

# 3 Infrastrukturen...

Die allgemeine Wettbewerbsfähigkeit unseres Landes in der Mikroelektronik erfordert die Verfügbarkeit von Infrastrukturen in allen benötigten Formen und in ausreichendem Umfang für wirklich alle in diesem Zusammenhang wichtigen Firmen und Institutionen. Der Begriff der "Infrastrukturen" ist hier im weitesten Sinne gebraucht: privatwirtschaftlich und öffentlich, kurzum alles, was die Wirtschaft, Forschung und Lehre benötigt um ausreichend schnell, effizient und breit der Wettbewerbsfähigkeit dienen zu können - kurzfristig und weitsichtig. Neben Entwurfswerkzeugen und dynamischem neuartig qualifiziertem Personal (vgl. Kapitel 3) sind auch die in diesem Kapitel behandelten technologischen Infrastrukturen von eminenter Wichtigkeit. Es handelt sich um ein ganzes Spektrum unterschiedlicher Infrastrukturen für die Fertigung von Prototypen und Produkten. Alle Arten solcher Infrastrukturen sind wichtig. Die Bedeutung der verschiedenen Strukturen läßt sich am besten vergleichen, wenn diese an den Ursachen ihrer historischen Entwicklung veranschaulicht wird.

## 3.1 Die Technologie der 60er- und 70er-Jahre

Die allgemeine Wettbewerbsfähigkeit unseres Landes in der Mikroelektronik erfordert die Verfügbarkeit technologischer Infrastrukturen und Dienstleistungen in allen benötigten Formen und in ausreichendem Umfang für wirklich alle in diesem Zusammenhang wichtigen Firmen und Institutionen. In etwa den ersten beiden Jahrzehnten nach Erfindung der integrierten Schaltung (vgl. Abschnitt 6.1.1 und 6.1.2) war die Situation noch relativ einfach. Es gab praktisch nur die Art integrierter Schaltungen, die man heute als Vollkondensaltungen bezeichnen würde (siehe im Glossar auf Seite 68 oder spätere Abschnitte). Erst nach Vorliegen des vollständigen Design erfolgte die Fertigung (Bild 3.1).

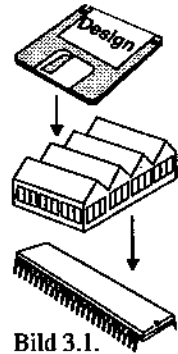


Bild 3.1.

**Die Fertigungskosten** waren zwar hoch, jedoch bei Weitem nicht so hoch wie bei modernen Technologien der 90er-Jahre (vgl. Kosten-Entwicklung für Fertigungslinien in Bild 1.12). Die große Anzahl von Fertigungs-Schritten ist die Ursache für hohe Kosten. Bei modernen Technologien sind bis zu mehr als hundert Einzelschritte erforderlich. Deshalb sind Fertigungs-Losgrößen von (seltener) 20 bis (häufiger) 50 Scheiben üblich. Dabei werden mit jeder Scheibe meist mehr als 100 Mikro-Chips gleichzeitig hergestellt (siehe Bild 9.12). Auf diese Weise werden mit einer Charge von Scheiben bis zu Zehntausend oder mehr Chips produziert.

**Mikro-Chips und Fotografie.** Wie ist denn die hohe Zahl von Fertigungsschritten zu erklären? Mit anschaulichen Darstellungen zeigt Kapitel 9 dies an Hand der relativ einfachen nMOS-Technologie. Die Herstellung integrierter Schaltungen kann mit Belichten und Entwickeln einer Farbphotographie bei der Aufnahme mit einer Repro-Kamera verglichen werden. Bei der Fotografie werden mit dem Objektiv der Kamera erfaßten geometrischen Muster der ebenen

3.1 Die Technologie der 60er- und 70er-Jahre .....	67
3.2 ASICs.....	71
3.3 Anwender-programmierbare Schaltungen .....	74
3.4 Schon wieder ein Struktur-Wandel ? .....	76
3.5 Prototyping.....	77
3.6 Was ist zu tun? .....	81
3.7 Literatur.....	84



## Glossar

*Die Dominanz von Fachausdrücken und Abkürzungen aus der englischen Sprache ist ein Symptom für Deutschlands zweit-rangige Rolle in der Mikroelektronik.*

**ASIC** (engl.: **application-specific IC**) Anwendungs-spezifische integrierte Schaltung  
1) **mASIC** (**M**asken-programmierbar), 2) **rASIC** (elektrisch rekonfigurierbar).

**bipolar** (schnelle teure Technologie).

**CLB** (engl.: **configurable logic block**) konfigurierbarer logischer Funktionsblock

**CMOS** (gängige Technologie hoher Dichte)

**field-** draußen beim Kunden (Anwender).

**field-programmable** strukturell Anwender-programmierbar; durch Laden von Konfigurations-Kode elektrisch programmierbar.

**FPGA** (engl.: **field-programmable gate array**) FPLCs mit einfachen CLB's.

**FPIC** 1) (engl.: **field-programmable interconnect circuit**) IC ALS elektrisch konfigurierbares Verbindungsnetzwerk (Beispiel: Kreuzschienen-Verteiler); - 2) (engl.: **field-programmable IC**) s. FPLC.

**FPLC** (**field-programmable logic circuit**) wichtigste Klassen: PLDs und FPGAs.

**FPLD** (engl.: **programmable logic device**) Klasse von PLC's mit PLA-ähnlichen konfigurierbaren Funktionsblöcken.

**full custom circuits.** Vollkunden-Schaltung

**GaAs** Gallium-Arsenid (schnelle Technologie)

**Gate Array** wichtige Klasse der **mASIC**'s.

**IC** (**integrated circuit**) integrierte Schaltung.

**Interkonnekt** Verbindungs-Leitungen.

**Interkonnekt-Programmierung** Konfigurierung von Interkonnekt.

**Katalog-Schaltung** s. Standard-Schaltung

**Konfigurierung** Personifizierung (strukturelle Programmierung).

**Konfiguration** strukturelles Programm.

**Konfigurations-File** Konfigurations-Kode

**Konfigurations-Kode** zur Personifizierung in FPLCs zu ladende Information.

**Maschinen-Kode** direkt ausführbarer sequentieller Programm-Kode f. Computer.

**mASIC** (engl.: **mask-programmable ASIC**)

siehe Masken-Programmierung.

**Masken-Programmierung** auf vorfabrizierte **mASIC**-Wafer (z.B. Gate Arrays) werden als Personifizierung die Anwendung-spezifischen letzten Verdrahtungs-Ebenen hinzu fabriziert (beim Hersteller).

**Mikroprozessor** Computer als IC.

**pASIC** (**programmable ASIC**) s. **rASIC**.

**Personifizierung** (Re-)Programmierung einer Schaltung "ohne Eigenschaften" (nackte Schaltung): 1) bei Computern durch Maschinen-Kode, 2) bei Maskenprogrammierbaren ASIC's.

**personalization** (engl.) s. Personifizierung

**Programm** 1) *prozedural*: sequentielles Programm für einen Mikroprozessor oder anderen Computer; 2) *strukturell*: Konfigurations-Information für eine ASIC.

**Programm-Kode** s. Maschinen-Kode

**Programmierung** Personifizierung: 1) *prozedural*: bei Mikroprozessoren (u. anderen Rechnern) durch Laden des Arbeitsspeichers mit Maschinen-Kode, 2) *strukturell*: bei ASIC's, a) bei **mASIC**'s durch *Masken-Programmierung* (s. dort), b) bei **rASIC**'s *elektrisch* (Konfigurations-Kode).

**prozedurale Programmierung** Computer-Programmierung d. Maschinen-Kode.

**prozeduraler Kode** s. Maschinen-Kode.

**qASIC** (**quick ASIC**) **mASIC**-Variante mit Technologie für schnelle Personifizierung.

**rASIC** elektrisch (re-)konfigurierbare ASIC.

**standard cell** (engl.) s. Standard-Zellen.

**standard circuit** s. Standard-Schaltung.

**Standard-Schaltung** IC nach Katalog

**Standard-Zellen-IC** Kunden-spezifische IC; automatische Verdrahtung von Bibliotheks-Zellen gleicher Höhe - kein **pASIC**.

**tooling** Eingangs-EDV f. Konfigurationskode

**Universal-Schaltungen** für eine sehr große Anzahl verschiedenartiger Anwendungen geeignete Schaltung; typisches Beispiel: Mikroprozessor (wegen seiner universell prozeduralen Programmierbarkeit durch das von-Neumann-Prinzip).

**Vollkundenschaltung** vollständig nach Kunden-Spezifikation hergestellte integrierte Schaltung (Gegensatz: **ASIC**, siehe dort).

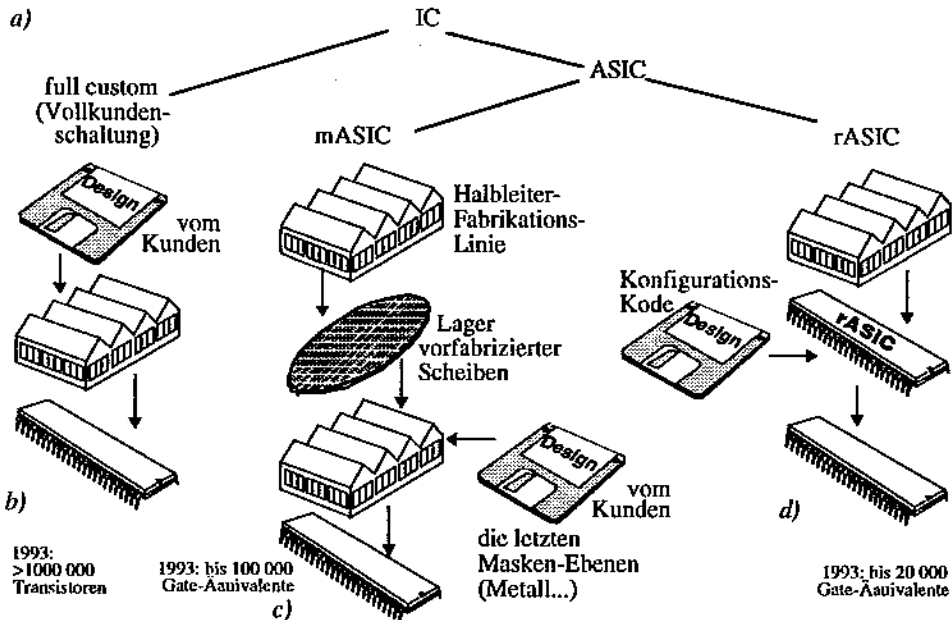


Bild 3.2: Kunden-spezifische ICs: a) drei Methoden der Arbeitsteilung mit dem Hersteller, b) Personalisierung voll beim Hersteller, c) Personalisierung auf die vorfabrizierten Scheiben nur über (spezifische) Verdrahtungs-Masken, c) Personalisierung voll durch den Anwender.

(planaren) Vorlage auf den Negativ-Film belichtet, allerdings für alle drei Farben gleichzeitig. Vom Film werden die Farben durch erneutes Belichten (Vergrößerungs-Apparat) auf das Fotopapier übertragen. Hinzu kommt noch das Entwickeln von Film und Papier. Dies sind insgesamt 4 Schritte: 2 mal belichten und 2 mal entwickeln.

**Weit mehr als drei Farben.** Wegen der Technologie der 90er-Jahre sei zugegeben, daß dieser Vergleich hinkt. Bei der frühen Technologie der 60er-Jahre bis hinein in die 70er wurde tatsächlich eine Repro-Kamera verwendet (vgl. Bild 9.19 a, linke Seite). Bei integrierten Schaltungen werden jedoch statt drei Farben die Muster für bis etwa ein Dutzend Materialien auf die Scheibe übertragen - nicht gleichzeitig, sondern nacheinander. Die Schichten sind: 1 oder 2 mal polykristallines Silizium, mehrere Oxid-Schichten, 2 oder mehr "Diff"-Schichten<sup>1</sup>, 1 bis 3 Metall-Lagen, und 2 bis 4 verschiedene Ebenen von Kontaktlöchern. Dies entspräche einer Fotografie mit bis zu 12 Farben. Auf dem Bildschirm moderner Design-Workstations können sogar bis zu 256 Farben dargestellt werden, denn auch Pseudo-Ebenen durch Überlappung mehrere Materialien sind für wichtig für Computer-gestützte Entwurfsverfahren.

1) Die Terminologie ist nicht ganz professionell, da ich kein Technologe bin. Es ist zu wünschen daß eine möglichst breite Öffentlichkeit von Nicht-Technologen sich für solche Techniken interessiert (vgl. Kasten "Aufruf zur General-Mobilmachung" auf Seite 49). Das Engagement der Wissenschafts-Journalisten sei aufgerufen!

### Drei mal Mikroelektronik

Bezüglich Infrastrukturen und der Zusammenarbeit zwischen Anwender und Hersteller von Mikro-Chips unterscheiden wir in der Industrie heute grob 3 verschiedene High-Tech-Szenen (Bild 3.2): 1.) Vollkondensaltungen (hergestellt wie Standard-Schaltungen), 2.) mASICs (Masken-programmierte ASICs), und 3.) rASICs (reconfigurable ASICs: Anwender-programmierbare Schaltungen). Alle drei Szenen sind eminent wichtig für unsere Wettbewerbsfähigkeit. Ein sehr gutes Gedeihen aller drei Zweige muß unabdingbar das Anliegen einer guten Industrie- und Handelspolitik sein.

**Hohe Zahl von Fertigungs-Schritten.** Außerdem ist die Zahl der Schritte pro Ebene meist deutlich höher als 4, da die Übertragung der geometrischen Muster auf den Chip wesentlich komplizierter ist (vgl. Kapitel 9). Im Vergleich zur Fotografie entspricht dies jeweils einer Folge mehrerer Zwischen-Negative: Retikel und Maske (Bild 9.20), sowie Fotolack und Ätz-Resist (Bild 9.10). Hinzu kommen noch weitere Fabrikations-Techniken (Bild 9.5): das Testen die Montage in das Chip-Gehäuse (Bild 9.24). Die Gehäuse-Technik ist mit zunehmender Zahl von Anschluß-Pins und fortschreibender Miniaturisierung (siehe Bild 13.9 und Bild 13.16) sogar zu einem oft dominierenden Kostenfaktor geworden.

**Wichtigkeit der Prototypen-Erprobung.** Kein neues Auto-Modell ohne Erbkönig, kein neuer Flugzeugtyp ohne Testflüge von Prototypen, kein neuer Mikro-Chip ohne vorherige Erprobung. (statt Mißbrauch des Kunden als Versuchskaninchen). Man denke vor Allem an kritische Anwendungen mit fatalenfolgen der Fehlfunktion eines Chip (z. B. in der Steuerung einer Rakete). Ansonsten ist das Testen des Prototypen ein Qualitätsfaktor, eine wichtige Voraussetzung zur Wettbewerbsfähigkeit. Aber auch für die Praxis-nahe Lehre ist der Test von Prototypen eine sehr wichtige Funktion. Im Unterschied zum Elfenbein-Turm praktiziert man Wettbewerbs-relevante Lehre durch "learning by doing". Industrie-gerechte Lehre mit Erziehung zur Verantwortung für die Wettbewerbsfähigkeit gibt es nur dann, wenn auch das Testproblem mit allen seinen Aspekten vom Studenten selbst erfahren wird. Gemäß dieser Kenntnis gibt es für Universitäten in den USA [4] (und zuerst in Deutschland [14], dann durch das EUROCHIP-Projekt Europa-weit [28][29]) eine 100% staatlich finanzierte Infrastruktur zur Besorgung der Prototypen-Fertigung.

**Wirtschaftliche Prototypen-Fertigung.** Angesichts hoher Fertigungskosten ist die Prototypen-Fertigung ein Problem. Für die Erprobung eines neuen Chip-Design werden jedoch nur wenige Chips benötigt, für die Lehre meist nur für kleinere Übungs-Schaltungen, wovon aufgrund ihrer geringen Fläche mehrere Schaltungen auf einem Chip untergebracht werden kann. Einen entsprechenden Chip nennt man *Multi-Projekt-Chip (MPC)* [12]. Bild 2.5 ist ein Mikrofoto des ersten MPC des E.I.S.-Projekt<sup>2</sup> (siehe auch Bild 9.6 a). Hierdurch können die Herstellungskosten eines Chip auf mehrere Übungsschaltungen umgelegt werden. An die Anschlußfahnen des Chip-Gehäuse (vgl. Bild 9.24) wird nur eine der Schaltungen angeschlossen gemäß der bonding map (s. Bild 9.26) für diese Schaltung. Das MPC-Umlegungsverfahren kann noch durch eine weitere Ebene verstärkt werden: mehrere MPCs auf einem Wafer, einem sogenannten Multi-Chip-Wafer (MCW) [12], wie in Bild Bild 9.6 b zu sehen ist.

**Klare Schnittstelle zwischen Design und Technologie.** Zwischen *Design* und Technologie entstand im Zuge der Entstehung des MOSIS-Dienst (s. Absatz "Neue Infrastrukturen" auf S.

<sup>2</sup> Das E.I.S.-Projekt (vgl. Kasten "Zähe dicke Bretter" auf Seite 59) und seine Rolle (vgl. Kasten "Professoren auf die Schulbank" auf Seite 57) wird in Kapitel 2 behandelt.



S. 60) eine stark vereinfachte Schnittstelle mit einer neuen Form von *Design Rules* (Layout-Regeln) zusammen mit einheitlichen Design-Datenformaten wie z. B. CIF (siehe auf Seite 298). Die Layout-Regeln werden meist von Seiten der Technologie vorgegeben (s. Kapitel 12). Gelegentlich werden Layout-Regeln auch in Zusammenarbeit mit der Design-Seite entwickelt, wie beispielsweise die "gerasterten CMOS-Layout-Regeln" nach Bild 12.34 und Bild 12.35 [14] oder nach Bild 12.36 u. Bild 12.37 (nähere Einzelheiten in späteren Abschnitten dieses Buches). Oft wird Layout per Datenfernübertragung übermittelt, wie beispielsweise für das E.I.S.-Projekt über das DFN (Deutsches Forschungs-Netz)[14]. Ein Vorbild ist der 1980 gegründete MOSIS-Dienst [4], der den Eingang von jährlich Hunderten von Entwürfen über das ARPA-Netz mit Computern vollautomatisch abwickelt nebst Eingangs-Tests der Design-Files, Zusammenstellung zu MPCs und Status-Information (per electronic mail mit Meldungen wie: received, checked, submitted for fabrication, fabricated, mounted, shipped etc., und notfalls detaillierten Fehlermeldungen).

### 3.2 ASICs

Geringe Stückzahlen sind nicht nur für Prototyping interessant sondern durch die weite Verbreitung der Mikroelektronik auch für Produkte. Im Zuge der Mehrwert-Schöpfung durch die Integration von Mikroelektronik in sehr viele andere Industriezweige hinein (vgl. Abschnitt 1.1) werden oft die unterschiedlichsten speziellen Mikro-

Chips in geringeren Stückzahlen benötigt. Somit ist die alte Regel unhaltbar geworden, daß der Design eines Mikro-Chip erst ab Stückzahlen über Zehntausenden oder gar über Hunderttausend lohne. Auch sind bei der kurzen Lebensdauer vieler Produkte die langen Lieferzeiten (Monate) untragbar.

**Schlüsselbausteine.** In ein Produkt eingebaute Anwendungs-spezifische integrierte Schaltungen sind oft Schlüssel zu dessen Wettbewerbsfähigkeit. Beispielsweise die meisten heutigen Produkte aus Unterhaltungs-Elektronik, Mobilfunk, Fahrzeug-Industrie, Luft- und Raumfahrt u. a. sind ohne solche Schlüssel-Chips undenkbar (vgl. Kasten "Schlüssel-ICs" auf Seite 71). Oft ist ein Mikrorechner oder eine Mikro-Steuerung (prozedural programmierbare Standard-

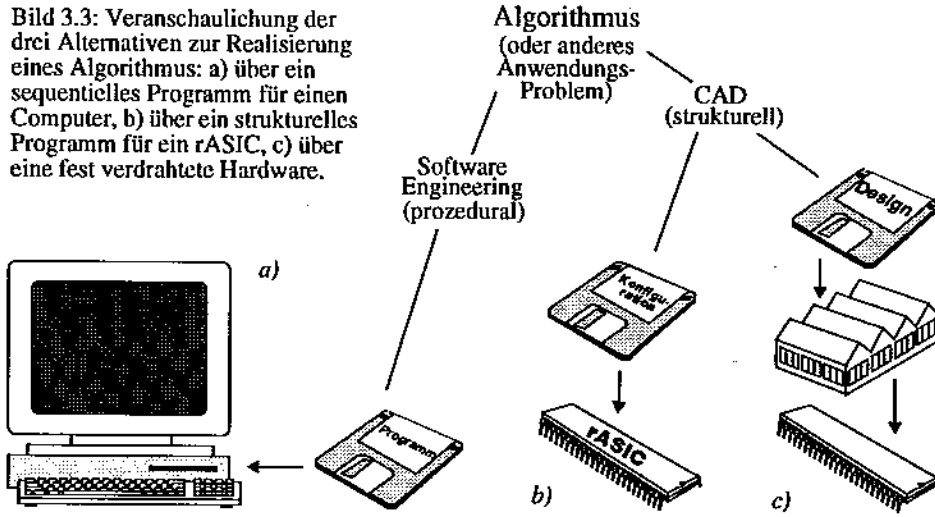
### Schlüssel-ICs

Bis Anfang der 80er-Jahre waren Schaltungen ab Katalog vorherrschend. Oft war für ein bestimmtes Problem kein brauchbares Chip im Handel: der Schlüssel zur Innovation fehlte. Ohne Anbieter kein Bedarf: die Produkt-Idee wurde als nicht machbar oder zu teuer verworfen.

Heute gibt es sich rasch fortentwickelnde Techniken und Anbieter, die selbst mittelständischen Firmen rasch zu Schlüssel-ICs verhelfen können. Die Technik der ASICs ist also der Schlüssel zu einem großen Schlüsselkasten fast grenzenloser Möglichkeiten der Wettbewerbsfähigkeit durch innovative und weitsichtig flexible Produkt-Familienplanung auf unzähligen Gebieten der Mikroelektronik-Anwendung. Angesichts drohender Industrie-Spionage sollten genügend solche Anbieter im eigenen Lande sitzen (Firmen japanischer Firmen?).

**Ein Bildungsproblem.** Schlüssel-ICs sind ein Bildungsproblem. Beispielsweise Informatiker, die sich in Technischer Informatik (Hardware) vertieft haben, sind hier selten und schwer zu bekommen (sogar jetzt noch: Ende 1993). Sind Schlüssel-ICs auch ein Management-Problem? Vor Jahren hatte die landesweite Umfrage der IHK Koblenz nach Hemmnissen der Mikroelektronik-Anwendung überraschende Ergebnisse: viele Firmen verstanden die Frage überhaupt nicht. Auch ein Bildungsproblem.

Bild 3.3: Veranschaulichung der drei Alternativen zur Realisierung eines Algorithmus: a) über ein sequentielles Programm für einen Computer, b) über ein strukturelles Programm für ein rASIC, c) über eine fest verdrahtete Hardware.



Bausteine) bei Realzeit-Anforderungen zu langsam. In einem solchen Fall muß meist der Durchsatz-Engpaß (etwa eine sehr oft durchlaufene Programm-Schleife) aus der Software ausgelagert und in Hardware verwandelt werden: es wird ein ASIC benötigt. Solche Probleme kommen sehr oft bei sogenannten *embedded systems* vor, wo Mikroelektronik mit einem technischen Gerät zusammenarbeiten muß (ein CD-Player ist beispielsweise ein solches *embedded system*). Der Marktanteil von *embedded systems* sowie deren Anteil am Verbrauch von Mikroelektronik ist steigend. Damit nimmt für unsere Wettbewerbsfähigkeit auch die Wichtigkeit von ASICs zu und von zur Anwendung qualifiziertem Personal. Nebenbei: auch ist der Verbrauch eines Landes an ASICs eine Meßzahl der Wettbewerbsfähigkeit (vgl. Kasten "Pro-Kopf-Verbrauch an Mikroelektronik" auf Seite 72).

**Pro-Kopf-Verbrauch an Mikroelektronik**

ist ein Indikator für den High-Tech-Entwicklungsstand eines Landes, ähnlich, wie die Zahl der Telefon-Anschlüsse und PKWs pro Kopf ein Maß für den Lebensstandard sind. Mitte der 80-er-Jahre betrug bei integrierten Schaltungen diese Zahl weniger als ein Drittel der Werte in den USA und Japan. Wir haben also schon vor 10 Jahren den Grundstein gelegt für unser heutiges Desaster. Ein noch besseres Maß für unsere Wettbewerbsfähigkeit wäre der Verbrauch an Schlüssel-Bausteinen. Diese sind von der deutschen Presse nicht publik gemacht worden (vgl. Absatz "Rückständiges Bewußtsein der Deutschen" aus Seite 28).

Verbrauch eines Landes an ASICs eine Meßzahl der Wettbewerbsfähigkeit (vgl. Kasten "Pro-Kopf-Verbrauch an Mikroelektronik" auf Seite 72).

**Schnelle Produktentwicklung tut not:**

Bild 1.10 zeigt einen Ausschnitt der sog. ersten Gordon-Moore-Kurve. Um zu veranschaulichen, wie wichtig es unter Umständen ist, mit der Produkt-Entwicklung sehr schnell zu sein, zeigt Bild 1.21 Beispiele rapiden Preis-Verfalls bei Katalog-Schaltungen. Bei verspätetem Markteintritt kann nichts mehr verdient werden, selbst bei an sich guten Verkaufs-Stückzahlen. Bei ASICs ist eine schnelle Produktentwicklung zumindest ebenso kritisch. Bei einer Reihe von Märkten, wie beispielsweise Konsum-Elektronik veralten



Produkte oft derart schnell, daß man bei verspätete Markteintritt das Produkt kaum noch absetzen kann. Oft hat das Timing dabei noch eine strategische Dimension, indem entschieden wird, welcher Standard sich durchsetzt (Beispiele: JVS gegen 2000 bei Video-Rekordern, digitale Kassette gegen Mini-CD bei kleinsten Audio-Rekordern).

**Eine neue, zweite Technologie- und Entwurfs-Szene.** Aus diesem Bedarf heraus entstand ab Anfang der 80er-Jahre ein zweiter neuer Zweig der Mikroelektronik, der heute unter dem Stichwort ASIC bekannt ist (application-specific integrated circuit). Zur Unterscheidung von einer gegen Ende der 80er-Jahre entstandenen wiederum neuen Szene (rASICs), nennen wir diese Schaltungen mASICs (vgl. Kasten "Glossar" auf Seite 68). Bild 3.2 veranschaulicht den Überblick (rASICs werden erst in Abschnitt 3.3 behandelt). Bald entwickelte sich neben den neuartigen technologischen Infrastrukturen hierzu auch eine Vielfalt

### ASIC oder Mikroprozessor?

Ein ASIC ist oft die bessere Alternative zu Software auf einem Computer (Bild 3.3). Bei vielen Anwendungen ist ein Mikroprozessor wegen seiner sequentiellen Arbeitsweise aber zu langsam, vielleicht auch zuteuer. Ein ASIC kann um Größenordnungen leistungsfähiger sein. Bei der Produktplanung muß daher oft entschieden werden, ob ein ASIC oder ein Mikroprozessor gewählt werden soll, oder gar eine Mischung aus beidem. Bei komplexen Systemen ist eine solche Misch-Lösung oft die häufigste.

Für die Industrie ist hier die Produkt-Entwicklung ein Bildungsproblem. Wegen vorwiegend prozeduraler Denkweise kann bei Weitem nicht jeder heutige Informatiker problemlos mit Hardware- und Software-Lösungen des gleichen Problems jonglieren. Das Problem wurde jedoch von den Universitäten erkannt. Die Innovierung der Studien-Angebote ist im Gange. Die neue Wissenschaft des Hardware/Software Ko-Design [35] entwickelt die Grundlagen auch für die Lehre. Hier ist sogar Deutschland präsent [16] mit 3 Universitäten und die Forschung des Siemens-Zentrallabor ist führend. Mögen allen Beteiligten der in diesem unseren Lande übliche Frust auch künftig erspart bleiben.

ASIC-orientierter CAD-Hilfsmittel für den Entwurf. Im Gegensatz zu ASICs werden nach klassischer Art entstandene Chips *Vollkunden-Schaltungen* genannt (engl.: *full custom circuits*). **Masken-programmierte ASICs (mASICs).** Die Grund-Idee der neuen Infrastruktur besteht darin, daß möglichst viele Fabrikations-Schritte für viele Designs gemeinsam durchgeführt werden können. Es wird also erst eine unspezifische Grundstruktur nach dem Design des Herstellers fabriziert, wie beispielsweise nach dem Beispiel des vereinfachten Schema des Gate Array in Bild 9.25. Es ist hier beispielsweise eine große Anzahl von Gattern und Anschlußpin-Treibern vorhanden, jedoch ohne Interkonnect. Diese so vofabrizierten Scheiben werden gelagert (vgl. Bild 3.2 c). Nach Eingang einer Kunden-Bestellung mit der spezifischen Verdrahtungsliste werden dann die wenigen restlichen Fabrikationsschritte speziell für den Kunden durchgeführt. Nur wenige der gelagerten "nackten" Wafer "ohne Eigenschaften" werden somit für den Kunden personifiziert. Auf diese Weise können die Kosten für die Vorfabrikation auf eine größere Anzahl Kunden umgelegt werden. Ein zweiter sehr wichtiger Vorteil ist wegen oft kritischer Dauer der Produkt-Zyklen die verkürzte Lieferzeit, da nur noch wenige Fabrikationsschritte gefehlt haben. Die Bezeichnung mASIC rührt daher, daß nur wenige Masken nach Kunden-Spezifikation quasi "die strukturelle Programmierung" der nackten Wafer durchführen. Der eingebürgerte Begriff der "Programmierung" ist in diesem Zusammenhang allerdings verwirrend (da man meist hierauf "von Neumann ...sequentieller Kode .." etc. assoziiert), weshalb ich das Wort "Personifizierung" bevorzuge.

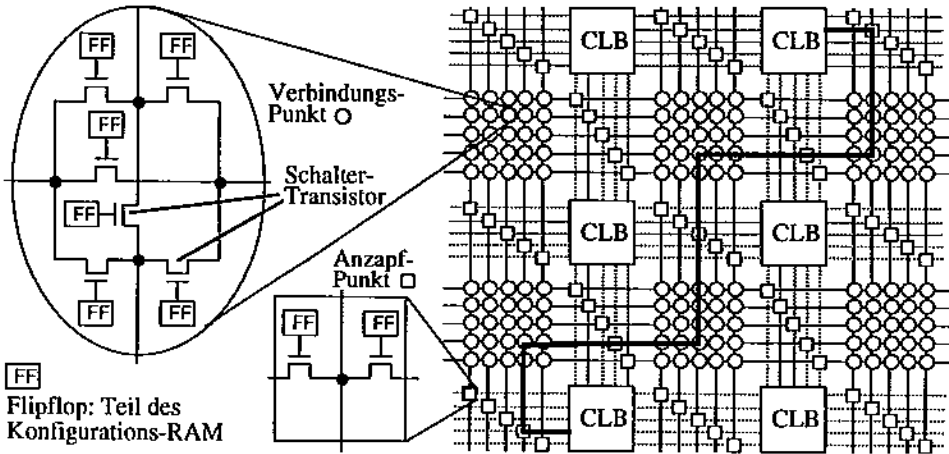


Bild 3.4: Routing eines "net" auf einem Interkonnekt-programmierbaren Leitungs-Netzwerk.

**Standard-Zellen.** Eine mit den mASICs konkurrierende ebenfalls gebräuchliche Form der Kunden-spezifischen Schaltung ist die Zellen-basierte Schaltung, wie beispielsweise mit (wegen ihrer einheitlichen Höhe so genannten) Standard-Zellen (Bild 4.7 bzw. Bild 4.8). Hier wird wie bei Vollkunden-Schaltungen der gesamte Fertigungs-Prozeß individuell durchgeführt, was teuer ist. Solche Schaltungen werden

auf der Basis einer Bibliothek von Standardzellen durch ein Routing-&-Placement-Programm automatisch synthetisiert, wodurch die Entwurfszeit stark kurz ist. Der Vorteil gegenüber mASICs ist die höhere Flexibilität: es können alle Zellen ausgenutzt werden, da die Platzierung der Zellen erst bei dem Design entschieden wird. Die Verdrahtungskanäle können überall so breit wie

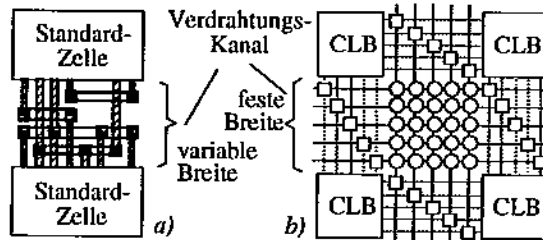


Bild 3.5: Zur Veranschaulichung des Verdrahtungs-Problems: a) bei Standard-Zellen, b) bei FPGAs.

nötig gestaltet werden (im Gegensatz zu mASICs, wo die Breite der Verdrahtungskanäle bereits bei der Vorfabrikation festgelegt wird). Im konkreten Fall ist oft schwer zu entscheiden, welche Lösung die günstigere ist. Mehr und mehr werden in Standard-Zellen ausgeführte Subsysteme als Teile von Vollkundenschaltungen eingesetzt, die ja ohnehin die gesamte Fertigungstiefe brauchen. Wichtig ist hier der hohe Automatisierungsgrad des Entwurf.

### 3.3 Anwender-programmierbare Schaltungen

Auch bei mASICs sind immer noch relativ hohe Stückzahlen zur Rentabilität erforderlich. Schon bei der Nachfertigung von nur einem einzigen Wafer fallen oft über hundert Schaltungen an. Gelegentlich bei Produkten, aber vor Allem beim Prototyping werden vielleicht nur 5



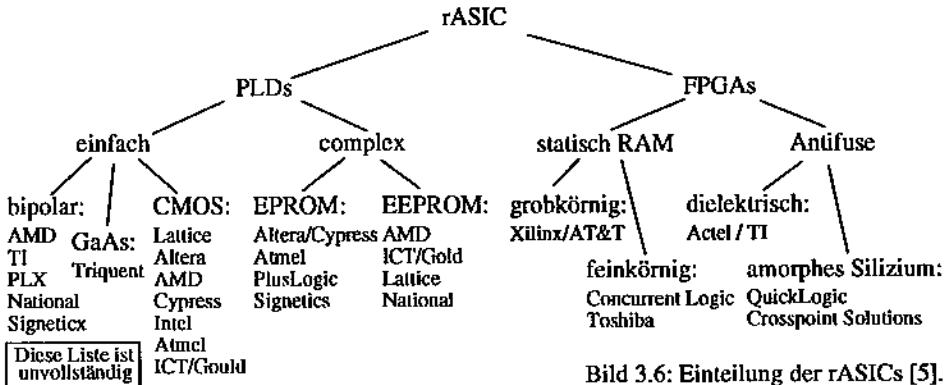


Bild 3.6: Einteilung der rASICs [5].

oder 10 Schaltungen benötigt. MPCs und MPWs sind in dieser Szene häufig nicht sinnvoll. mASICs sind also immer noch ein wichtiger Kostenfaktor. Eine Alternative sind hier sogenannte *rASICs*, bei denen die Personifizierung nicht durch Fabrikations-Schritte bewirkt wird, sondern auf elektrischem Wege nach der völligen Fertigstellung und Montage der Schaltung (vgl. Bild 3.2 d). Der Kunde kann die Schaltungen selbst lagern, wodurch die Lieferzeit praktisch auf Null zurückgeht. Wir haben hier eine **dritte Szene der Mikroelektronik-Anwendung**.

Die Abkürzung *rASIC* steht für *reconfigurable ASIC*. Näheres über rASICs und die Terminologie ist in diesem Buch an anderer Stelle zu finden (vgl. Kasten "Glossar" auf Seite 68) sowie in [1] [2] [5] [10] [11] [23] [26] [32] [34] [36] [37]. Bild 3.6 zeigt eine Einteilung der rASICs und nennt einige Anbieter (diese Liste ist nicht vollständig [5]). Algotronix ist der einzige Europäische Anbieter (Edinburgh). Die USA sind mit großem Abstand führend, auch vor Japan. PLDs sind im Grundprinzip ähnlich wie PLAs aufgebaut (vgl. Abschnitt 11.4.1). FPGAs hingegen sind im Prinzip den Gate Arrays ähnlich (vgl. Bild 9.25). Wie bei Gate-Arrays sind fest platzierte Funktionsblöcke vorhanden, zwischen denen die Verdrahtung hindurch gelegt werden kann. Allerdings werden die kunden-spezifischen Verdrahtungen nicht durch Fabrikations-Schritte aufgebracht, sondern in einem schon vorliegenden schaltbaren Geflecht von Interkonnekt konfiguriert durch laden eines Konfigurations-Files in den Baustein. Dies wird veranschaulicht am Beispiel in Bild 3.4.

Für Funktionsblöcke (CLB), deren Funktion aus gegebenem Repertoire wählbar sind per Konfigurations-File, gibt es viele Architekturen. Ein Beispiel solcher CLBs nach dem LUT-Prinzip

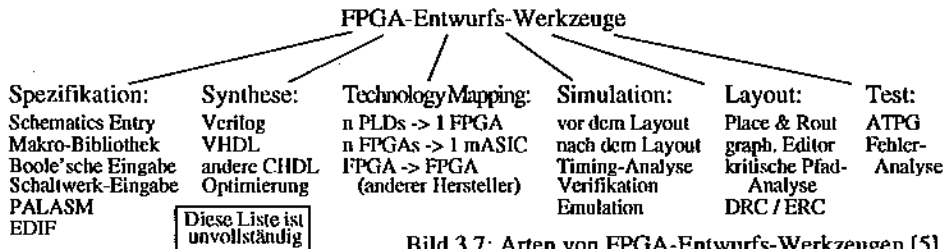


Bild 3.7: Arten von FPGA-Entwurfs-Werkzeugen [5].

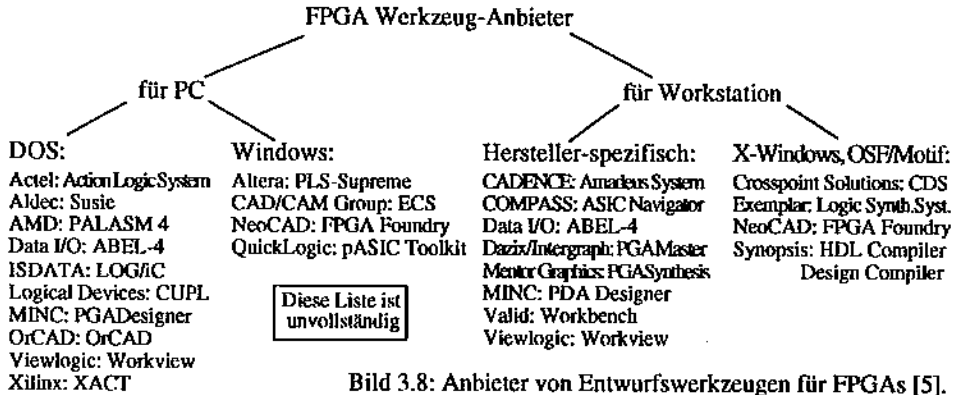


Bild 3.8: Anbieter von Entwurfswerkzeugen für FPGAs [5].

ist durch die Multiplexer-basierte Realisierung von ULMs in Bild 17.11 illustriert. Auch hier ist ein Spektrum guter CAD-Werkzeuge der Schlüssel der Wettbewerbsfähigkeit bei der rASIC-Anwendung (vgl. Kasten "Ohne Software geht nichts" auf Seite 76). Bild 3.7 listet die wichtigsten Arten von Entwurfswerkzeugen für die rASICs (z. B. [21]). Bild 3.8 zeigt die wichtigsten Anbieter von Entwurfswerkzeugen und deren Produkte. Beide Listen erheben keinen Anspruch auf Vollständigkeit. Im Zuge des Vordringens der Mikroelektronik-Anwendung nimmt der Bedarf an Design-Kapazität wegen der zunehmenden Vielfalt benötigter "Schlüssel-Chips" sehr rasch zu. Sie dienen dort als Schlüssel der Innovationskraft durch Mikroelektronik und der Wertschöpfung.

### Ohne Software geht nichts

Bei ASICs (mit Ausnahme des Standardzellen-Verfahrens, s. Bild 3.5, vgl. auch Kapitel 14) können in der Regel nicht alle vorhandenen Gatter auch verwendet werden. Deshalb sind die von Vertriebsleuten genannten Gatterzahlen ebenso mit Vorsicht zu genießen wie die MIPS-Zahlen bei den Mikroprozessor-Anbietern (interessant ist doch nur, was beim Endverbraucher effektiv ankommt). Oft ist nur weniger als 50% der Gatter auch verdrahtbar wegen Verdrahtungs-Stau. Das Problem ist abhängig von der Architektur des rASIC-Bausteins als auch von der Qualität der CAD-Werkzeuge. Wir stehen hier noch ganz am Anfang. Viel Forschung ist noch nötig. Nicht nur beim Routing-Problem ist die Güte der verfügbaren Software zur Schlüssel der Qualifikation des ASIC-Anwenders und somit seiner Wettbewerbsfähigkeit: Bild 3.7 zeigt eine Einteilung der Entwurfswerkzeuge für FPGAs. Bild 3.8 listet einige Anbieter mit ihren CAD-Produkten. Mitunter verfügen Firmen über Eigenentwicklungen, die der käuflichen Software überlegen ist.

### 3.4 Schon wieder ein Struktur-Wandel ?

Mit einer völlig neuartigen Chip-Fabrik, genannt *Single Wafer Fab*, will Texas Instruments die Halbleiterproduktion künftig wesentlich wirtschaftlicher gestalten, insbesondere auch bei sehr niedrigen Stückzahlen [30] (vgl. Kasten "Chips aus dem Automaten" auf Seite 82). Hier bahnt sich wiederum ein neuer Strukturwandel an (wir können gleich durchschlafen vom bisher verschlafenen Strukturwandel in den nächsten). Mit dieser neuen Technologie sind sogar einzelne Wafer schnell (in Stunden oder Tagen) und wirtschaftlich herstellbar. Werden dann die mASICs als eine Übergangs-Technik wieder von der Bildfläche verschwinden ?



### 3.5 Prototyping

Design von Hardware kann mit der Implementierung von Software verglichen werden (vgl. Gegenüberstellung in Bild 3.13). Es werden später gezeigt, daß es hier viele Parallelen gibt. Die Software-Leute haben zwar eine etwas unterschiedliche Terminologie, jedoch auch hier wird vernünftigerweise erst einmal ein Strukturplan entworfen, bevor man mit der Implementierung beginnt. Nach der Kodierung des Programmes, etwa in einer Hochsprache, wird dieses durch einen Compiler übersetzt, der daraus den Maschinen-Kode generiert. Dies ist dann sozusagen das Endprodukt, das dann auf einem Rechner exekutiert werden kann. Beim Test dieses Endproduktes werden Fehler gefunden, die dann durch *debugging* beseitigt werden müssen. Mit dem Maschinen-Kode der verbesserten Version des Programmes (durch erneute Kompilation generiert) wird wiederum ein Testlauf gestartet. Es entsteht ein Zyklus (der Debugging-Zyklus), wie in Bild 3.11a zu sehen ist. Die Zeit zwischen dem Starten der *n*-ten Version und der (*n*+1)-ten Version des Programmes sei "Zykluszeit" (*turn-around time*) genannt. Diese Zykluszeit beträgt in der Software in einer interaktiven Umgebung manchmal nur Minuten.

**Durchlauf-Zeiten.** Einen ähnlichen Debugging-Zyklus (hier jedoch *prototyping* genannt) mit einer jedoch um Größenordnungen längeren *turn-around time* (vgl. Bild 3.11b) kann man im Entwurfsprozeß für integrierte Schaltungen bei etwa folgender Organisation identifizieren. Nachdem der Entwurf einer Schaltung in Form eines Layout vorliegt, wird diese hergestellt und dann anschließend getestet. Wenn hierbei Entwurfsfehler gefunden wurden, wird der Design der Schaltung und dann das daraus abgeleitete Layout unter teilweise neu

#### Die "Schnittstelle"

Die Zusammenarbeit zwischen Anwender und Hersteller von Schlüssel-ICs ist ein komplexes Problem, es gab sogar internationale Konferenzen eigens über dieses Thema [19].

1. Design nach Gefühl (vor 1980). Anfangs wurde der Design vielfach durch Technologen selbst durchgeführt, was bei den anfänglich niedrigen Integrationsdichten relativ unproblematisch war (vgl. Kasten "Das bißchen Design erledigen wir mit Links" auf Seite 56). Die Entwurfsregeln wurden informell gehandhabt, weshalb es immer wieder zu Unklarheiten kam, vor Allem mit externen Personen.

2. klare Schnittstelle (ab ca. 1980). Zunehmende Arbeitsteilung (vgl. Bild 2.4) verlangt präzise Entwurfsregeln zwecks Minimierung des Beratungs-Aufwandes. Da sich Computer-gesteuerte Pattern-Generatoren durchsetzten (vgl. Bild 9.20), wurden für Designs Datenaustausch-Formate vereinbart, wie CIF [13]. Die klare Schnittstelle ist Voraussetzung für die Zwischenschaltung sogenannter *silicon broker* wie der MOSIS-Dienst, der Empfang über das ARPA-Netz, Eingangsprüfung der Designs und Administration vollautomatisch abwickelt.

3. ICs aus dem Automat (ab 1997?). Noch werden die Schiffchen mit 20 bis 50 Wafers werden mit der Hand von Gerät zu Gerät transportiert, von Menschen, die in teuersten Reinst-Räumen herumlaufen. Lange Durchlaufzeiten (Wochen bis Monate) herrschen vor. Mit dem Technologie-Fortschritt werden die Fabriken immer teurer (Bild 9.12) und kosten bereits bis zu mehr als eine Milliarde US-Dollar. Es besteht Bedarf für eine Rationalisierung der Fertigung integrierter Schaltungen.

Neue Fabrikformen entstehen [23] [30] (vgl. Kasten "Chips aus dem Automaten" auf Seite 82). Jede Scheibe einzeln wird in einer Reinst-Kassette automatisch von Maschine zu Maschine transportiert und dort von Robotern eingelegt. Die Reinräume werden nicht für den Menschen benötigt. Der Computer plant den Durchlauf separat für jede einzelne Silizium-Scheibe: eilige Scheiben sind kein Problem mehr. Wir nähern uns dem Idealfall: Sie legen Ihren Datenträger in die Lesestation und warten bis die fertigen Chips ausgestoßen werden.

Eigenschaft bzw. Vorgang	full custom		(Standard-) Zellen-basiert	ASICs	
	Standard-Schaltungen	Anwender-spezifisch		mASICs	pASICs
Anwendungs-Know-how	Hersteller	Anwender			
Design-Auftrag					
Spezifikation					
Kosten-günstige Stückzahl	sehr hoch		hoch	gering	
Entwurfskosten	hoch		gering		
Design-Zeit	lang		kurz		
turn around			lang	kurz	sehr kurz
Personifizierung: Definition	Hersteller	Anwender			
Realisierung	Hersteller			Anwender	
Grundstruktur	frei		CAE-Softw.	Hersteller	
Zahl Anwender	hoch	niedrig			

Bild 3.9: Detaillierte Gegenüberstellung v. Vollkundschtaltung, Standard-Zellen und ASIC.

tem Nachvollziehen des Entwurfsprozesses korrigiert. Die so entstandene neue Version wird wiederum hergestellt und anschließend getestet: somit liegt also ein Prototyping-Zyklus vor. Im Gegensatz zur Software treten hier nun völlig andere Zeitverhältnisse auf. Anstatt Minuten oder Stunden dauert die "turn-around time" meist mehrere Monate: bei den für Europäische Universitäten hergestellten Multiprojekt-Chips zum Beispiel typischerweise 6 bis 9 Monate. Wie unten erwähnt wird, ist die Länge dieses Zyklus sehr kritisch für die Wettbewerbsfähigkeit, weshalb zwei Strategien angestrebt werden: entweder rapid prototyping (sehr schnelle Prototypen-Herstellung [19][12]) durch geeignete Hersteller-seitige Infrastrukturen, oder *correct first silicon* durch sehr sorgfältige Entwurfs-Durchführung in Verbindung mit hochwirksamer Verifikation in hochwertigen Entwurfs-Umgebungen.

**Correct First Silicon.** Es sollte daher die Schaffung geeigneter Entwurfsumgebungen dringend angestrebt werden, sodaß der Design der ersten Version einer Schaltung fehlerfrei ist (*working first silicon*), damit der Design-Prozeß (Bild 3.11c) nur einmal durchlaufen werden muß. Das heißt über Simulatoren und andere CAD-Werkzeuge (*CAD-Tools*) muß man in der Lage sein, praktisch den kompletten Lauf der Schaltung erschöpfend zu simulieren oder sonst-

Vorgang	full custom		(Standard-) Zellen-basiert	ASICs	
	Standard-Schaltungen	Anwender-spezifisch		mASICs	pASICs
Personifizierung: Definition	Hersteller	Anwender			
Realisierung	Hersteller			Anwender	
Grundstruktur	frei		CAE-Werkzeug	Hersteller	

Bild 3.10: Gegenüberstellung der Arbeitsteilung bei Vollkundschtaltungen und ASICs.

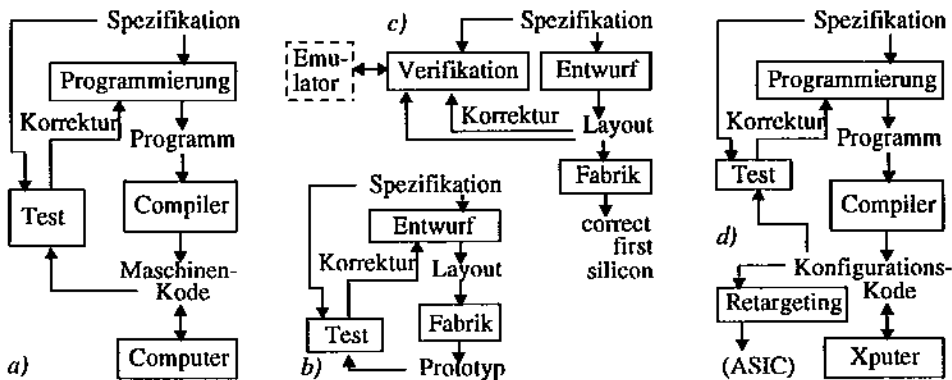


Bild 3.11: Prototyping: a) bei Software, b) per Prototyp, c) per CAD, d) per Xputer-Paradigma.

wie zu verifizieren bevor der erste Prototyp hergestellt wird.

Das heißt: die physisch noch nicht existierende Schaltung wird durch eine Art *virtual reality* per Computer voll zum Leben erweckt und erschöpfend überprüft. Dazu ist eine enorme Rechenleistung erforderlich, wozu stets mindestens die leistungsfähigste Version der neusten schnellsten Workstation benötigt wird. Die Workstations veralten schnell, da sich die Komplexität der Schaltungen und auch die Funktionalität der CAD-Software rasch weiterentwickelt. Der Bedarf an Rechenleistung steigt überproportional.

Ein Schaltungs-Test mit dem Ziel der Lokalisierung von Entwurfsfehlern ist ohnehin schwieriger als der Chip-Test in der Fabrik. Letzterer hat nur den Sinn herauszufinden, ob ein Herstellungsfehler vorliegt oder nicht, wobei der Test bereits nach der ersten Entdeckung eines Fehlers sofort abgebrochen wird (da dieser Chip dann weggeworfen wird). Ein solcher "go/no go-Test" eignet sich nur sehr schlecht zum Herausfinden von Entwurfsfehlern.

### ASICs nur für Hardware-Leute?

Der Umgang mit ASICs ist ein Bildungsproblem. Elektrotechniker ohne etwas Informatik-Background und Informatiker, die sich nicht in Technischer Informatik (Hardware) vertieft haben, tun sich hier schwer. Deshalb findet man hier in Praxis und Forschung fast nur Hardware-Leute, gleich von welcher Seite. Dies gilt auch für die neu Wissenschaft des Hardware-Software-Ko-Design [35] [16]. In der Produkt-Entwicklung sind zwei Leute (aus Software und Hardware) oft zu teuer (vgl. Kasten "Lean Engineering" auf Seite 61).

Wie sollen Software-Leute heutiger Art ein ASIC als Akzelerator einsetzen, wenn sich z. B. eine Programm-Schleife als kritischer Durchsatz-Engpaß herausstellt? (Kapitel 19 zeigt z. B. die Umwandlung einer Sortier-Schleife in eine IC mit millionenfachem Durchsatz-Gewinn. Kapitel 20 zeigt weitere Beispiele von "Hardware statt Software".) Hardware-Software Ko-Design zeigt Wege zur Innovation des universitären Studien-Angebots zur Ausbildung der Hardware/Software(-Ko)-Ingenieure und -Informatiker der Zukunft. Was machen wir jedoch mit den "alten Hasen", die bereits vor 10 Jahren die Universität verlassen haben? Ein Weg führt über neue Maschinen-Paradigmen [3] wie das Xputer-Paradigma [18], das die Effizienz von ASICs mit der Vertrautheit der Programmiersprachen verbinden kann (vgl. Kasten "Xputer: ASIC für Software-Leute" auf Seite 80).

### Xputer: ASIC für Software-Leute

Der in Kaiserslautern entwickelte Xputer ist die flexiblere Alternative zum (von-Neumann-)Computer [17] [18] [20]. Xputer haben keinen Befehlszähler, sondern stattdessen einen (oder mehrere) Datenzähler. Dabei entfällt die beim Computer sehr feste Kopplung zwischen Steuerwerk und Rechenwerk (ALU), wodurch rASICs statt ALU anwendbar sind<sup>1</sup>. Der Xputer ist somit ein Zwitter aus ASIC und prozeduraler Maschine. Dies kommt der Vertrautheit des Software-Ingenieurs mit Maschinen-Paradigmen entgegen. Von diesen leicht erlernbare Programmiersprachen wurden entwickelt. Ein Compiler ist in Arbeit. Doch es ist noch viel zu tun.

Bei einer großen Klasse kommerziell wichtiger Algorithmen (Signalverarbeitung, Bildverarbeitung, wissenschaftliches Rechnen u. a.) sind Xputer oft um ein bis zwei Größenordnungen effizienter als der Computer [18]. In einem Einzelfall (Design Rule Check, vgl. Kapitel 12) wurden experimentell sogar mehr als 3 Größenordnungen ermittelt [18]. Der Xputer ist als Akzelerator also durchaus eine leistungsfähige Alternative zum ASIC. Der Entwurf wird ebenfalls beschleunigt, da die sehr zeitaufwendige Simulation (vgl. a. Kapitel 14) oder Emulation durch die sehr effiziente "Exekution" ersetzt wird.

Bis zu einer Kommerzialisierung ist noch viel zu tun. Neben Compilern wird eine Flächen-effiziente rALU benötigt, die ohne ein gutes industrielles Design-Team kaum machbar ist. Leider haben wir für Xputer keine Drittmittel zur Verfügung (mit weniger innovativen Ideen war mein Lehrstuhl in der Vergangenheit erfolgreicher). Ein ESPRIT-Antrag um 1988 wurde abgelehnt. Die Gutachter hatten wohl zu wenig Background auf dem durch Xputer betretenen Neuland. Eine sehr aufwendige Suche nach Kooperationspartnern in der Europäischen Industrie schlug fehl. Auch hier herrscht wohl der IH-Effekt (s. a. S. 512). Aber nun arbeitet man in Japan über Xputer. Dies erinnert mich an Lotfi Zadeh (Professor emeritus an der UC Berkeley), der 25 Jahre lang belächelt und gelegentlich sogar beschimpft wurde bis seine *fuzzy logic* [6] [7] [24] aus Japan rückkehrend ein Welterfolg wurde. Beim Xputer rechne ich aber mit weniger als 10 Jahren insgesamt. Auch werden wir nicht belächelt ("awards" 1991/1992 in Deutschland und USA).

1) anpaßbares Rechenwerk "rALU" (reconfigurable ALU).

Somit muß dringend ein Debugging-Zyklus angestrebt werden, der ohne Fabrikation und ohne den damit verbundenen Prototypen-Test auskommt. An die Stelle des Test tritt die Verifikation (Bild 3.11c), die eine Entwurfsüberprüfung auf der Basis des Layout und anderer Entwurfsdaten zum Gegenstand hat (s. a. [9]).

**Verifikation.** Der Begriff der Verifikation wird gelegentlich als Sammelbegriff für alle denkbaren Formen der Entwurfs-Überprüfung (außer Prototypen-Test) gebraucht. Hierzu gehören die Simulation, die Emulation und die formale Verifikation [9] (vgl. Kapitel 14). Die formale Verifikation ist als eine spezielle Richtung der Entwurfs-Wissenschaften die große Hoffnung der Szene. Von der formalen Verifikation wird die Überwindung des Unmengen von Personalkapazität verschlingenden Moloch "Simulation" erwartet. Das Gebiet findet sich noch im Stadium der Grundlagenforschung. Europas Universitäten sind führend. Aber leider besteht bei uns die fatale Tendenz, die Grundlagenforschung abzuwürgen (siehe Abschnitt 1.6).

**Emulation.** Unter Verwendung von rASICs wurden Simulations-Akzeleratoren entwickelt, die auch *ASIC-Emulatoren* genannt werden. Bekanntester Anbieter ist derzeit Fa. Quickturn (USA). Solche Geräte von Schrank-Größe können über eine mitgelieferte Schnittstelle an die gängigsten Workstations angeschlossen werden. Aus der beim Entwurfsprozeß entstandenen Netz-



liste (Design-Datenstruktur) durch einen von Quickturn mitgelieferten Compiler die nötigen Konfigurations-Files erstellt werden. Bei einem komplexen Design dauert die Kompilation bis zu einem Tag. Dafür dauert hier die Emulation statt Wochen (wie bei der Simulation) nur Tage.

**Xputer.** Ein Vorschlag zur ASIC-Verifikation ist die Verwendung des Xputer-Paradigma [18]. Hierbei wird das Entwurfs-Problem aus einer höheren Programmiersprache in den Konfigurations-Kode für einen Xputer übersetzt (vgl. Kasten "Xputer: ASIC für Software-Leute" auf Seite 80).

An die Stelle von Simulation oder Emulation tritt die direkte Exckution (Schema ähnlich Bild 3.11 a), um Größenordnungen effizienter als selbst die Emulation. Wenn erforderlich, kann durch Retargeting (vgl. Spalte "Technology Mapping" in Bild 3.7) aus dem Konfigurations-File der File für die mASIC-Fertigung automatisch generiert werden (Schema nach Bild 3.11 d).

### 3.6 Was ist zu tun?

Bild 3.12 faßt die wichtigsten Aspekte beim ASIC-Design zusammen, die für die Wettbewerbsfähigkeit durch den Designer und das Design-Management beachtet werden müssen. Die Struktur und Dynamik der Nachfrage und der Charakter des Anwendungsgebietes müssen voll begriffen werden. Leistungsfähige Entwurfswerkzeuge zusammen mit zu deren geschicktem Gebrauch hochqualifiziertem Personal sind ein Schlüssel zum Erfolg. Wege zur Wirtschaftlichkeit unter alternativen Infrastrukturen müssen beherrscht werden. Produkt bzw. Produkt-Familie müssen durch Design und Umgang mit Infrastrukturen der Dynamik des Marktes entsprechen. Alle Beteiligten brauchen ein komplexes Spektrum neuartiger Qualifikationen. Eine Ursache der Mißerfolge Deutschlands ist die mangelnde und zu langsame Verbreitung solcher Qualifikationen am Arbeitsmarkt.



Bild 3.12: Randbedingungen der Produktentwicklung bei integrierten Schaltungen.

### Risiko-scheues Management

Auf dem rASIC-Markt ist kein deutscher Hersteller vertreten und nur ein einziger Europäer (Algotronix). Diese "Spielerei" wurde lange unterschätzt bis ab ca. 1988 dann der Umsatz rapide anstieg (derzeit ca. eine Milliarde US-Dollar). Man hatte wohl die Diffusionszeit nicht berücksichtigt. Bei an sich sehr interessanten Innovationen dauert es oft Jahre, bis diese sich herumgesprochen haben. Erst nach einiger Zeit, zumal wenn inzwischen das Produkt besser geworden ist, erreicht die Diffusions-Lawine eine sichtbare Größenordnung.

Ein Beispiel ist der Macintosh. Apple dachte, die Lisa sei nur wegen des hohen Preises ein Flop gewesen. Man baute eine überdimensionierte Fabrik für den Macintosh und verkaufte dann enttäuschend schlecht. Die neue Benutzer-Oberfläche<sup>1</sup> war zwar ungeheuer attraktiv (wurde deshalb später von der Konkurrenz nachempfunden: Windows, X-Windows, OSF/Motif u. a.). So etwas muß sich jedoch erst einmal herumsprechen; das braucht Zeit. Die Verkaufszahlen stiegen dann später rasch.

Die Diffusionszeit-Abschätzung ist unsicher. Das Risiko ist groß. Es muß mit flexiblem Marketing geantwortet werden, das sich laufend der beobachteten Markt-Entwicklung anpaßt. Die Fertigungskapazität und Produktfamilie muß auf einer flexiblen Strategie der schnellen Anpaßbarkeit beruhen. Risikofreudigkeit allein führt jedoch leicht zum Scheitern.

1) Vorbild: ALTOS-Rechner im Xerox PARC

### Chips aus dem Automaten

Die Fabrikation integrierter Schaltungen ist immer teurer geworden. Nur mit finanzkräftigen Partnern konnte Texas Instruments (TI) in den letzten 4 Jahren Großanlagen klassischen Typs für je 1,2 bis 1,5 Milliarden US-Dollar in Italien, Taiwan, Japan und Singapur bauen. Selbst auf viel billigeren Anlagen wurden die Fabrikationskosten für geringere Stückzahlen nur mit organisatorischen Tricks, wie Umlageverfahren (MPCs) oder Vorfabrikation (Gate Arrays) einigermaßen tragbar.

**Slow turn-around.** Die reinen Fabrik-Zykluszeiten betragen 30 Arbeitstage und länger. In der kommerziellen Praxis betragen die Lieferzeiten 4-6 Wochen für Gate-Arrays und 3-4 Monate für Vollkunden-Schaltungen. Beim ersten MPC des E.I.S.-Projekt betrug sie 9 Monate. Der Student, der die Schaltung entwarf, steht dann zum Testen der Schaltung (was wichtig wäre für die Qualität der Lehre) nicht mehr zur Verfügung wegen Prüfungen.

**Single Wafer Fab.** Mit einer völlig neuartigen Chip-Fabrik<sup>1</sup> in Austin in Texas will TI die Halbleiterproduktion künftig wesentlich wirtschaftlicher gestalten [30]. Statt 200 Scheiben gleichzeitig in Rohröfen soll per Rapid Terminal Processing (RTP) jeweils nur ein einziger Wafer aufgeheizt werden. Die Scheiben werden nicht mehr in teuersten Reinst-Räumen mit der Hand von Station zu Station getragen und oft langfristig zwischengelagert, sondern in kleinen Vakuum-Kassetten einzeln automatisch durch den Prozeß transportiert.

1) 86 Mio US-Dollar Entwicklungskosten unter Förderung durch die ARPA (Verteidigungsministerium)

**Software teurer als der Gerätepark.** In dieser Single Wafer Fab kann jeder Wafer eine individuelle Behandlung erfahren. Dies wäre bei der klassischen Wafer-Fabrik völlig unmöglich, da Ein- und Umstellungen so extrem zeitraubend sind, daß solche oft nur nach Monaten unveränderter Produktion in Frage kommen. Möglich wird diese Single-Wafer-Fab nur durch aufwendige CIM-Softwarepaket<sup>2</sup> Works. Jede Scheibe hat ihren individuellen Fahrplan, evtl. als Eil-Scheibe alle anderen Scheiben überholend. Man rechnet, daß Works auch in anderen Branchen zum Standard werden kann. Wenn es den komplexen Mikro-Chip-Prozeß beherrscht, kann es auch die Fertigung von Modems, Arzneimitteln oder Autos steuern [30].

**Stärkung der amerikanischen Industrie.** Während Speicherchips im Laufe ihres Lebenszyklus, im Preis schwankend wie in Schweinezyklen, zeitweise wie *jelly beans* (Gummi-Bärchen) oder Schaufelware verschleudert werden (vgl. Bild 1.20), sind bei Logik-Chips die Wertschöpfung größer, die Preise stabiler und die Erträge besser [30]. Bei Logik-Chips, den Schaltungen mit Intelligenz für komplexe Aufgaben, sind die Innovationszyklen kürzer, Kunden-Ansprüche differenzierter, und die Serien kleiner. Robert Doering (Chef des MMST-Programms<sup>3</sup> bei TI): "Mit der wirtschaftlichen Fertigung kleinster Serien von Logik-Chips kann die amerikanische Halbleiterindustrie ihre technische Führung behaupten."

2) Computer-Integrated Manufacturing

3) Microelectronic Manufacturing Science and Technology

**Marketing wie bei Waschpulver?** Zum Marketing von Waschpulver muß man nicht unbedingt Chemiker sein. Die Hausfrau kann ohnehin kaum prüfen, welches Waschpulver besser ist. Das Produkt selbst ist in solchen Branchen oft nicht so wichtig. Oft ist sogar die Werbung und Verpackung teurer als das eigentliche Produkt selbst. Bei innovativen komplizierten High-Tech-Produkten geht dies so nicht. Das Marketing für Innovationen ist viel schwieriger (vgl. Kasten "Risiko-scheues Management" auf Seite 81). Produktplanung und Marketing kann hier nicht von Außenstehenden ohne den Ingenieur konzipiert werden [8]. Wenn Anwendungs-spezifische ICs für eine bestimmten Marktnische gebraucht werden, sind die Anforderungen der eventuellen Kundschaft oft sehr kompliziert. Die Auswahl der geeigneten Infrastruktur ist eine weitere Komplikation, wobei die Markt-Entwicklung evtl. später auch noch einen Wechsel der Technologie-Plattform erfordern



kann. Bei allen technischen Überlegungen, die während der Produkt-Planung und -Entwicklung durchgeführt werden, müssen stets alle Aspekte der Vermarktung, der Stückzahlen-abhängigen Kosten und der Möglichkeiten zur flexiblen Reaktion auf den Markt berücksichtigt werden.

**Infrastruktur Universität.** Dies ist eine enorme Herausforderung der Lehre. Laufend müssen die neusten Qualifikationen schnellstens verfügbar gemacht werden. Anforderungen an die Lehre sind vielseitig

(vgl. Kasten "Lean Engineering" auf Seite 61) und schwierig, da wegen der raschen Entwicklung zum Teil noch kein Lehrmaterial existiert. Dazu müssen alle Infrastrukturen reichhaltig verfügbar und ohne Hürden zugänglich sein: Prototypen-Fertigung, neuste CAD-Umgebungen, Workstations, und vor Allem: reichlich hochqualifiziertes Betreuungs-Personal, denn all die o. g. Fähigkeiten sind nur durch intensive bestausgerüstete Praktika, Studien- und Diplom-Arbeiten zu vermitteln.

Beispielsweise die diskutierte Kürzung der Diplomarbeits-Dauer von einem halben Jahr auf 3 Monate ist hier nicht hilfreich, womit wir wieder beim Thema Politik wären. Die ständige öffentliche Diskriminierung der Professorenschaft ist beispiellos: keine andere Beamtengruppe hat so etwas erlebt. Es ist nicht wahr, daß die Professoren faul sind und nur ihren Hobbies nachgehen (vgl. Kasten "Professoren auf die Schulbank" auf Seite 57). Auch die meisten der ständig wechselnden Therapie-Vorschläge zum Patient Universität sind in ihrer Inkompetenz ungeheuer demotivierend. Dies erinnert manchmal an Eunuchen: die wissen immer, wie man's macht.

**Förderung aus Bundesmitteln.** Die universitäre Selbstverwaltung kann kaum Schwerpunkte bilden und so das Problem nicht lösen, weder das Betreuungs-Problem, noch das Ausstattungsproblem (vgl. Kasten "Qualität der Lehre" auf Seite 63). Deshalb muß hier eine großzügige Förderung im Rahmen der Wirtschaftspolitik für den Standort Deutschland erfolgen, und zwar kontinuierlich unter allererster Priorität (vgl. Kasten "Management by Potatoes" auf Seite 64). Auch die Forschung muß gefördert werden, damit das Lehrpersonal der raschen Entwicklung der Entwurfswissenschaften überhaupt folgen kann und nicht rasch inkompetent wird. All dies zusammen ist eine so ungeheure Herausforderung, daß auf keinen Fall Hürden vor die Erlangung der Mittel stehen dürfen [31] (vgl. Kasten "Die Schrittmacher-Rolle der Universitäten" auf Seite 65). Hier liegt keine Subventionierung vor. Es handelt sich ganz eindeutig um Infrastrukturen im öffentlichen Interesse, ja, um den Schlüssel zu unserer Wettbewerbsfähigkeit.

### 3.7 Literatur

- [1] M. Bolton: Digital Systems Design with Programmable Logic; Addison-Wesley, 1990
- [2] S. Brown, R. Francis, J. Rose, Z. Vranesic: Field-Programmable Gate Arrays; Kluwer, 1992
- [3] A. Buell, K. L. Pocek: Proc. IEEE Workshop on FPGAs for Custom Computing Machines, April 5-7, 1993, Napa, CA; IEEE Computer Society Press, Los Alamitos, 1993
- [4] D. Cohen: MOSIS - User Interface for Silicon Foundries; in: [19]

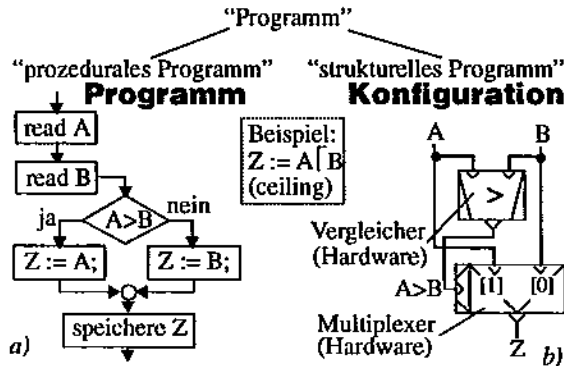


Bild 3.13: a) Software- gegen b) Hardware-Realisierung.



- [5] V. Coli: FPGA design technology; *Microprocessors and Microsystems* 17,7 (July 1993)
- [6] B. Cosco: *Neural Networks and Fuzzy Systems*; Prentice-Hall, 1992
- [7] E. Cox: *Fuzzy Fundamentals*; *IEEE Spectrum*, Oct. 1992
- [8] W. H. Davidow: *High Tech Marketing*; Campus Verlag 1987
- [9] H. Eveking: *Verifikation digitaler systeme*; Teubner, 1992
- [10] J. Green, E. Handy, S. Beal: *Antifuse Field-Programmable Gate Arrays*; *Proc. IEEE* 81,7, Jul '93
- [11] H. Grünbacher, R. Hartenstein: *FPGAs - Architectures and Tools for Rapid prototyping*; Springer-Verlag, Heidelberg, 1993
- [12] R. Hartenstein: *VLSI-Bausteine in geringen Stückzahlen für Spezialanwendungen*; *Elektronische Rechenanlagen* 22 (1980), Heft 4
- [13] R. Hartenstein: *Shared Cultures: CIF Library, Starting Frames and Scalable Design Rules*; [22]
- [14] R. Hartenstein: *Ausbruch der E.I.S.-Zeit*, *Computer Magazin*, März 1986
- [15] R. Hartenstein: *KARL and ABL*; in (ed.: J. P. Mermet): *Fundamentals and Standards in Hardware Description Languages*; Kluwer Academic Publishers, Boston, 1993
- [16] R. Hartenstein: *Hardware/Software Co-Design*; *Proc. 3rd Int'l Workshop on Field-programmable Logic and Applications*, Oxford, Sept. 1993
- [17] R. Hartenstein et al.: *Xputers: a new R&D area*; Bericht, Universität Kaiserslautern 1990
- [18] R. Hartenstein: *A Novel ASIC Design Approach Based on a New Machine Paradigm*; *IEEE Journal of Solid-State Circuits*, J-SSC 26, 7 (July 1991)
- [19] R. Hartenstein, K. Woelcken: *Proc. European Conf. on Customer / Vendor Interfaces in Microelectronics (EURO CVIM)*, Kaiserslautern 1986; GMD, St. Augustin, 1987
- [20] R. Hartenstein, V. Moshnyaga: *Xputers*; IT Press, Bruchsal / Chicago (in Vorbereitung)
- [21] C. Jay: *VHDL and synthesis tools provide a generic design entry platform into PPGAs, PLD and ASICs*; *Microprocessors and Microsystems* 17,7 (July 1993)
- [22] P. Jespers, C. Sequin, F. van de Wiele: *Design Methodologies for VLSI Circuits*; Noordhoff & -Stijthoff, Rockville, Maryland, 1981
- [23] D. K. Kahaner: *Completely automated assembly*; *IEEE Micro* 13,3 (Juni 1993)
- [24] G. Klear, T. Folger: *Fuzzy Sets, Uncertainty, and Information*, Prentice-Hall, 1988
- [25] C. Mead, L. Conway: *Introduction to VLSI Systems* Addison-Wesley, 1980
- [26] W. Myers: *Get to market faster with FPGAs*; *IEEE Micro* 13,3 (Juni 1993)
- [27] N. N. (ha): *0,1 µm mit Röntgenstrahlen*; *Markt & Technik* 27 / 1993 (2. Juli)
- [28] N. N.: *Eurochip Services (Academic)*; *VLSI Des. Train'g Action*; CEC, DG XIII, Sept 1993
- [29] N. N.: *Eurochip Services for Industry*; *VLSI Des. Train'g Action*; CEC, DG XIII, Sept 1993
- [30] W. Osel: *Die Objekt-orientierte Chip-Fabrik der Zukunft steuert sich selbst*; *Computer-Zeitung* Nr. 32/33 (12. August) 1993
- [31] S. Ramo: *America's Technology Slip*; John Wiley & Sons, New York 1980
- [32] J. Rose, A. El Gamal, A. Sangiovanni-Vincentelli: *Architecture of Field-Programmable Gate Arrays*; *Proc. IEEE* 81,7 (Jul 1993)
- [33] I. Ruge, H. Mader: *Halbleiter-Technologie*; Springer-Verlag, 1991
- [34] A. Sangiovanni-Vincentelli: *Synthesis Methods for Field-Programmable Gate Arrays*; *Proc. IEEE* 81,7 (Jul 1993)
- [35] A. Sedlmeier, K. Buchenrieder: *Handout 2nd Int'l workshop on Hardware / Software Codesign (Codes/CASHE'93)*, Siemens-AG, München 1993
- [36] S. Trimmerger: *A Reprogrammable Gate Array and Applications*; *Proc. IEEE* 81,7 (Jul 1993)
- [37] T. A. York: *Survey on field-programmable logic devices*; *Microprocessors and Microsystems* 17,7 (July 1993)