

Zeitschrift: Bulletin des Schweizerischen Elektrotechnischen Vereins, des Verbandes Schweizerischer Elektrizitätsunternehmen = Bulletin de l'Association Suisse des Electriciens, de l'Association des Entreprises électriques suisses

Band: 78 (1987)

Heft: 11

Artikel: Design-Methoden für digitale Schaltungen

Autor: Fichtner, W.

DOI: <https://doi.org/10.5169/seals-903868>

Nutzungsbedingungen

Die ETH-Bibliothek ist die Anbieterin der digitalisierten Zeitschriften. Sie besitzt keine Urheberrechte an den Zeitschriften und ist nicht verantwortlich für deren Inhalte. Die Rechte liegen in der Regel bei den Herausgebern beziehungsweise den externen Rechteinhabern. [Siehe Rechtliche Hinweise.](#)

Conditions d'utilisation

L'ETH Library est le fournisseur des revues numérisées. Elle ne détient aucun droit d'auteur sur les revues et n'est pas responsable de leur contenu. En règle générale, les droits sont détenus par les éditeurs ou les détenteurs de droits externes. [Voir Informations légales.](#)

Terms of use

The ETH Library is the provider of the digitised journals. It does not own any copyrights to the journals and is not responsible for their content. The rights usually lie with the publishers or the external rights holders. [See Legal notice.](#)

Download PDF: 19.11.2024

ETH-Bibliothek Zürich, E-Periodica, <https://www.e-periodica.ch>

Design-Methoden für digitale Schaltungen

W. Fichtner

Heute können moderne digitale Schaltungen mit einigen hunderttausend Transistoren auf einem Chip integriert werden. Ein erfolgreicher Entwurf von derart komplexen Schaltungen ist nur dann möglich, wenn strukturierte Design-Methoden verwendet werden. Dieser Beitrag gibt einen kurzen Überblick der derzeit gebräuchlichen Entwurfsmethoden für digitale integrierte Schaltungen sowie einen Ausblick über mögliche Entwicklungen in der nahen Zukunft.

On peut maintenant intégrer dans une puce des circuits numériques modernes avec quelques centaines de mille transistors. Un projet comportant de tels circuits complexes n'est possible qu'en appliquant des méthodes de conception structurées. Bref aperçu de celles actuellement usuelles pour projets de circuits numériques intégrés et des prochains développements possibles.

Adresse des Autors

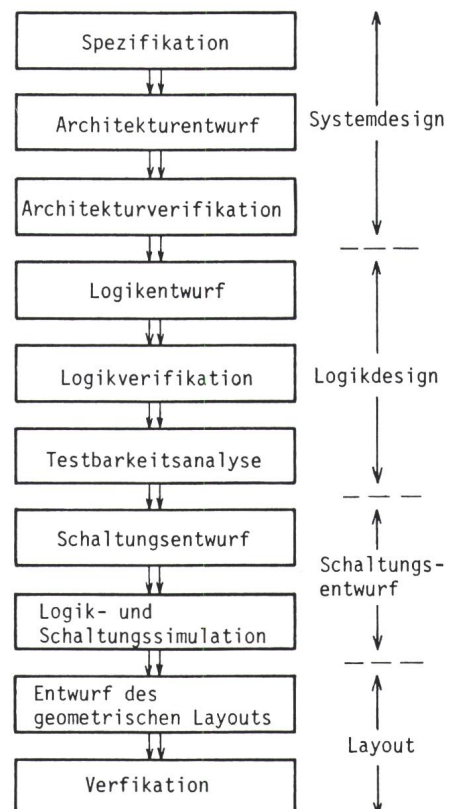
Prof. Dr. Wolfgang Fichtner, Institut für Integrierte Systeme, ETH-Zentrum, 8092 Zürich.

1. Einleitung

Die Mikroelektronik begann ihren Aufstieg Ende der 50er Jahre, als die ersten integrierten Schaltungen (ICs, Integrated Circuits) auf einem Chip erfolgreich hergestellt und in elektronische Systeme eingebaut wurden. Dies war der Start einer Entwicklung, deren Konsequenzen in alle Teile des täglichen Lebens reicht. Man denke nur an den Arbeitsplatzrechner oder den Personal Computer, die dem Anwender eine Rechenleistung bieten, die der eines Grossrechners der letzten Generation entspricht. Mehrere Komponenten haben zu dieser noch immer dynamischen Entwicklung beigetragen: verbesserte Fabrikationsmethoden (zunehmende Automatisierung), ein besseres Verständnis der einzelnen Prozessschritte sowie deren Wechselwirkung mit dem elektrischen Verhalten der Bauelemente.

Die heutige Si-Halbleitertechnik erlaubt die Integration von einigen hunderttausend Transistoren auf einem einzigen Chip. Dieses enorme Potential kann aber nur dann ausgenutzt werden, wenn die Kosten für die Entwicklung eines solchen VLSI-Bausteins auf einer vernünftigen Höhe gehalten werden können. Ein wesentlicher Anteil dieser Kosten wird durch den Entwurf der Schaltungen gebildet. So ist es bei VLSI-Projekten nicht ungewöhnlich, dass einige hundert Mannjahre investiert werden müssen. Der Schlüssel zur Bewältigung dieser Komplexität ist die Verwendung hierarchischer Designmethoden; sie erlauben, den Entwurf des Gesamtsystems in mehrere – leichter überschaubare – Teilprojekte aufzuteilen. Die Erfahrungen der letzten Jahre haben gezeigt, dass der Entwurf eines Chips mit 10^5 Transistoren ohne Hierarchie nicht mehr möglich ist. Eine zusätzliche Verminderung der Designkosten kann durch ein der Hierarchie des Entwurfsprozesses angepasstes CAD-System erreicht werden.

Der Entwurf einer integrierten digitalen Schaltung beinhaltet sämtliche wichtigen Einzelschritte, von der Spezifikation des Systemverhaltens bis zur Erstellung der für die Herstellung wichtigen Strukturdaten. Die Figur 1 zeigt die verschiedenen Abschnitte eines Entwurfs, die im Rahmen eines Projektes durchlaufen werden. Das Systemdesign umfasst die funktionelle und strukturelle Definition der integrierten Schaltung, während Logik- und Schaltungsdesign die Umsetzung in elektronische Gatter und deren korrekte elektrische Funktion zum Ziele haben. Der Layoutentwurf beinhaltet die Konstruktion der für die Chipfabrikation wichtigen geometrischen Informationen.



Figur 1 Die verschiedenen Phasen des IC-Entwurfs

Historisch ist die Entwicklung von rechnerunterstützten CAD-Methoden eng mit den Fortschritten in der Halbleiterfertigungstechnik gekoppelt. Schon bald nach der Realisierung der ersten ICs wurde der rechnerunterstützte Schaltungsentwurf (CAD) integraler Bestandteil eines Entwurfssystems. Die 60er Jahre waren durch die Entwicklung von Verifikationstechniken und Verfahren zur Bauelemente- und Schaltungssimulation geprägt. Mit der Realisierung komplexerer Schaltungen wie des ersten Mikroprozessors von Intel im Jahre 1971 wurde bald der Ruf nach computerunterstützten Layoutwerkzeugen (damals Digitizer genannt) laut. Die ersten Ansätze zur Entwicklung einer Designkonzeption fallen ebenfalls in diesen Zeitraum.

Die zweite Hälfte der 70er Jahre begann mit Versuchen, die Phasen Logik- und Schaltungsdesign hierarchisch miteinander zu verknüpfen (Fig. 1). Durch die Existenz von Mini-Computern war es auch möglich geworden, interaktive CAD-Tätigkeiten vorzunehmen. Auch bei den Entwurfsmethoden gab es grosse Fortschritte. Die noch heute populären *Semi-Custom*-Verfahren wie Gate-Array- oder Standardzellentechnik fanden in diesen Jahren erstmals weite Verbreitung.

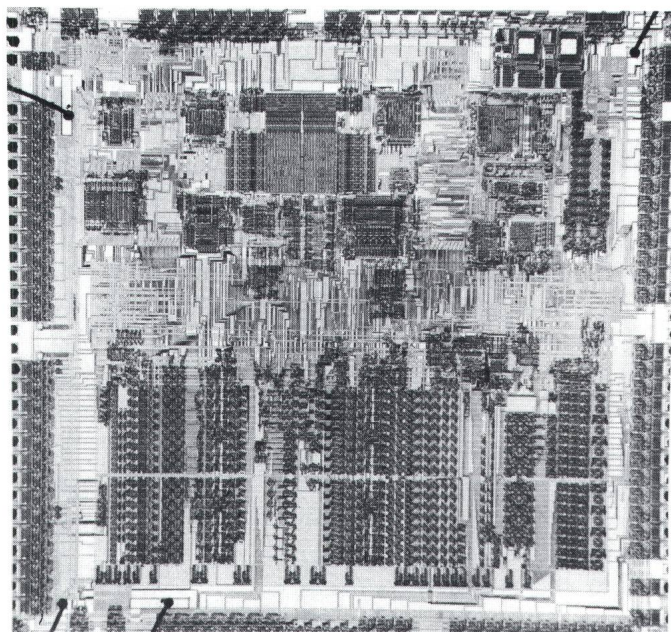
Die eigentliche CAD-Revolution startete aber erst zu Beginn dieses Jahrzehnts mit den ersten kommerziell erhältlichen Designsystemen. Der individuelle Anwender ist heute in der Lage, Schaltungen mittlerer Komplexität auf Arbeitsplatzrechnern selbst und ohne zu grossen Aufwand zu entwerfen. Im folgenden wird ein Überblick über die derzeit verwendeten Design-Methoden gegeben [1].

2. VLSI-Design-Methoden

2.1 Full-Custom-Entwurf

In dieser Methode ist der Design-Ingenieur nur an die von der Technologie vorgegebenen Randbedingungen, die sogenannten Design Rules (Entwurfsregeln), gebunden. Die Platzierung der verschiedenen Grundbausteine einer integrierten Schaltung ist völlig frei, man spricht daher in diesem Falle auch von Handentwurf oder manuellem Design. Der grosse Nachteil dieser Designmethode liegt in den grossen Kosten und langen Designzeiten. So ist es durchaus nicht ungewöhnlich, dass für einen in dieser Methode entworfenen VLSI-Chip mehrere

Figur 2
Chip-Fotografie
eines typischen
Full-Custom
VLSI-Chips



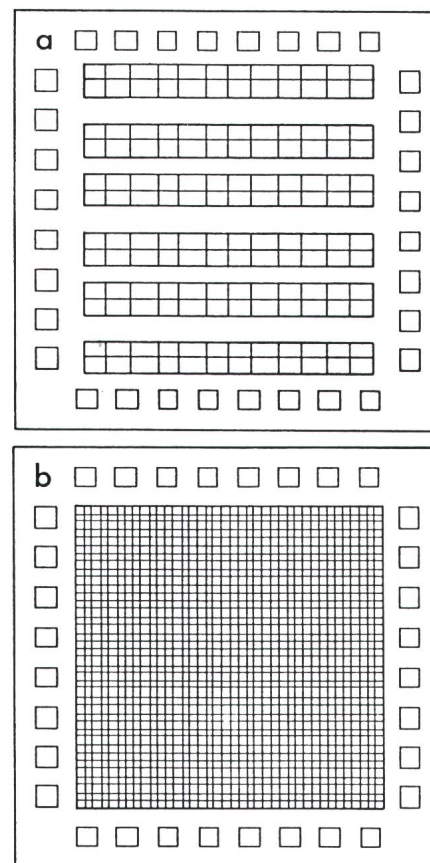
hundert Mannjahre benötigt werden. Warum wird Full-Custom dennoch verwendet? Die in dieser Methode offerierte Freiheit im Entwurf gibt dem erfahrenen Designer die Möglichkeit, das Layout eines Chips in optimaler Weise zu gestalten. Full-Custom-Chips zeigen damit auch meistens sehr gutes Verhalten bei minimalem Flächenbedarf. So gibt es heute wohl keinen *High Commodity*-Chip, der nicht im Full-Custom-Verfahren entworfen wurde. Typische Beispiele dafür sind moderne Mikroprozessoren.

Die Figur 2 zeigt als repräsentatives Beispiel die Chipphotographie eines Interface-Bausteines, welcher in der Microvax-II¹ für den Datentransfer, die Buskontrolle und das Error Handling verwendet wird [2]. Implementiert in einer 3- μ m-NMOS-Technologie, enthält dieser Chip mehr als 24 000 Transistoren. Ein anderes Beispiel für einen Full-Custom-Chip findet sich in [3].

2.2 Gate Arrays

Gate Array-Technologien bilden die gegenüberliegende Seite des Designspektrums. Ein vorfabrizierter Chip (Master) enthält ein reguläres Muster von Logikgattern. In einem letzten Prozessschritt wird nun eine bestimmte Digitalschaltung durch Verdrahtung

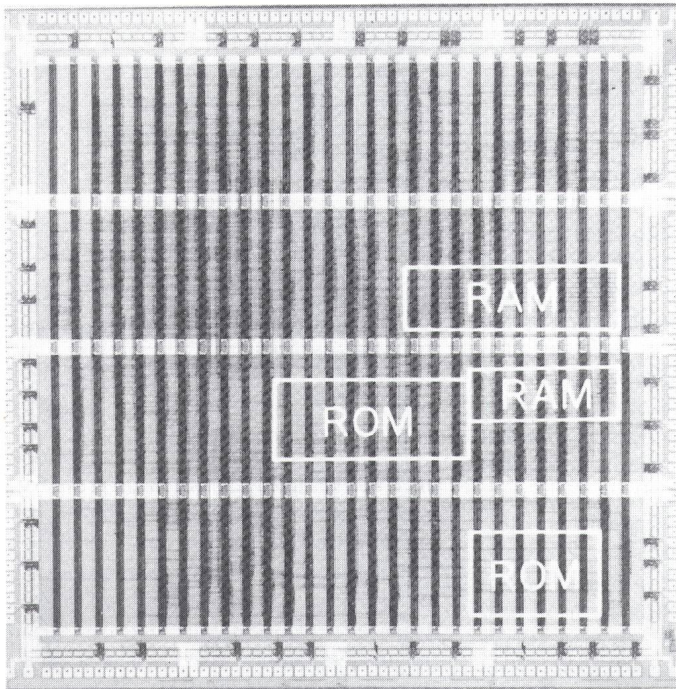
dieser Zellen implementiert. Die Figur 3a zeigt das Prinzip des *Gate Array*-Verfahrens. Der vorfabrizierte Master-Chip enthält bis zu 20 000 (20 K) äqui-



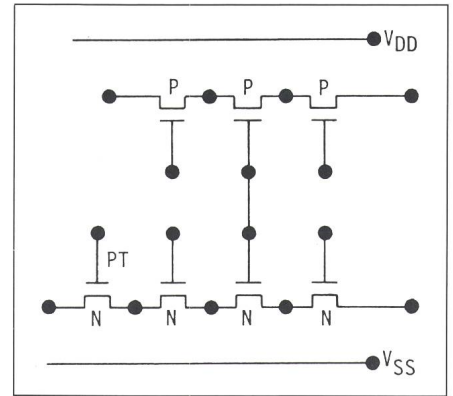
Figur 3 Gate Arrays

- a Schema eines Gate-Array Masters
- b Schema eines Sea-of-Gate Masters

¹ Arbeitsplatzcomputer von Digital Equipment Corporation



Figur 4 Gate-Array Chip



Figur 5 Transistorschema eines Sea-of-Gate Masters

PT Pass transistor

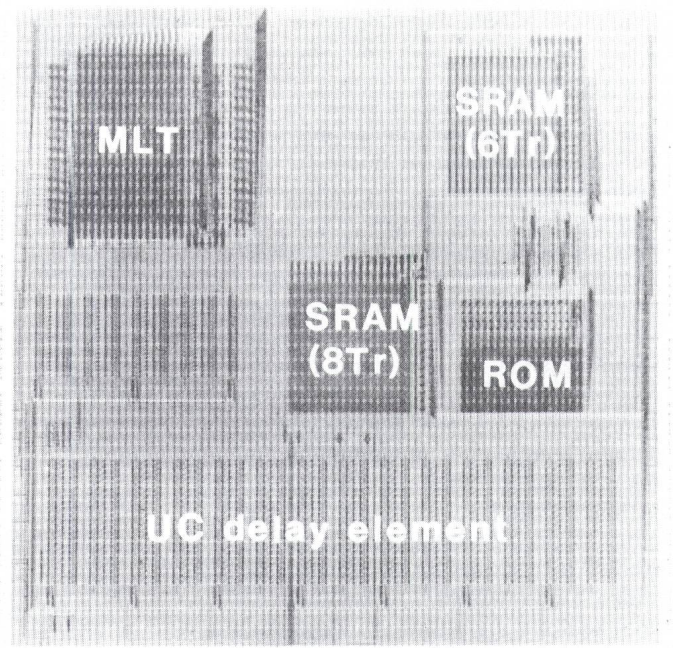
valente Gatter. Die reguläre Struktur des Masters wird durch eine reihenförmige Anordnung von Grundzellen (Core Cells) gebildet. Moderne Gate Array-Technologien erlauben sogar die Einbeziehung von analogen Blöcken oder Speicherelementen. Als Beispiel ist in Figur 4 das Photo eines komplexen Gate Array-Chips gezeigt [4]. Diese Schaltung realisiert einen komplexen Mikroprozessor mit RAM- und ROM-Komponenten. Das Gate Array enthält 864 Spalten und 28 Reihen von Grundzellen sowie 190 I/O-Bufferzellen und 24 Pads zur Spannungsversorgung. Die reihenförmige Anordnung der Grundzellen hat den Nachteil, dass beispielsweise sequentielle Funktionen mit mehr als 50 Gattern wegen der Platzbeschränkung der Verdrahtungskanäle nur schlecht zu implementieren sind. Weiter hat sich gezeigt, dass die Realisierung komplexer Funktionen zu langsamen Schaltungen mit grossem Platzbedarf ergibt.

Das sogenannte *Sea-of-Gates*- oder *Flex-Array*-Verfahren überwindet die soeben erwähnten Schwierigkeiten der herkömmlichen Gate Array-Technologien. Bei dieser Methode besteht der Master nur aus Grundzellen ohne Verdrahtungskanäle (Fig. 3b). Eine vernünftige Wahl der Grundzelle erlaubt, RAM, ROM und beliebige Logikzellen auf demselben Master Grid zu realisieren [5]. Diese Zelle bestimmt nun die Geschwindigkeit, das elektrische

Verhalten und die Ausnutzbarkeit². Eine typische Grundzelle ist in Figur 5 dargestellt. Sie besteht aus drei p- und n-Kanal-Transistorenpaaren sowie einem Transmissionsgate. Das Sea-of-Gates-Konzept erfreut sich zunehmender Beliebtheit. So bieten die meisten

² Unter Ausnutzbarkeit wird das Verhältnis der Anzahl verwendeten Grundzellen zu ihrer Gesamtzahl verstanden.

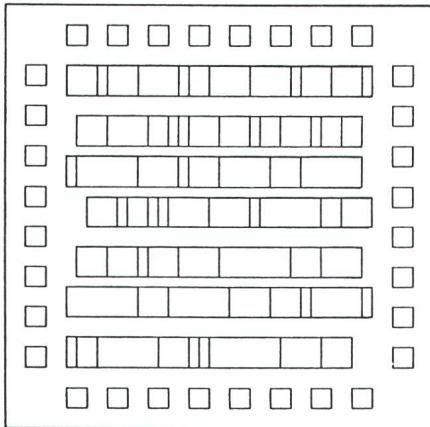
Figur 6 Sea-of-Gates Chip



CAD-Firmen Programme für diesen Designstil an. Die mögliche Komplexität dieses Konzeptes ist in Figur 6 angedeutet. Die Figur zeigt einen Testchip mit 240-K-Gattern, auf dem komplexe Schaltungen wie zwei verschiedene SRAM, ein ROM, ein Multiplizierer sowie eine Kontrollschaltung implementiert wurden [6].

2.3 Standardzellen

Die Standardzellentechnik verwendet in «Bibliotheken» gespeicherte Zellen, welche technologiespezifisch entworfen und mit Hilfe von CAD-Programmen im Detail charakterisiert werden. Während bei der Gate Array-Methode eine vorfabrizierte Matrix



Figur 7 Schema eines Standardzellen-Chips

nur verdrahtet wird, muss bei der Standardzellenmethode der komplette Technologiezyklus durchlaufen werden. Ein Grund für die Popularität dieses Konzeptes liegt darin, dass die Standardzellen eine normierende Wirkung ähnlich der CMOS- und TTL-Bausteinfamilien ausüben.

Das Schema eines typischen Standardzellenchips ist in Figur 7 dargestellt. Die einzelnen Zellen können dabei von verschiedener Höhe und Breite sein. Der Ausgangspunkt des Designprozesses ist in diesem Verfahren (wie bei den Gate Arrays) das Logikdiagramm der gewünschten Schaltung (die sogenannte Netlist). Während bei Gate Arrays nur eine Verdrahtung durchgeführt werden muss, ist hier durch die Verwendung von Bibliothekszellen sowohl eine Platzierung als auch eine Verdrahtung nötig.

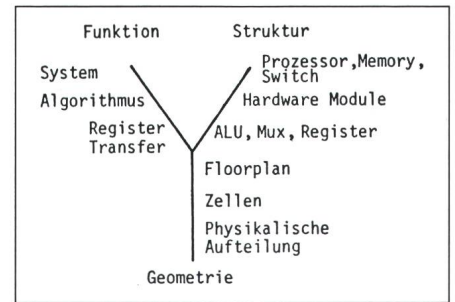
2.4 Makrozellen

Die Makrozellentechnik vereinigt die Vorteile der Full- und der Semicustom-Methoden. Das Prinzip dieser Technik liegt in der Ausnutzung der jedem System inhärenten Hierarchie (Fig. 8). Die Subblöcke bzw. Makrozellen können in verschiedenen Entwurfstechniken entstehen und aus Standardzellen, handentworfenen Blöcken oder auch strukturierten Blöcken wie PLA, RAM und ROM bestehen. Das Makrozellenkonzept findet insbesondere bei anwendungsspezifischen Schaltungen (ASIC, Application Specific Integrated Circuits) breite Anwendung. Verglichen mit der reinen Standardzellen- oder Gate Array-Methode benötigt es wesentlich kleinere Flächen.

2.5 Silicon Compiler und Assembler

Wie schon erwähnt wurde, offeriert nur die Full-Custom-Design-Methode optimale Ergebnisse für komplexe VLSI-Schaltungen. Die damit verbundenen hohen Kosten sind allerdings für Chips mit geringen Stückzahlen nicht tragbar. Die beim VLSI-Design auftretenden Probleme sind in gewissem Masse ähnlich zu jenen, die bei der Entwicklung grosser Softwareprojekte auftreten. Deshalb versucht man auch, bewährte Methoden der Softwareentwicklung und deren Hilfsmittel wie Editoren, Compiler und Assembler sowie Debugger in angepasster Form beim VLSI-Entwurf zu verwenden. Die bekanntesten Beispiele dafür sind Graphik-Editoren, Verifikationsprogramme, Simulatoren und Struktur assembler. Der etwas unglücklich gewählte Begriff des *Silicon-Compilers* fällt ebenfalls in diese Kategorie. Auch der zum gegenwärtigen Zeitpunkt feststellbare Trend zu *symbolischem Layout* (s. Fenster) hat mit der wachsenden Bedeutung von höheren Programmiersprachen eine Analogie im Softwarebereich.

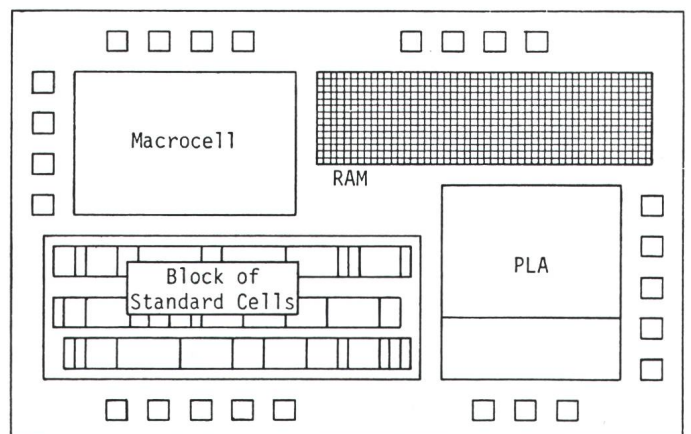
Der Trend zur Entwicklung *automatischer* Chipgeneratoren begann vor etwa zehn Jahren. Trotzdem sind diese bis heute nicht sehr erfolgreich gewesen. Die späten siebziger Jahre waren geprägt durch Versuche, den Chipentwurf in Analogie zur Compilation höherer Programmiersprachen durchzuführen. Inzwischen hat man gelernt, dass das automatische Entwerfen von integrierten Schaltungen wesentlich schwieriger als das Compilieren von Software ist. Die Gründe dafür sind, dass der Hardwareentwurf komplexer



Figur 9 Verschiedene Darstellungen des Hardwareentwurfes

und vielschichtiger ist und Transformationen zwischen verschiedenen Repräsentationen auf unterschiedlichen Abstraktionsebenen erfordert. Die Figur 9 bringt eine vereinfachte Darstellung der drei Bereiche *Funktion, Struktur und Geometrie*. Die Darstellung des funktionellen Verhaltens (Behavioral Domain) entspricht der Formulierung eines Algorithmus in einer höheren Programmiersprache. Sie enthält Deklarationen von Systemvariablen und Prozeduren, die das Systemverhalten als Funktion der Eingabedaten bestimmen, ohne Information über die Struktur zu enthalten. Auf den höheren Ebenen der Strukturrepräsentation beschreibt man das System als Netzwerk von logischen Grundbausteinen wie Registern, Multiplexern, ALU (Arithmetic Logic Units) und Kontrolleinheiten. Die dritte Darstellung beinhaltet die geometrischen Spezifikationen der verschiedenen Entwurfsebenen. Auch sie ist hierarchisch strukturiert; sie umfasst den Entwurf des einzelnen Transistors und seiner Abmessungen bis hin zum Layout des gesamten Chips.

Figur 8 Das Makrozellenkonzept



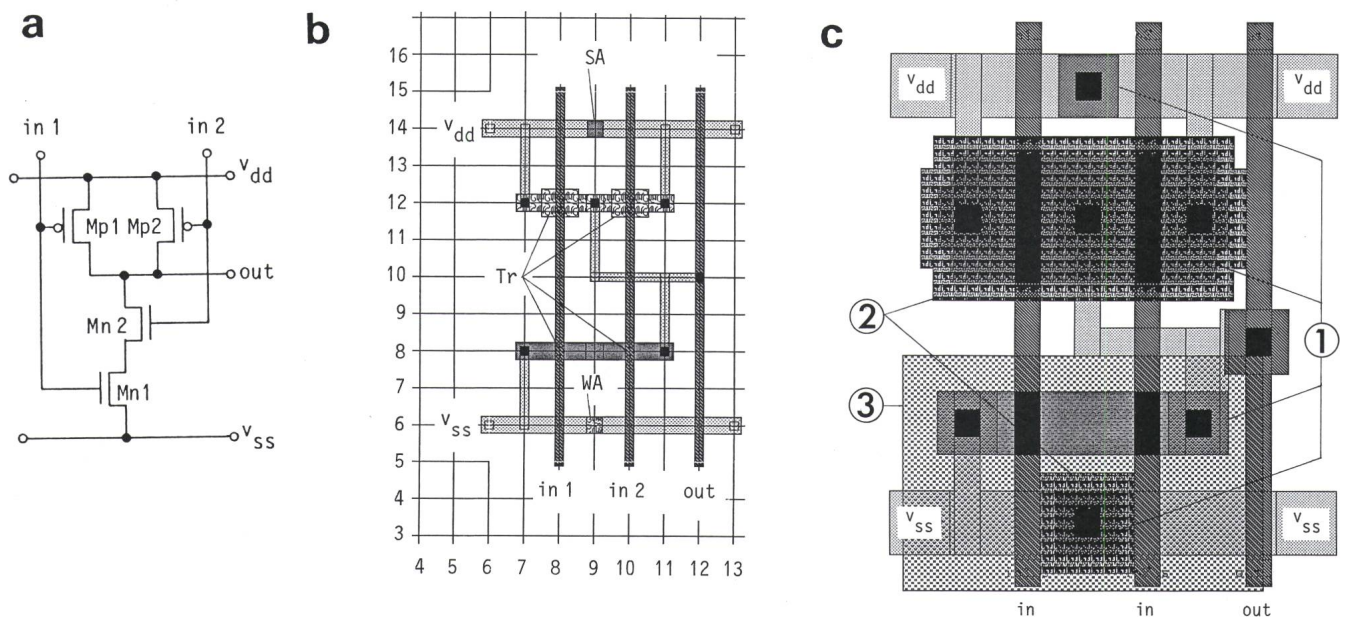
Symbolisches Layout

Mit der Programmierung in höheren Sprachen wird bekanntlich die Unabhängigkeit der Software von der Zielmaschine angestrebt und auch weitgehend erreicht. Beim Schaltungsentwurf mit symbolischem Layout ergibt sich eine gewisse Unabhängigkeit vom Zielprozess dadurch, dass das Layout zunächst nur als gegenseitige Anordnung von Elementen ausgelegt wird. Die Designregeln, welche die prozessbedingt einzuhaltenden Abmessungen und Abstände der Layoutstrukturen numerisch beschreiben, werden vom sog. Compactor automatisch berücksichtigt. Auch das Einsetzen von p-Wannen um die n-Kanaltransistoren herum, oder entsprechend umgekehrt in einem n-Wannenprozess, erfolgt selbsttätig. Der Designer wird insbesondere von der mühsamen Pflicht entlastet, während des Auslegens von Schaltungen ständig alle Designregeln des jeweiligen Herstellungsprozesses im Kopf

behalten und berücksichtigen zu müssen. Dadurch wird sowohl der Entwurf von Zellen als auch deren Wartung, d.h. Korrekturen und Anpassungen an neue Prozesse, wesentlich vereinfacht. Ein allfälliger Wechsel zu einer anderen Art von Technologie (z.B. von CMOS zu NMOS) bedingt allerdings nach wie vor einen Neuentwurf.

Die Figur a zeigt das Schema eines simplen 2-input-NAND-Gatters in CMOS-Technologie, die Figuren b und c dieselbe Schaltung vor und nach der Kompaktierung mit einem Virtual-grid-System. Wie der Name sagt, arbeitet dieses Werkzeug mit einem virtuellen Gitter. Das bedeutet, dass der Designer Layoutelemente auf einem vorgegebenen Gitter mit vorderhand noch unbestimmter Maschenweite plazieren muss. Beruhend auf graphentheoretischen Algorithmen werden bei der

Kompaktierung die Abstände benachbarter Gitterlinien unter Beachtung der Designregeln des Zielprozesses und der Verbindungseigenschaften (connectivity) des Netzwerks auf das erforderliche Minimum festgelegt. Gegebenenfalls können Gitterlinien bis zur Deckung zusammengeschoben werden (Abstand = 0); sie dürfen sich jedoch nicht überholen und werden auch nicht geknickt. Layoutelemente, welche sich im symbolischen Layout auf derselben Gitterlinie befinden, kommen also auch im Maskenlayout exakt untereinander resp. nebeneinander zu liegen, und zwar unabhängig davon ob sie elektrisch miteinander verbunden sind oder nicht. Das Zusammenschieben der Gitterlinien erfolgt in kommerziell erhältlichen Systemen heute noch meist separat, d.h. nacheinander in horizontaler und vertikaler Richtung.



Figur f1 Symbolisches Layout eines NAND-Gatters

a Schema in CMOS-Technik

b Symbolisches Virtual-grid-layout: Die aktiven Gebiete der MOS-Transistoren befinden sich überall dort, wo Polysilizium eine Diffusionsbahn kreuzt. Um ein symmetrisches elektrisches Verhalten des Gatters zu erzielen, müssen die p-Transistoren breiter dimensioniert werden als die n-Transistoren

SA Substratanschluss TR Transistoren (aktives Gebiet) WA Wannenanschluss

Metall Polysilizium n-Diffusion p-Diffusion Kontakt

c Maskenlayout für einen p-Wannen-Prozess nach der Kompaktierung. Die etwas von Figur fb abweichende Signatur der Diffusionsgebiete ist in diesem Zusammenhang belanglos.

① Dünnoxid ② Selektormaske ③ p-Wanne

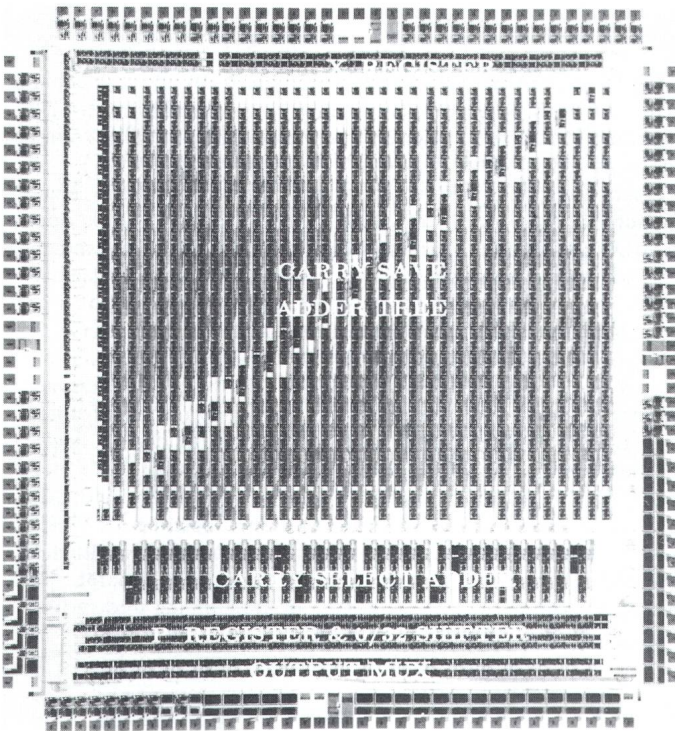
p-Diffusionsgebiet = Dünnoxid und Selektormaske

n-Diffusionsgebiet = Dünnoxid und nicht Selektormaske

Moderne CAD-Systeme erlauben dem VLSI-Designer nur in unzureichendem Masse, die verschiedenen Repräsentationen miteinander zu verarbeiten. Der typische Designer arbei-

tet heute mit CAD-Werkzeugen, die einen hierarchischen Entwurf kaum unterstützen. Meist verfolgt man dabei eine *Bottom-up*-Philosophie und steht so in direktem Gegensatz zu den Er-

fordernissen eines strukturierten Entwurfsablaufes. Als Ausnahmen sind hier die in letzter Zeit populär gewordenen Layout-Assembler und Modulgeneratoren zu nennen. Ein Layout-



Figur 10
16×16-bit-Multiplizierchip

Diese Schaltung wurde völlig automatisch von einem Silicon Assembler generiert.

Assembler erlaubt, grosse, bereits in geometrischer Form vorhandene Blöcke auf automatische Weise zu plazieren und zu verbinden. Ein Modulgenerator geht noch einen Schritt weiter: er unterstützt die Verwendung parametrisierbarer Zellen, die nicht in geometrischer, sondern in prozeduraler Form vorliegen. Modulgeneratoren sind besonders erfolgreich, wenn es um die Implementation von klar definierten und teilweise regulären Schaltungsstrukturen geht. Die Figur 10 zeigt die Abbildung eines 16×16-bit-

Multiplizierchips, der völlig automatisch durch einen Silicon Assembler generiert wurde. Die Qualität dieses Designs kann an Hand der gemessenen elektrischen Daten durchaus mit einem Full-Custom-Produkt verglichen werden [7].

3. Trends für die unmittelbare Zukunft

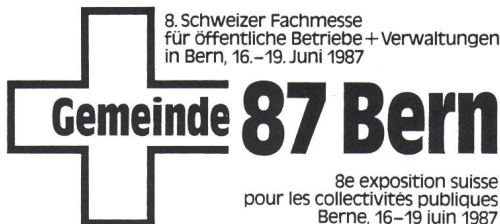
Es beginnen sich schon jetzt einige Trends klar abzuzeichnen. Das Konzept des Sea-of-Gates-Entwurfes offe-

riert dem ASIC-Designer beim Entwurf von Prototypen viele Vorteile. Wenn die erforderlichen CAD-Programme weitere Verbreitung finden, wird auch das Makrozellenkonzept an zunehmender Popularität gewinnen. Insbesondere die Kombination Makrozelle-Silicon Assembler wird hier von besonderem Interesse sein. Im Bereiche der CAD-Forschung wird man den Traum des Silicon-Compilers weiter verfolgen. Allerdings müssen noch einige Hindernisse überwunden werden, bevor der Mensch durch ein Programm ersetzt werden kann.

Literatur

- [1] *E. Horbst, M. Nett and H. Schwartzel:* VENUS – Entwurf von VLSI-Schaltungen, Springer-Verlag, Berlin, 1986.
- [2] *R. Schumann and W. Parker:* A 32b bus interface chip. 1984 IEEE International Solid-State Circuits Conference (ISSCC 84). Digest of technical papers; p. 176...177.
- [3] *T. von Eicken:* Entwurf eines 32-Bit-Mikrocontrollers. Bull. SEV/VSE 78(1987)11, p. 606...611.
- [4] *T. Sano a.o.:* A 20ns CMOS functional gate array with a configurable memory. 1983 IEEE International Solid-State Circuits Conference (ISSCC 83). Digest of technical papers; p. 126...127.
- [5] *P. Birzele and P. Michel:* Reduced design and manufacturing time by new semicustom standards and VLSI-design methodologies. IFIP Workshop on Fast Prototyping, Grenoble/France, 1987.
- [6] *M. Udea a.o.:* A 1,5 µm CMOS gate array with configurable ROM and RAM. 1985 IEEE International Solid-State Circuits Conference (ISSCC 85). Digest of technical papers; p. 126...127..
- [7] *A. El-Gamal a.o.:* A CMOS 32b Wallace tree multiplier-accumulator. 1986 IEEE International Solid-State Circuits Conference (ISSCC 86). Digest of technical papers; p. 194...195.

8. Schweizer Fachmesse
für öffentliche Betriebe + Verwaltungen
in Bern, 16.-19. Juni 1987



Besuchen Sie uns, Halle 6, Stand 601

Integrierte Verarbeitung von Geometrie, Sachdaten und Berechnungen

Siemens-Albis stellt vor

SICAD-Grafikinformationssystem

die zentrale Lösung für den Aufbau eines Raum-
bezogenen Informationssystems.

Blattschnittfreie Speicherung des Datenbestandes.

Wechselweiser Zugriff auf Geometriedaten und
Sachdaten.

Anwendung: **Vermessung, Energieversorgung,
Planung**

System: Siemens-Systeme 7.500
Grafikarbeitsplatz 9732

SICAD-Digitalisiersystem

das autonome Zeichnungssystem für die Graphische
Datenverarbeitung.

Lückenloser Datenfluss von der Datenerfassung bis
zur Darstellung der Ergebnisse.

Anwendung: **Vermessung, Planung**

System: Grafikarbeitsplatz 9732

GEOS

Ein Softwarepaket zur Bearbeitung numerischer
Parzellarvermessungen.

Eignet sich speziell für die Berechnungsaufgaben
bei der Erstellung und Nachführung numerischer
Parzellarvermessungen.

Es können Pläne, einzelne Parzellen oder Ausschnitte
grafisch dargestellt werden.

Anwendung: **Vermessung**

System: Personal-Computer
PC-D2, PC-X, PC-MX2

SIBAU

Die erschwingliche Lösung für das Architekturbüro.

Erstellt im Rahmen der Bauplanung Ansichten, Per-
spektiven und Schnitte.

Modelliert dreidimensional und erstellt Massenaus-
züge.

Anwendung: **Architektur, Ingenieurbüros**

System: Grafikarbeitsplatz 9732

Bürosystem

Ein System zur Bearbeitung und freien Gestaltung
von Texten, Diagrammen und Grafiken.

Das Bürosystem legt Dokumente ab, verschickt Post
und unterstützt gleichzeitig alle PC-Funktionen.

Anwendung: **Büroarbeiten**

System: 5800

Gemeindesoftware

Die anwenderfreundliche Lösung für Kommunal-
behörden.

System: PC-MX2

Computer + Communication von Siemens-Albis

Siemens-Albis Aktiengesellschaft

Daten- und Informationssysteme
Freilagerstrasse 28, 8047 Zürich



Steigen Sie ein in den Wachstums-Markt der ASICs. Mit Fr. 49'000.-

»ASICs sind teuer, kosten Zeit und lohnen sich nur bei großen Stückzahlen.« **Forget it.** ASIC-Technologie kann jeder nutzen. Dank LASARRAY. Mit dem LASARRAY-Design-System. Komplett für Fr. 49'000.-. Es ermöglicht die Entwicklung von ASICs in kürzester Zeit. Im eigenen Unternehmen. Durch eigene Elektroniker. Und es ermöglicht die schnellste IC-Fertigung. Durch das LASARRAY-Processing-System.

Nutzen auch Sie »Bread-boarding on Silicon«. Und die Möglichkeit zur schnelleren Produktentwicklung und Markterschließung. Durch den Kauf eines eigenen Design-Systems. Oder mit der Hilfe des Dienstleistungszentrums der FELA. Ausführliche Informationen senden wir Ihnen gerne zu.

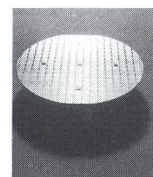
LASARRAY-Equipment:



LASARRAY S. A. · Gottstattstr. 24 · 2504 Biel
Tel. 032/410841 · Tx. 931 452 las ch · Fax 032/412828

Das Geheimnis: die »durchgängige Software«.

Für die LASARRAY-Software wurde ein neuer Begriff geprägt. **Durchgängig.** Durchgängig, weil die vom Elektroniker auf einer eigenen Workstation erzeugten und produzierten Design-Daten sich wie ein roter Faden durch die Fertigung ziehen: die Software ermöglicht die komplette Konstruktion des ASIC, inkl. der Logikprüfung und der Simulation gleicher Software liefert belichtete vorfabriziergleiche Software testet Chips. Das erklärt zwei LASARRAY: **schnellste Stunden.** Und: **Wirt-ab 5 Chips.**



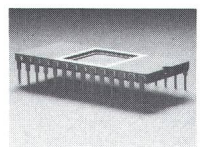
Vorgefertigte Basis-Wafer



LASARRAY-Design-System



Belichten, Verpacken, Testen im LASARRAY-Processing-System



Verpackte und getestete Chips

LASARRAY-Dienstleistung:



FELA Electronic AG · 8512 Thundorf
Tel. 054/54 11 11 · Tx 76 130 · Fax 054/53 3494