest	sogen... „von Neumann“-Computer und Mikroprozessoren: Paradigmen-Wechsel zu: Algorithmen: variabel Ressourcen:fest	rekonfigurierbare Schaltungen & Systeme: Paradigmen-Wechsel zu: Algorithmen:variabel Ressourcen:variabel

Bild 1. Makimotos Welle: Geschichte und Prognose der bestimmenden Silizium-Anwendungen. Nach dem Umschwung zu den variablen Algorithmen in den siebziger Jahren steht nun der Wechsel zu den variablen Ressourcen an.

diumsdiskussionen über die Entwurfskrise. Ein Großteil des per Technologiefortschritt gegebenen Überflusses an Chipfläche wird für immer größere Zwischenspeicher verwendet, um den Kommunikations-Engpaß zwischen Prozessor und Speicher zu überbrücken. Dadurch erreichte Verbesserungen sind jedoch von der Art des Anwendungsprogrammes abhängig und werden sicher oft überschätzt. Neuerdings werden Chips mit mehreren Mikroprozessoren entwickelt – eher aus Verlegenheit, wie es scheint. Der Mikroprozessor als physischer Baustein wird immer fragwürdiger angesichts des aufkommenden „System on Chip“ (SoC). Bezüglich der Produktionsstückzahl wurde der Mikroprozessor als diskreter Baustein vom eingebetteten Mikroprozessor inzwischen mehrfach über-

nehmer fand, mußte Intel für 250 000 potentielle Kunden Kurse halten; denn die Idee, den LötKolben gegen die Tastatur einzutauschen, empfanden Elektroniker damals als hochgradig beschuert: Berührungängste gegenüber einer neuen Kultur, ein Bildungsproblem. Dies ist verständlich auch angesichts der ersten Design-Krise, da CAD weder gelehrt wurde noch als Industriezweig existierte.

Die Verdienste der Einführung des Mikroprozessors sind unbestritten, weshalb hier nicht näher darauf eingegangen sei. Erwähnt sei aber kurz der Aufstieg der Automatisierungstechnik und die Einführung des Desktop-Computers (Apple und Vorgänger), womit das Computer-Zeitalter begann – vor einem Vierteljahrhundert. Die Bundesregierung begann gerade, die Förderung der Computerforschung zurückzufahren auf heute praktisch Null. Das von der Mainframe-Zeit übernommene Maschinen-Paradigma (von Neumann) ist dabei die entscheidende Basis der Methoden-Fokussierung bei der erfolgreichen Weiterentwicklung der Programmiersprachen und Compiler-Techniken – auch eine essentielle Basis der Weiterentwicklung des Software-Engineering und der Entstehung einer Hunderte von Milliarden schweren Software-Industrie.

Die Gordon-Moore-Kurve, die das Wachstum der Integrationsdichte (Transistoren je Chip) angibt, gilt nur für Speicherbausteine, jedoch nicht für den Mikroprozessor als Gesamt-Mikrochip. Der Rückstand hat inzwischen einen Faktor von fast 100 erreicht (Bild 2). Eine Ursache für diese Lücke ist der Stand der Entwurfsmethoden gegenüber dem Mikroprozessor. Speicherbausteine sind viel einfacher und regelmäßiger strukturiert aufgebaut. Auf Konferenzen gibt es wieder Po-

Der Niedergang des Mikroprozessors als Baustein

Seit einiger Zeit zeigt sich ein Preisverfall bei PCs und – in den USA – darüber hinaus neuerdings ein Rückgang der jährlich verkauften Stückzahlen. Eine wachsende Konkurrenz zum PC ist die zunehmende Vielfalt von meist portablen oder sogar batteriegetriebenen Endgeräten mit eingebettetem Mikroprozessor, wie intelligente Handys, mobile

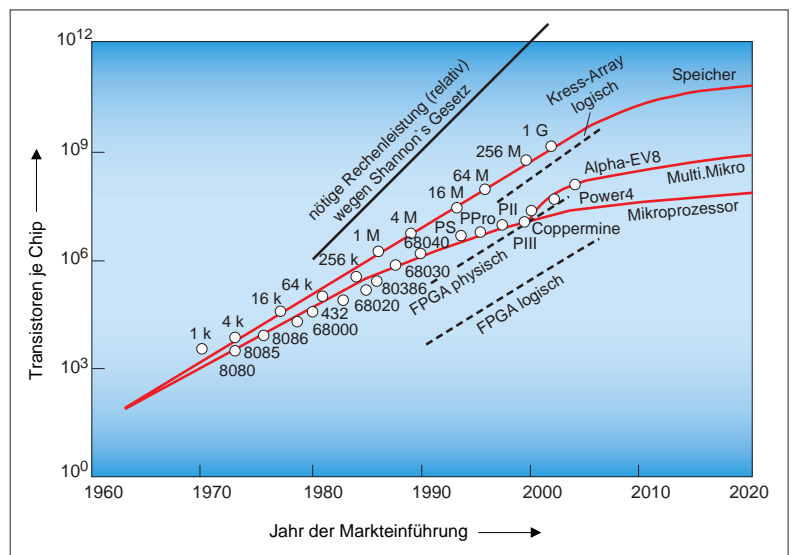


Bild 2. Die Entwicklung der Integrationsdichte bei Speicher, Mikroprozessor und rekonfigurierbaren Bausteinen. Das Moore'sche Gesetz gilt in seiner vollen Konsequenz nur noch für Speicher. Die Mikroprozessoren haben wegen ihres komplizierten Aufbaus bereits einen Rückstand fast um den Faktor 100 gegenüber Speichern erreicht.

Bildtelefone (derzeit nur Japan), Surfpads, Organizer, Spielekonsolen und anderer Unterhaltungselektronik. Die Verkaufszahlen gehen steil in die Höhe und die Prognosen sind glänzend. Allein bei Handys rechnet man bis etwa zum Jahre 2003 mit einem aktiven Bestand von einer Milliarde Stück. Der PC als mikroprozessorgesteuertes Gerät kommt allmählich, aber eindeutig auf die Verliererseite.

Das Wettrennen zwischen Intel und seinen Konkurrenten bringt zwar ständig immer mehr Megahertz. Aber die Durchsatz-Anforderungen wachsen schneller als die MIPS, wie etwa für mehr Auflösungsvermögen bei der Verarbeitung und Präsentation von Bildern und Videos, von einst Briefmarkengröße auf schon zwei Millionen Pixel und bald mehr, um nur eine aus einer Vielzahl von Anwendungen zu nennen. Mikroprozessoren sind des öfteren nicht einmal in der Lage, ihren eigenen Bildschirm schnell genug zu bedienen.

Die schon lange nötigen Beschleuniger werden nicht überflüssig, nein, im Gegenteil: Der Anteil des Mikroprozessorkerns an der Gesamt-Siliziumoberfläche digitaler Systeme wird immer kleiner. Zusatzlogik zur Beschleunigung nimmt den Löwenanteil der Fläche in Anspruch – mit zunehmender Tendenz. Daran wird sich auch dann nichts ändern, wenn in nicht mehr ferner Zukunft das gesamte Digitalsystem eines Produkts auf einem einzigen Chip untergebracht wird: das ganze „System on Chip“ (SoC).

Die Halbleiterindustrie ist seit der Erfindung des Transistors einem ständigen Wandel unterworfen [5]. Die aktuellen Trends zeigen, daß der Mikroprozessor als Einzelbaustein mehr und mehr zurückfällt. Statt dessen wird er zu einem kleinen Bestandteil innerhalb eines umgebenden Systems auf dem Chip, zu einem „IP-Core“. IP steht für „intellectual property“. Selbst komplexe CISC-Mikroprozessor-Kerne nehmen schon jetzt mit kaum noch 2 mm² nur noch einen Bruchteil einer Mikrochip-Fläche ein.

Selbst mit immer größeren Caches wird die Chipfläche nicht mehr voll. Schon stehen Chips mit 2 oder 4 Prozessoren vor der Markteinführung, wie beispielsweise von IBM, Hewlett-Packard und anderen. Doch damit stehen die vielen ungelösten Parallelisierungs-Probleme aus dem Nachlaß der klassischen Parallelrechner-Szenen vor einer Neuauflage.

Die Anbieter von Mikroprozessoren als physische Bausteine haben ihren ständigen Kampf um die richtige Balance zwischen Höchstleistung und Universalität eigentlich schon verloren – nicht nur wegen der Entwurfs-lücke. Die Universalität ging mehr und mehr verloren. Mit den Von-Neumann-Maschinen-Prinzipien aus den 40er Jahren und nach mehr als zehn Technologie-Generationen ist selbst der erst 30 Jahre später eingeführte Mikroprozessor ohnehin schon ein Methusalem.

Die Technologie-Lücke

Die notwendige Vielzahl an Beschleunigern als Prothesen für den altersschwach gewordenen Mikroprozessor ist nur eine unbefriedigende Lösung. Mit steigenden Durchsatzanforderungen und steil steigenden Bandbreiten von Kommunikationsnetzen werden Beschleuniger immer aufwendiger, die Entwurfskosten steigen steil an, während der Produkt-Lebenszyklus immer kürzer wird: eine zweite Design-Krise am Ende der „Bronzezeit“, die an die erste Design-Krise zu Ende der „Steinzeit“ der Silizium-Anwendung erinnert.

Die beklagenswerte Typenvielfalt ergibt sich aus der Spezialisierung der Beschleuniger. Mehr Flexibilität ist gefragt, um größere Stückzahlen und einen längeren Produkt-Lebenszyklus zu erreichen. Zwischen dem hochflexibel einsetzbaren aber zu langsamen Mikroprozessor und dem hochleistungsfähigen, aber inflexiblen Beschleuniger klafft eine Technologie-Lücke (Bild 3). Gesucht wird eine neue Technik, die bei hoher Flexibilität einen größeren Durchsatz als der Mikroprozessor bietet. Eine solche neue Technik sind Rechnersysteme, die während des Betriebs ihre eigenen Hardware-Schaltkreise selbst modifizieren. Sie öffnen ein neues Gebiet der Entwicklung von Digital-Systemen: das Gebiet der dynamisch rekonfigurierbaren Schaltungen und ihrer Anwendungen. Schon bewähren sie sich in Nischenmärkten wie Zeichen- und Muster-Erkennung, Bildverarbeitung und Verschlüsselung und einigen anderen Gebieten.

Obwohl von Gerald Estrin schon 1960 an der UCLA vorgeschlagen, ist „Reconfigurable Computing“ noch ein junges Gebiet, das aber einen raschen Aufschwung erlebt. Dies beweisen nicht zuletzt auch wachsende Teilnehmerzahlen entsprechender Konferenzen zum Thema programmierbarer Logikbausteine. Tsugio Makimoto hat recht gehabt mit seiner mehr als ein halbes Jahrzehnt alten Prognose [1]. Die von Makimoto vorhergesagte dritte Phase der Silizium-Anwendung (siehe Bild 1), die Zeit der rekonfigurierbaren Systeme, bringt dynamisch rekonfigurierbare Plattformen in den Brennpunkt des Interesses: Algorithmen und Ressourcen sind variabel [2, 3].

Mit dem Terminus „reconfigurable“ (rekonfigurierbar) wird strukturelle Programmierbarkeit digitaler Hardware in Gegensatz gestellt zur prozeduralen Programmierung, dem heute allgegenwärtigen und dominanten Paradigma der zweiten Hälfte des zu Ende gegangenen Jahrhunderts. Meist FPGAs genannt (field-programmable gate arrays), besetzen strukturell programmierbare integrierte Schaltungen schon seit mehr als einem Jahrzehnt eine ansehnliche Drei-Milliarden-Nische (US-Dollar weltweit; Stand 1999), mit interessanten Wachstumsraten. Sogar auflagenstarke Medien haben inzwischen die Rekonfigurierbarkeit entdeckt.

Der Markt für FPGAs lebt nicht nur von Forschung und Lehre, vom ASIC-Ersatz für Produkte mit geringen Stückzahlen wo das „Programmieren“ von FPGAs billiger ist als die Fabrikation maßgeschneiderter ICs, und für die Be-

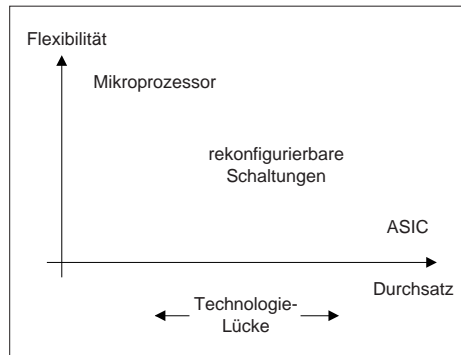


Bild 3. Die Algorithmen von Mikroprozessoren sind flexibel, ihr Durchsatz aber mager. ASICs rechnen schnell – aber nur die ihnen „eingebrennte“ Aufgabe. Dazwischen klafft eine Lücke, die rekonfigurierbare Schaltungen füllen: Sie sind schnell und können sich gleichzeitig an unterschiedliche Aufgaben anpassen.

stückung von ASIC-Emulatoren (für Rapid Prototyping: Verifikation komplexer IC-Designs, da Simulation wegen astronomischer Rechenzeiten nicht praktikabel ist). Immer wichtiger wird auch die Anwendung rekonfigurierbarer Schaltungen als Beschleuniger. Dieses Forschungsgebiet erlebte in den vergangenen Jahren eine rasche Expansion. In vielen Anwendungsgebieten bewies rekonfigurierbare Hardware erhebliche Durchsatzgewinne gegenüber Lösungen auf der Grundlage von Mikroprozessoren [6]. Das

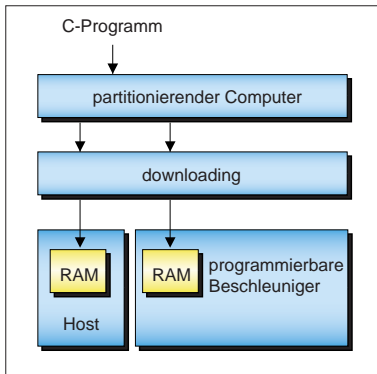


Bild 4. Wenn Mikroprozessor und rekonfigurierbarer Baustein sich in einem System befinden, kann ein Compiler die Lastverteilung übernehmen, indem er die Aufgaben zwischen Prozessor und programmierbarem Baustein aufteilt.

Spektrum der Anwendungen nimmt immer mehr zu. Mehrere internationale Serien von jährlichen Konferenzen wie die FPL [7] haben dieses Thema zum Gegenstand. Neuerdings gibt das US-Verteidigungsministerium an die Industrie und an Spitzenuniversitäten erhebliche Fördermittel für „Adaptable Computing“ aus. In Forschungsprojekten auf der Basis kommerziell verfügbarer FPGAs wurden schon vor vielen Jahren für eine Reihe von Algorithmen gegenüber vergleichbaren Software-Implementierungen erhebliche Speed-ups erreicht, teilweise sogar um Größenordnungen. Zu den meistzitierten Arbeiten zählen hier u.a. das PAM (Programmable Active Memory) [8]. Viele Beispiele entstammen dem Forschungs-Gebiet FCCM (FPGA-based Custom Computing Machines) [9, 10].

Was folgt nach der dritten Welle? Makimoto gibt darauf keine Antwort. Doch in Tredennicks Sichtweise steckt indirekt die Antwort. Nach variablen Algorithmen und variablen Ressourcen sind alle Kombinationen ausgeschöpft. Danach kann nichts Neues mehr kommen, zumindest nicht mit der bekannten Silizium-Technologie, die wohl ohnehin in 10 bis 20 Jahren ausgereizt ist. Bild 1 zeigt deshalb nach der dritten Welle keinen weiteren Nulldurchgang bei 2017. Es bleibt also relativ viel Zeit dafür, daß rekonfigurierbare Systeme die Rolle der Technologie als Innovations-Beschleuniger übernehmen kann.

Reconfigurable Computing

Die „feinkörnigen“ FPGAs sind jedoch nicht gut geeignet für berechnungsintensive Anwendungen, weshalb deren traditionelle Anwendung mehr die „glue logic“ ist. FPGA-Anwendungsentwicklung findet im Wesentlichen nur in der Ebene des Logik-Entwurfs statt. Wegen Rekonfigurierbarkeits-Overhead wird auf einem typischen FPGA oft nur etwa um 1 % der vorhandenen Transistoren für „aktive

Schaltungen“, d.h. für eigentliche Anwendung selbst benutzt [11]. Mit immer beliebter werdenden „grobkörnigen“ (coarse grain) rekonfigurierbaren Plattformen kann jedoch eine um Größenordnungen höhere Flächen-Effizienz erreicht werden als mit FPGAs (vgl. Bild 2).

Eine Anzahl Start-ups im Silicon Valley zielt auf grobkörnig rekonfigurierbare Beschleuniger bei Endgeräten und Basisstationen der drahtlosen Kommunikation, für Multimedia-verarbeitung, und für Multiprotokoll-Netzwerkrouter. Faszinierend ist bei ersteren die „Software Radio“-Philosophie: Beim Einschalten hat das Gerät noch keine Eigenschaften, wie ein Computer, der erst beim Booten aufwacht. Doch anstatt von der Festplatte wird drahtlos vom Netzwerk gebootet. In der drahtlosen Kommunikation wird durch Knappheit an Frequenzen die Kanaldecodierung immer aufwendiger. Wegen Shannons Gesetz ist der Anstieg des Bedarfes an Rechenleistung steiler als die Moore-Kurve (Bild 2): eine enorme Herausforderung für Low-Power-Design.

Grobkörnige Architekturen haben gegenüber FPGAs für Berechnungsaufgaben (computational tasks) eine Reihe von Vorteilen. Breite Datenpfade bieten viel mächtigere Operatoren (wie beim Kress-Array in rDPUs (reconfigurable Data Path Units) mit erheblich höheren Datenpfad-Breiten wie z.B. 4, 12, 16, 18 oder 32 bit) sowie eine drastisch höhere Flächen-Effizienz, und müssen nicht wie bei FPGAs aus nur ein Bit breiten CLBs (Configurable Logic Blocks) zusammengesetzt werden. Während für die FPGA-Synthese Hardware-Experten benötigt werden, lassen sich die „grobkörnigen“ durch Compiler konfigurieren. Allerdings sind dazu Compiler mit neuartigen Code-Generatoren

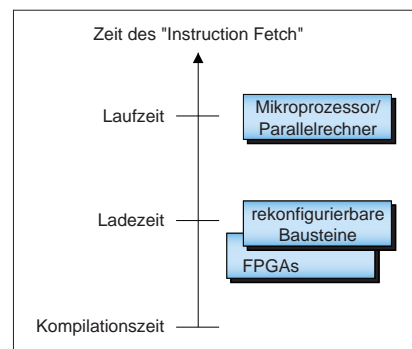


Bild 5. Programmierbare Bausteine verlangen neue Compiler. Z.B. verlagert sich der „instruction fetch“ von der Laufzeit in die Kompilations- oder Ladephase.

nötig. Bei Symbiose mit einem Mikroprozessor als „Host“ ist ein partitionierender Compiler sinnvoll, der die Lastverteilung optimiert (Bild 4).

Beschleuniger, die auf einem Maschinen-Paradigma beruhen, unterstützen Techniken des Compiler-Baus und damit höhere Programmiersprachen. Da hier die Rekonfiguration von Datenpfaden eine Verlagerung des „instruction fetch“ von der Laufzeit in die Kompilations- bzw. Ladezeit bedeutet (Bild 5), macht ein „instruction sequencer“ à la von Neumann keinen Sinn mehr. Der entsprechende Wechsel das Maschinen-Paradigmas hinüber zum „Data-Sequencing“ (Bild 6 b und c) ergibt ein allgemeines deterministisches Modell (Xputer) für rekonfigurierbare „Maschinen“, bei denen – im Gegensatz zu „Data Flow Machines“ – auch übliche Debugging-Techniken anwendbar sind. Eine erste Version eines Xputer-basierten partitio-

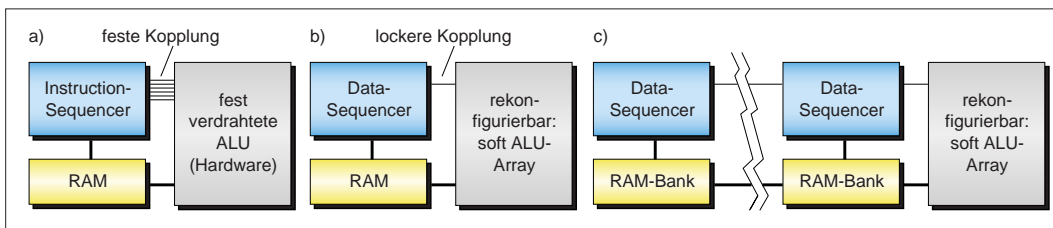


Bild 6. Ein „Instruction Sequencer“ wie bei Von-Neumann-Architekturen macht bei rekonfigurierbaren Systemen keinen Sinn mehr. An seine Stelle tritt ein „Data Sequencer“, auf den übrigens auch konventionelle Debugging-Techniken anwendbar sind.

nierenden Compilers wurde in Kaiserslautern im Rahmen des Co-De-X Framework implementiert. Zu diesem gehört quasi als Codegenerator ein Datenpfad-Synthese-System (DPSS), das aus C-Programmen Configware-Code für Kress-Arrays generiert (rDPU Arrays).

Der universelle rekonfigurierbare Beschleuniger ist eine Illusion. Realistisch sind auf Anwendungsgebiete optimierte r-Akzeleratoren. Deshalb wurde aus dem DPSS der hochgradig parametrisierte Kress-Array-Xplorer entwickelt [12], der ganze Familien von Kress-Array-Architekturen definiert. Durch Experimentieren mit alternativen Architekturen und deren Vergleich mittels Xplorer-eigenem Profiler (zur Vorausschätzung von Durchsatz, Flächenbedarf und Verlustleistung) kann jeweils für ein bestimmtes Anwendungsgebiet, wie beispielsweise Bildverarbeitung, eine optimierte Architektur ermittelt werden.

Für wiederholtes Abbilden einer Anwendung benötigt dabei der Xplorer für jede neue Architektur jeweils nur wenige Sekunden oder Minuten. Andere Syntheseumgebungen können über einen Verilog- und VHDL-Beschreibungsgenerator und einen Layoutgenerator des Xplorer genutzt werden.

Verzögerung durch Akzeptanzprobleme

Mit den Von-Neumann-Maschinen-Prinzipien aus den 40er Jahren und nach mehr als zehn Technologie-Generationen ist der erst 30 Jahre später eingeführte Mikroprozessor schon ein Methusalem. Der Anteil des eigentlichen Mikroprozessorkerns an der Gesamt-Siliziumfläche digitaler Systeme wird immer kleiner. Beschleuniger nehmen den Löwenanteil der Fläche in Anspruch. Die Universalität wird mehr und mehr beeinträchtigt durch die Durchsatzgrenzen des Mikroprozessors. Der Mikroprozessor verliert allmählich seine Kraft als ein Treiber hohen Innovations tempos. Aber auch die Siliziumtechnologie an sich wird vermutlich um die Jahre 2010 oder 2015 beginnen, an Dynamik als Innovationstreiber verlieren. Eine Aufrechterhaltung des Siliziumbooms darüber hinaus ist vermutlich nur durch bessere Designs und bessere Architekturen möglich.

Eine vielversprechende Basis hierzu sind die grobkörnigen dynamisch rekonfigurierbaren Plattformen. Deren gegenüber FPGAs um Größenordnungen höhere Flächen-Effizienz zusammen mit neuartigen Anwendungs-Entwicklungswerkzeugen haben das Potential, höhere Programmiersprachen als Eingabe zu akzeptieren, viele neue anspruchsvolle Anwendungsgebiete zu erschließen und damit die Dominanz des Mikroprozessors als das Mainstream-

Produkt anzugreifen. Eine Anzahl wohlfinanzierter Start-ups im Silicon Valley unterstreicht, daß die neue Kultur der Configware ein „hot topic“ geworden ist. Wie seinerzeit bei Einführung des Mikroprozessors als Kulturrevolution der Elektronik liegt jedoch auch hier ein Bildungsproblem vor, das den breiten

Durchbruch um Jahre verzögern wird. Der Mikroprozessor ist auch in Zukunft unverzichtbar, nämlich als der Schwanz, der mit dem Hund wedelt. jk

Literatur

- [1] *Manners, D.; Makimoto, T.:* Living With The Chip. Chapman & Hall, 1995.
- [2] *Tredennick, N.:* The Case for Reconfigurable Computing. Microprocessor Report, 10, 10 (5 Aug 1996).
- [3] *Tredennick, N.:* Technology and Business: Forces Driving Microprocessor Evolution. Proc. IEEE 83, 12 (Dec 1995).
- [4] *Hennessy, J.:* The Future Systems Research. Computer, August 1999.
- [5] *Hartenstein, R.:* Fundamentals of Structured Hardware Design. American Elsevier, 1977.
- [6] *Mangione-Smith, W. et al.:* Seeking Solutions in Configurable Computing. Computer, Dec. 1997.
- [7] *International Conf. On Field-programmable Logic (FPL):* <http://xputers.informatik.uni-kl.de/fpl/fpl.2000>
- [8] *Vuillemin, J. E. et al.:* Programmable Active Memories: Reconfigurable Systems Come of Age. IEEE Trans. on VLSI Systems, 4(1): 56 – 69, March, 1996. <ftp://pam.devinci.fr/pub/doc/To-Bepublished/PAMieee.ps.z>
- [9] *Hartenstein, R.:* Custom Computing Machines – an Overview. Proc. Workshop on Design Methodologies for Microelectronics (opening keynote), DMM '95, Smolenice Castle, Czech Republic, Sept. 1995.
- [10] *Buell, D. A. ; Arnold, J. M.; Kleinfelder, W. J.:* Splash 2: FPGAs in a Custom Computing Machine. IEEE CS Press, 1996.
- [11] *DeHon:* Reconfigurable Architectures for General Purpose Computing. report no. AITR 1586, MIT AI Lab, 1996.
- [12] *Hartenstein, R.; Herz, M.; Hoffmann, Th.; Nageldinger, U.:* Mapping Applications onto Reconfigurable KressArrays; Proc. 9th Int'l Workshop On Field-programmable Logic And Applications (FPL '99), 9 Glasgow, Scotland, Aug. 30 – Sept. 1, 1999, LNCS series, Springer-Verlag, Heidelberg, 1999.

Prof. Dr.-Ing. Reiner Hartenstein ist Professor am Fachbereich für Informatik der Universität Kaiserslautern und leitet dort den Lehrstuhl für Rechnerstrukturen. Er ist Program Chair mehrerer internationaler Konferenzen (Tschechien, Estland, Frankreich, Großbritannien, Österreich, Schweiz, U.S.A.). Seine derzeitigen Spezialgebiete sind Rechnerstrukturen und Entwurfsverfahren für Mikroelektronik sowie rekonfigurierbare Rechnerstrukturen und deren CAD-Werkzeuge und Compiler.

E-Mail: hartenst@rhrk.uni-kl.de
URL: <http://xputers.informatik.uni-kl.de/fpl/>

