

SYS³: was ist neu ?

"Systolischer Arrays"; hierbei handelt es sich um einen wichtigen Bereich der Wissenschaft vom Entwurf komplexer VLSI-Digitalschaltungen (in den USA gelegentlich auch "Design Sciences" genannt), der sich zur Zeit besonders rasch entwickelt. Eine rasch wachsende wissenschaftliche "Szene" mit einer zunehmenden Zahl hierauf spezialisierter internationaler und lokaler Konferenzen ist ein Anzeichen zunehmenden Interesses an diesem Gebiet. Die formalen Methoden und die allgemeine Systematik dieses noch recht jungen Gebietes ist - selbst im Vergleich mit anderen Zweigen der an sich schon sehr dynamischen "Design Sciences" - erstaunlich weit entwickelt. (Diese Dynamik erklärt sich auch durch die besonders enge Verzahnung mit den Grundlagen einer Reihe von Anwendungsgebieten.) Inzwischen sind bereits auch Ausstrahlungen in das Gebiet der Compiler für Parallelrechner zu beobachten. Der Begriff "Systolic Arrays" wurde vor ca. einem Jahrzehnt von H.T. Kung (Carnegie- Mellon University) geprägt.

Eine hohe Parallelität der Datenverarbeitung wird - im Gegensatz zu Parallelrechner- Systemen ("Processor arrays": Ensembles von Universalprozessoren) - bei Systolischen Arrays durch "VLSI processing arrays" angestrebt: durch regelmäßige Felder spezialisierter und eher einfacherer "processing elements" (PEs), insbesondere durch solche PE-Felder, die auf einem einzigen VLSI-Chip untergebracht werden. "Systolische Entwurfsverfahren" sind besonders gut dafür geeignet, eine VLSI-gerechte optimale Balance zwischen Kommunikationsaufwand und Verarbeitungsaufwand bereits in sehr frühen Phasen des Entwurfsprozeß zu bestimmen und festzulegen.

Im Gegensatz zu vielen sonstigen Synthese- und Optimierungsverfahren des VLSI- Entwurf basiert der Systolische Entwurf wesentlich auf einer semantischen Analyse des zu implementierenden Algorithmus und seiner jeweiligen Anwendung. U.a. aus dieser engen Verzahnung resultiert die Tatsache, daß bestimmte Anwendungen besser oder weniger gut für eine systolische Implementierung geeignet sind.

Bei systolischen Entwurfsverfahren werden die aus der Anwendung abgeleiteten formalen Datenabhängigkeiten durch Transformation möglichst optimal entsprechend den typischerweise

gegebenen Kommunikations- und Platzierungs-Restriktionen auf die planare Chip-Oberfläche abgebildet. Beispielsweise wird die bei VLSI sehr flächeneffiziente "Broad-Cast"-Kommunikation in flächeneffiziente Pipeline-Strukturen umgewandelt über die Einfügung von Puffer-Registern unter einer globalen Taktungsstrategie.

Insbesondere eignen sich auf rekursiven linearen Gleichungssystemen basierende Anwendungen (z.B. aus der Signalverarbeitung) für eine Abbildung auf eine systolische Verarbeitung. Natürlich eignen sich auch solche Anwendungen besonders gut, die schon von vornherein auf lokalen Nachbarschaftsrelationen aufbauen (z.B. Bildvorverarbeitung). Die Zahl der Anwendungsgebiete ist groß (hier sei auf die spezielle Literatur verwiesen). Eine große Wichtigkeit systolischer Verfahren für eine Anzahl von Anwendungen, insbesondere in der Kommunikationstechnik und auch in anderen Anwendungsgebieten ist unumstritten.

Obwohl eine Anzahl von Beiträgen über Synthese-Verfahren oder über Ansätze zu solchen Verfahren publiziert worden sind, besteht noch ein beträchtlicher Bedarf für weitere Verbesserungen. Auch wurden nicht alle der publizierten Verfahren implementiert. Anhängendes Bild faßt die bisherige Entwicklung in einer Art von "Ahnentafel" zusammen. Die wichtigsten implementierten Vorläufer des SYS³-Verfahrens sind das SYS nach Lam (Carnegie-Mellon-University, 1983) und Mostow (ISI, University of Southern California), SYSTARS nach Omtzigt und Theodore (Yale University, 1988), HIFI nach Annevelink und Dewilde (Universität Delft, 1988), DIASTOL nach Quinton (IRISA-CNRS, Rennes, 1984) und ADVIS nach Moldovan (University of Southern California, 1987). Andere publizierte Beiträge zum Gebiet behandeln eher theoretische Beiträge zu den Grundlagen (beispielsweise durch Entwicklung eines Kalküls oder andere Elemente zur Unterstützung der Synthese), mehr für den manuellen Gebrauch, denn eigentliche Synthese-Algorithmen.

SYS nach Lam und Mostow ist das erste auch implementierte Verfahren, das mehr liefert als nur einen Kalkül. Der Benutzer definiert hier das Anwendungsproblem als algorithmische Beschreibung in Form geschachtelter Schleifen. Zu jeder dieser Schleifen muß der Benutzer in einem speziellen Format auch noch deren (gewünschte ?) Realisierungsform angeben (z.B.: ob parallel, sequentiell, oder als pipe line). Das System generiert dann daraus eine systolische Architektur. Eine optimale Lösung ist nur durch oft längeres Experimentieren mit immer wieder neuen Lösungen möglich. Es ist eher ein Werkzeug denn ein Synthese-Algorithmus. Von Seiten des Benutzers ist dabei ein sehr weitgehendes Verständnis des Verfahrens notwendig. Die Qualität der erzielten Lösungen hängt sehr weitgehend von der "Qualität" des Benutzers ab. Das Verfahren kann man daher nicht als automatisch bezeichnen.

SYSTARS hingegen kann lediglich Algorithmen mit nur einer einzigen Rekursion verarbeiten, ist also von recht geringer Mächtigkeit. Es können hiermit nur Anwendungen mit einem einzigen Ergebnisdatenstrom reasilisiert werden. Algorithmen, zu deren Implementierung Zellen

verschiedenen Typs benötigt werden, können mit diesem Verfahren nicht bearbeitet werden. Die Generierung der Raumabbildung erfordert die interaktiv graphische Manipulation eines Graphen. Auch ist SYSTARS noch nicht vollständig implementiert, sodaß der Benutzer den Zeitabbildungsvektor manuell ableiten und eingeben muß.

Das HIFI-Verfahren ist insofern viel allgemeiner als die anderen Verfahren, als hier die Generierung systolischer Arrays nur ein Spezialfall ist. Es kann auf eine viel größere Klasse von Array-Strukturen angewandt werden. Bei Anwendung des HIFI-Verfahren muß allerdings der Benutzer aus der algorithmischen Beschreibung des Anwendungs- problemes manuell einen regelmäßig strukturierten Signalfluß-Graphen herstellen. Trotz der Unterstützung durch einen entsprechenden interaktiven Graphik-Editor ist der Mechanisierungsgrad daher recht gering. Von voller "Automatisierung" kann keineswegs die Rede sein. Dies ist ein erheblicher Nachteil gegenüber (dem allerdings an Sonsten weit weniger mächtigen) SYSTARS, bei welchem der Mechanisierungsgrad höher ist.

Beim DIASTOL-Algorithmus liegt schon eine weitergehende Mechanisierung vor: Transformation einer Rekursionsgleichung in eine systolische Architektur erfolgt zwar vollautomatisch. Allerdings muß hier zuvor die Rekursionsgleichung manuell in einer bestimmten Form vorbereitet werden: die Datenabhängigkeiten müssen konstant sein. Die so wichtige Elimination von "Broadcast"-Kommunikation ist deshalb auch hier nicht automatisiert. Algorithmen, zu deren Implementierung Zellen verschiedenen Typs benötigt werden, können auch mit diesem Verfahren nicht verarbeitet werden.

Beim ADVIS-System muß der Benutzer aus der algorithmischen Beschreibung der Anwendung einen speziell formatierten sehr abstrahierenden Datensatz manuell ableiten. Das System generiert daraus dann automatisch eine sehr große Zahl alternativer Raum/Zeit-Abbildungen. Diese abstrakten Daten müssen vom Benutzer dann architekturell interpretiert werden, was von diesem erhebliche theoretische Kenntnisse verlangt. Sozusagen ist nur das Mittelteil der Entwurfsprozedur automatisiert. Algorithmen, zu deren Implementierung Zellen verschiedenen Typs benötigt werden, können auch mit diesem Verfahren nicht verarbeitet werden.

Bei allen diesen Verfahren, die man als Vorgänger des SYS³ ansehen muß, sind nur Teile automatisiert. Weitere wesentliche Teile der Entwurfs-Prozedur müssen immer noch "manuell" ausgeführt werden, derart daß der Benutzer sowohl mit den theoretischen Grundlagen als auch mit implementierungstechnischen Besonderheiten der jeweiligen Synthese-Prozedur sehr gut vertraut sein muß. Meist werden auch keine alternativen Architekturen generiert. Die gleichzeitige Anwendung verschiedener PE-Zellentypen wird von den wenigsten dieser Verfahren beherrscht. Die Elimination von Broadcasting muß bei allen diesen Verfahren vom Benutzer manuell durchgeführt werden, wobei durchweg eine aufwendige notationelle Aufbereitung von Hand erforderlich ist. Dies ist insofern wesentlich, als bei VLSI-Realisierungen gerade hier ein

besonders kritischer Optimierungs-Aspekt gegeben ist.

SYS³ hingegen ist allein schon von seinen äußeren Schnittstellen her den bisher bekannten Systemen deutlich überlegen. Die Eingabe des Algorithmus erfordert keine vorausgehende manuelle Aufbereitung, sondern erfolgt direkt in einer dem Benutzer vertrauten Pascal-ähnlichen Notation. Auch die Ausgabe der Ergebnisse über eine relativ siliziumnah modellierende nichtprozedurale und sehr gut lesbare Hardwarebeschreibungssprache erfordert keinerlei Nachbereitung. Die gesamte dazwischenliegende Prozedur wird vollautomatisch ausgeführt. Neu ist also dabei, daß der Benutzer dem System keinerlei Hilfestellung geben muß. Das System ist also das erste seiner Art, zu dessen Bedienung kein Verfahrens-Experte notwendig ist. Der Anwender kann SYS³ als "black box" anwenden und hat somit die Möglichkeit, sich voll auf sein Anwendungsgebiet zu konzentrieren.

SYS³ eliminiert darüber hinaus automatisch das Überschreiben von Variablen, wodurch solche Zuweisungskonflikte automatisch gelöst sind, die von anderen Verfahren entweder nicht entdeckt werden oder aber wegen geringer Mächtigkeit des Verfahrens garnicht darstellbar sind. Weiter werden keinerlei Orientierungs-Varianten generiert, die zu anderen Varianten funktional äquivalent sind. (Anmerkung: "Orientierungs- Varianten" entstehen durch Rotation und/oder Spiegelung.)

Das Verfahren nach Moldovan ist der ausgereifteste Vorgänger von SYS³. Hier zeigt ein Vergleich recht gut den Fortschritt durch SYS³ auf, nämlich: die automatische Elimination des Überschreibens, die Beherrschung von Systemen mit mehreren PE-Zellentypen, die automatische Elimination von "broadcasting", die automatische Aussortierung von Orientierungs-Varianten, sowie die Vermeidung einer aufwendigen manuellen Aufbereitung der Eingabebeschreibung. SYS³ bedient durch all dies auch einen wesentlich breiteren Anwendungsbereich als ADVIS.

Die Implementierung ist fast abgeschlossen. Schnittstellen und Bedieneroberfläche sind gut an die CAE-Umgebung der Arbeitsgruppe angepaßt. Vom SYS³ generierte Architekturbeschreibungen können direkt durch das KARL-System weiterkompiliert und simuliert werden, sowie direkt in andere CAE-Werkzeuge unserer Umgebung eingegeben werden. Es wurde eine voll aufwärtskompatible KARL-3-Erweiterung KARL-4 entwickelt, die auch von SYS³ als Eingabe akzeptiert wird. Auch die von SYS³ generierte Ausgabe wird in der gleichen Sprache ausgedrückt (genauer: in deren Subset KARL-3).

	SCIRTSS	Lai's Algorithmus	S-Algorithmus	KARATE
Vektor-Pfade		X	X	X
Propagierung von Variablen		X	X	X
Propagierung von "constraints"		X		
Propagierung von "joint constraints"	-	-	-	-
Propagierung von Ausdrücken				X
Propagierung durch Register hindurch	(X)			X
austauschbare Fehlermodelle		X		X
erweiterbare Hardware-Sprache				X
automatische Generierung von VP		X		X
vollautomatisch	X	X		X

Abkürzungen:

TM: Test-Muster

KT: Komponenten-Testmuster

VT: Verbund-Testmuster

PM: Propagierungs-Muster

KP: Komponenten-Propagierungsmuster

VP: Verbund-Propagierungsmuster

Vergleich KARATE / S / Lai / Hill