LSISIMULATOR – ein leistungsfähiges Programmsystem zur Funktionsverifikation hoch- und höchstintegrierter Schaltkreise

F. Rößler, KDT, Erfurt; P. Fischer, Dresden; A. Möschwitzer, Dresden; W. Hecker, KDT, Erfurt

Eine grundlegende Voraussetzung für den erfolgreichen und effektiven Entwurf von LSI- und VLSI-Schaltkreisen ist die weitgehende Ausschaltung menschlicher Fehlerquellen, die Entlastung des Menschen von Routinearbeit. Insbesondere gilt dies für den Prozeß der Funktionsverifikation. Dieser Prozeß war bisher mit einer erheblichen Routinearbeit verbunden, die wiederum eine große Fehlerquelle war. Denkt man nur einmal an den mühseligen Prozeß der Parameterermittlung aus dem Layout zum Zwecke nachfolgender Simulationen. Diese gesamte Strecke wird jetzt mit LSISIMULATOR in vollständig automatisierter, zuverlässiger und effektiver (sehr kleine Bearbeitungszeiten) Weise möglich. Damit ist es auch gleichzeitig möglich, noch vor der ersten Präparation in weitgehend perfekter Form jede Fehlerquelle (bis auf komplizierte physikalische und innerelektronische Fehlerbilder, Effekte 2. Ordnung) auszuschließen und damit Änderungsläufe auf ein Mindestmaß zu reduzieren. Die gesamten Turnaround-Zeiten können damit drastisch reduziert werden.

In Bild 1 ist ein Überblick über das Programmsystem LSISIMULATOR gegeben. In einer Datei liegen die Layoutdaten in der üblichen polygonorientierten Form vor. Im Programm LSISIM 01/03 wird sie in eine Bitmusterdatei (s. Abschn. 3. Bild 2) umgewandelt. Mit dem Programm LSISIM 04 kann eine Prüfung der Layoutregeln (design rule checking) erfolgen. Im Programm LSISIM 05 erfolgt eine Rückwandlung der Layoutstrukturen in Bauelemente. Die rückerkannten Bauelemente (Transistoren, Kapazitäten, Widerstände) werden mit ihren zugehörigen Knoten und elektrischen Werten (b/L-Verhältnisse, Kapazitätswert) in speziellen Tabellen abgelegt. Diese Probleme werden im Abschn. 3. behandelt. Schließlich erfolgt im Programm LSISIM 06 sowohl eine Logiksimulation als auch eine ereignisorientierte dynamische Netzwerksimulation und zusätzlich eine grafische Gatterdarstellung [1].

1. Zielstellung und erste Ergebnisse des Programmsystems

Mit diesem Programmsystem ist es z.Z. möglich, Schaltkreise bis zu 50000 Transistoren in geschlossener Form zu behandeln, die Entwurfsregeln zu prüfen und elektrisch zu analysieren.

Die Bauelementedefinitionen sind über eine einfache Sprache variabel und technologieunabhängig. So lassen sich bis zu 27 Bauelementearten verwalten. Die Anzahl der leitfähigen Ebenen ist beliebig hoch wählbar. Jede Bauelementeart wird durch eine charakteristische Ebenenfolge aus dem Layout erkannt. Aus heutiger Sicht sind alle MOS/CMOS-Technologien simulierbar. Das Programm ist auch für interaktive Optimierung von Schaltkreisen ausgelegt. Deshalb ist es mit der grafischen Plottersoftware gekoppelt. Die vom Programm verwalteten Transistoren können über grafische Darstellung direkt zugeordnet und damit interaktiv beherrscht werden, wodurch auch die Sonderbehandlung entwurfsseitig vorgegebener markanter Stellen des Schaltkreises möglich ist. Infolge der Bitmusterdarstellung hängt die Rechenzeit linear von der Gesamtkantenlänge (Summe der Kantenlängen der Einzelelemente) ab und wird deshalb im VLSI-Bereich weniger als linear mit der Bauelementeanzahl zunehmen.

So zeigten Rechenzeiten mit LSI/VLSI-Beispieltypen folgendes:

— für einen Schaltkreis mit etwa 8000 Transistoren betrug die Rechenzeit für Bauelementerückwandlung aus dem Layout ≈ 4 h

— für einen Schaltkreis mit etwa 30000 Transistoren betrug die Rechenzeit für Bauelementerückwandlung aus dem Layout $\approx 5{,}5$ h

(etwa die gleichen Zeiten ergaben sich für das vollständige design rule checking).

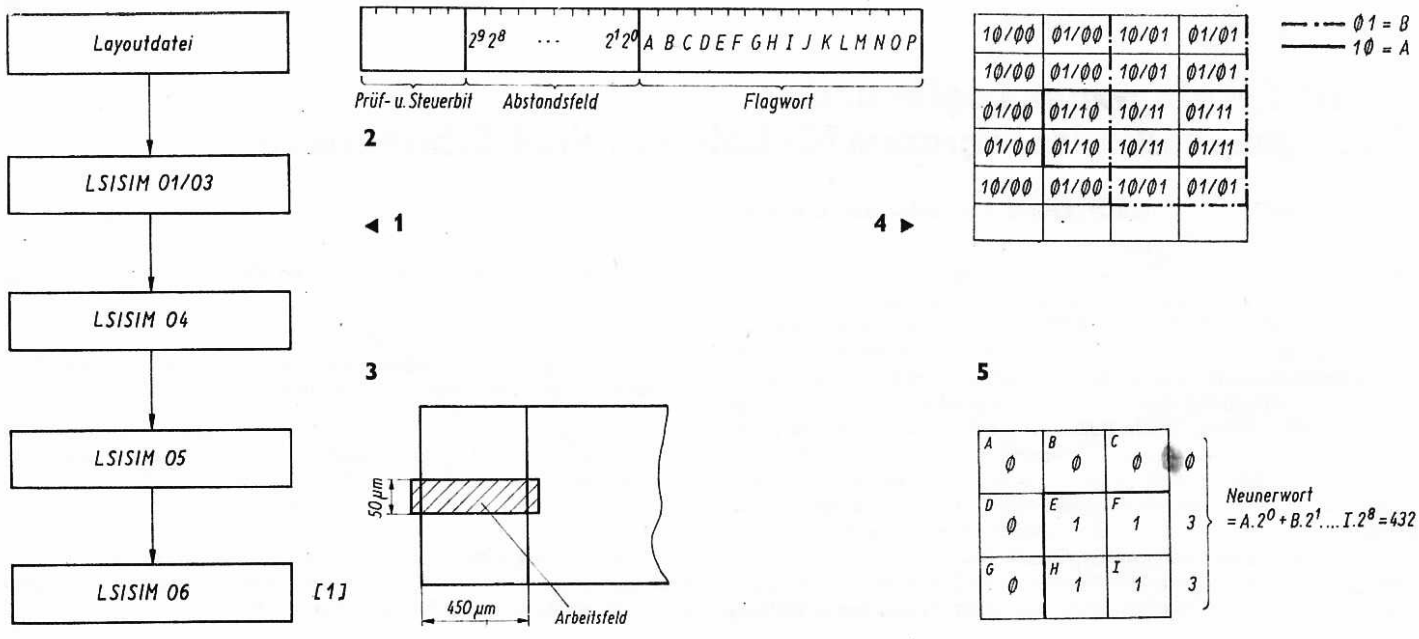
2. Abbildung der Geometrie im Programmsystem LSISIMULATOR

2.1. Datendarstellung

Die geometrische Beschreibung eines Schaltkreises erfolgt in den einzelnen technologischen Ebenen in Polygonform. Für die Behandlung inhaltlicher Probleme des Datensatzes ist es notwendig (z.B. Entwurfsregelprüfung), jedes Polygon mit jedem Polygon in Verbindung zu bringen. Das führt in LSI-Schaltkreisen zu unvertretbaren Rechenzeiten. Beim LSISIMULATOR wird in jedem Punkt (Bildpunkt) eines Chips die Kombination aller Ebenen in der Form eines Datenwortes der sogenannten Bitmusterdatei zugeordnet (Bild 2). Das erfolgt mit dem Programm LSISIM 01/03.

Das Flagwort für die Ebenen (16 bit) enthält an der Stelle eine 1, wenn die Ebene im Bildpunkt vorhanden ist, sonst eine Ø.

Sind z.B. alle 16 Ebenen übereinander vorhanden, so sind alle Flagbits 1. Das Abstandsfeld markiert die Anzahl der Bildpunkte dieser Zeile, für die die gegebene Ebenenkombination erhalten (d.h. ungeändert) bleibt. Der Abstand wird durch rechtsbündige Binärzahlen (s. Bild 2) ausgedrückt. Bei 10 bit und einem 1 μm-Raster für die Bildpunkte wäre damit ein Arbeitsbereich von 1024 μm darstellbar. Im Interesse eines realisierbaren Speicherplatzes wird das gesamte Halbleiterchip in Streifen mit der Breite 450 μm zerlegt (Bild 3). Diese Streifen werden von „Arbeitsfeldern" mit den Abmessungen 50 μm × 500 μm von unten nach oben ab-



getastet. An einer konkreten Struktur mit zwei Ebenen soll das Prinzip der Bildung des Flagwortes und des Abstandsfeldes gezeigt werden (Bild 4). Waagerechte Kanten werden dabei besonders gekennzeichnet. Das erfolgt durch Setzen eines Bits (Prüf- bzw. Steuerbit im Bild 2).
Aus den 2 Fragen

— welche Ebenenkombination liegt im Aufpunkt (Bildpunkt) vor?
— welcher Ebenenwechsel findet in welcher Weise statt?

werden alle Programmverzweigungen im LSISIMULATOR erzeugt.
Für manche elektrischen Aussagen ist es notwendig, eine Ebene unbedingt auszuschließen bzw. andere Ebenen wahlfrei zu lassen. Das geschieht im ersten Fall durch ein Minus (—) und im zweiten Fall durch eine Flagmaske.
So lautet z.B. die Ebenenbedingung für NMOS- SGT Kontaktloch Al-n^+-Kontakt: A-BCD (ACD aber nicht B)
oder für
Depletionstransistor: AB-FG (A, B, G aber nicht F).
Alle nicht aufgeführten Ebenen sind gleich. Das Ausblenden nicht wichtiger Ebenen erfolgt mit der Flagmaske wie folgt:
Zum Beispiel für Kontaktloch Al-n^+-Kontakt

Flagwort	XXX1101
Flagmaske	0001111
Ebenen	GFEDCBA

und für den Depletionstransistor

Flagwort	10XXX11
Flagmaske	1100011
Ebenen	GFEDCBA

Diese Bitmusterdatei kann auch zum Rückwandeln in die Polygondarstellung verwendet werden. Dadurch kann diese Darstellungsweise auch z.B. auf

— Layoutkorrekturen bei Entwurfsregelverletzung
— Gummizellenentwurf

erweitert werden.

3. Bauelemente- und Strukturdefinition, Bauelementerückwandlung

Wie bereits im Abschn. 2. festgelegt, erfolgt das Erkennen bestimmter Layoutstrukturen (z.B. Bauelemente, Verbindungsleitungen u.a.) über Ebenenflags und Flagmasken. Im LSISIMULATOR wird jeder Ebene eine Zahl (mathematische Abbildungsebene) zugeordnet.
Es lassen sich mit dem LSISIMULATOR je 50 verschiedene Bedingungen für Knoten, Verbindungsleitungen, Transistoren, Kapazitäten bei 27 verschiedenen mathematischen Abbildungsebenen verwalten. Als Beispiel soll die Bedingung für einen Enhancementtransistor in NMOS-SGT betrachtet werden:
Die Dreipoldefinition besteht aus 8 Teilen.

1. Kennzeichen T
2. Transistorflagwort AB-F-G
3. Flagbelegung für die b-Richtung B-A
4. Flagbelegung für die L-Richtung A-B
5. Gateanschlußebene B
6. Sourceanschlußebene A
7. Drainanschlußebene A
8. Transistorname ET1

Diese Informationen werden in Common-Feldern, das Netzwerk wird in 3 Listen abgebildet:

— Knotenliste
— Kapazitätsliste
— Transistorliste

Die zeilenweise Abarbeitung bedingt, daß die Knoten und Bauelemente „zerschnitten" und „durcheinander" gefunden werden. Aus diesem Grunde werden zunächst Mikroknoten und Mikrobauelemente zugelassen, die durch eine QUEUE als zusammengehörig gekennzeichnet werden und so ohne langen Suchprozeß aufrufbar sind. Der Suchalgorithmus bedient sich eines Neunerfeldes (Bild 5), das mit einem Neunerwort charakterisiert wird, z.B. wie im Bild 5 mit angegeben ist.
Die 1 kennzeichnet jeweils, daß eine bestimmte Ebenenkombination vorliegt. Das Neunerwort dient zur Auswahl der verschiedenen Behandlungsroutinen (z.B. auch im Entwurfsregelprüfprogramm LSISIM 04). Näherungen müssen bei diesem Prinzip bei Schrägen gemacht werden. Das Programm LSISIM 05 zur Bauelementerückwandlung besteht im wesentlichen aus folgenden 3 Bestandteilen:

TRANS
führt Regie über das Rückerkennnen der Mikrobauelemente, wobei das Rückerkennen hauptsächlich durch den oben erläuterten Algorithmus SUCHE ausgeführt wird

VERDI
fügt die Mikrobauelemente zu konzentrierten Bauelementen zusammen

DIM
weist die elektrischen Parameter der Bauelemente zu (C, b, L).

4. Schlußbemerkung

Bei gewissenhafter Arbeit mit diesem Programmsystem sind Entwurfsfehler nahezu auszuschließen. LSISIMULATOR bringt damit Aufwandseinsparungen an Versuchsdurchläufen, die Vermeidung von manuellen Plotterkontrollen. Es ist für das hierarchische VLSI-Entwurfssystem überhaupt unentbehrlich. Das Prinzip kann auch für den „Gummizellenentwurf" und „probeweise Layoutkorrekturen" angewendet werden. Die zur Aufstellung dieses Projekts notwendigen umfangreichen programmier- und rechentechnischen Arbeiten wären ohne die Mithilfe zahlreicher Kollegen nicht möglich gewesen. Stellvertretend soll besonders Dr. *Wüstefeld*, Dr. *Wiederhold* gedankt werden.

Eingegangen am 6. Mai 1983 NaA 9099

Literatur

[1] *Hecker, W.* u. a.: LSINET – ein neues Logik- und Timingsimulationsprogramm für LSI- und VLSI-Schaltkreise, Nachrichtentech., Elektron., 34 (1984) 6, S. 214–218

Dr. sc. techn. *Franz Rößler* und Dr.-Ing. *Wolfgang Hecker*, VEB Mikroelektronik Karl Marx Erfurt, 5020 Erfurt, Rudolfstr. 47
Dr.-Ing. *Peter Fischer* und Prof. Dr.-Ing. habil. *Albrecht Möschwitzer*, Technische Universität Dresden, Sektion Informationstechnik, 8027 Dresden, Mommsenstr. 13

LSI NET – ein neues Logik- und Timingsimulationsprogramm für LSI- und VLSI-Schaltkreise

W. Hecker, KDT; F. Rößler, KDT, Erfurt; A. Möschwitzer, Dresden

In LSI- und VLSI-Schaltkreisen sind Tausende (1000 bis 1000000) von Einzelbauelementen integriert. Klassische Netzwerkanalyseprogramme wie MISNET und SPICE ermöglichen eine effektive Netzwerksimulation von Schaltungen mit weniger als 500 Knoten. Logiksimulationsprogramme können dagegen größere Schaltungen simulieren, jedoch ist hier als Eingabesprache meist die Gatterstruktur erforderlich. Reale Schaltkreise bestehen aber aus Transistorschaltungen, denen die Gatterstruktur (z.B. wegen Nutzung dynamischer Effekte und Transfergates) nicht so ohne weiteres angesehen werden kann. Außerdem liefern selbst dynamische Logiksimulationsprogramme nicht die Zeitfunktionen der Knotenspannungen, sondern nur ausgewählte Zustände.
Deshalb ist ein Programm wünschenswert, das selbst für große und sehr große Schaltkreise direkt aus einer Transistordarstellung eine Logiksimulation bzw. dynamische Netzwerksimulation möglich macht.

1. Zielstellung, Realisierungsgedanken und praktisch erprobter Leistungsumfang des Programms LSINET

Es sollte ein ereignisorientiertes Hybridsimulationsprogramm (Logiksimulation und dynamische Netzwerksimulation) geschaffen werden, das für hochintegrierte Schaltkreise bis zu 125000 Zweigelemente geeignet ist und sich in das Programmsystem LSISIMULATOR [1] einfügt.
Die Realisierung eines solchen Programms muß zahlreiche Vereinfachungen gegenüber den klassischen Netzwerkanalyseprogrammen enthalten. Die wesentlichen sind:

— Verwendung einfacher Bauelementemodelle in Form einfacher Stromgleichungen für die MOS-Transistoren
— ereignisorientierte Simulation, d.h., es werden zu jedem Zeitpunkt nur die Knoten des Gesamtschaltkreises betrachtet, die in der aktuellen Knotenliste stehen. In die aktuelle Knotenliste werden nur die Knoten aufgenommen, an denen im Teilzeitschritt eine Änderung erfolgt
— Verwendung einer einfachen Integrationsformel zur Berechnung der Knotenspannung $UK(T)$ im Zeitpunkt T

$$UK(T) = UK(T-1) + \frac{I \times S}{CK} \qquad (1)$$

CK Knotenkapazität, S Zeitschrittweite, I Nettostrom im Knoten K, der sich aus der Differenz von zu- und abfließenden Strömen zusammensetzt.
— Die Steuerung der Zeitschrittweite S erfolgt so, daß die Knotenspannungsänderung $UK(T)$- $UK(T$-1) einen bestimmten Wert nicht überschreiten kann (s.u.).
Die bisher mit diesem Programm erzielten Ergebnisse haben die LSI-Fähigkeit nachgewiesen. Die Rechenzeiten an einem Beispielschaltkreis mit etwa 8000 Transistoren ergaben folgendes:
Logiksimulation (statisch): CPU-Zeit 3 s/Flanke in Transistordarstellung, CPU-Zeit 2 s/Flanke in Gatterdarstellung
Netzwerksimulation (dynamisch): CPU-Zeit 2,5 min/Flanke.
Darüber hinaus hat das Programm verschiedene Möglichkeiten, von denen die grafische Gatterdarstellung besonders hervorzuheben ist.

2. Beschreibung einiger Algorithmen und Verfahren in LSINET

2.1. *Netzwerkbeschreibung*

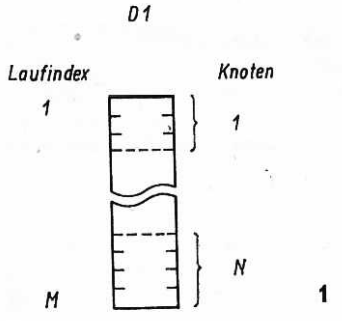
Die Beschreibung der Schaltungsstruktur erfolgt durch Abbildung in verschiedenen Listen, die durch die Bauelementerückwandlung mit dem Programm LSISIM 05 gewonnen wurden [1].
Anschlußknotenliste (D1, Bild 1)
Der Laufindex dieser Liste ist eine Zweignummer 1 ··· M. In dieser Liste stehen in fortlaufender Reihenfolge alle Knotennummern, die als Anschlußknoten zum Knoten 1, 2, ..., N gehören. Ein Anschlußknoten i zu einem Bezugsknoten k ist ein Knoten, der über einen Zweig (MOS-Transistor, Kapazität) mit k verbunden ist. Zu einem Bezugsknoten k gibt es i.allg. mehr als einen Anschlußknoten. Um das zu kennzeichnen, dient die
Zeigerliste (D3, Bild 2)
Der Laufindex dieser Liste ist die Knotennummer 1 ··· N. In dieser Liste steht jeweils die Adresse = Laufindex in der Anschlußknoten- (bzw. Steuerknoten-, s.u.)liste des ersten Anschlußknotens (Steuerknotens) des Bezugsknotens.
Steuerknotenliste (D2, Bild 3)
Der Laufindex dieser Liste ist eine Zweignummer 1 ··· M. In dieser Liste stehen die Steuerknoten zu den Bezugsknoten in fortlaufender Reihenfolge. Ein Steuerknoten i zu einem Bezugsknoten k ist ein Knoten, der den zu diesem Knoten führenden Zweig steuern kann, also das Gate des in diesem Zweig liegenden MOS-Transistors. Zu einem Bezugsknoten k gibt es i.allg. mehr als einen Steuerknoten. Um das zu kennzeichnen, dient die Zeigerliste D3 (Bild 2) und die Zweiganzahlliste (D4, Bild 4)
Der Laufindex dieser Liste ist die Knotennummer 1 ··· N. Darin steht jeweils die Anzahl der Zweige, die an den Bezugsknoten angeschlossen sind. Deshalb kann D3 durch folgende Rekursion aus D4 erzeugt werden

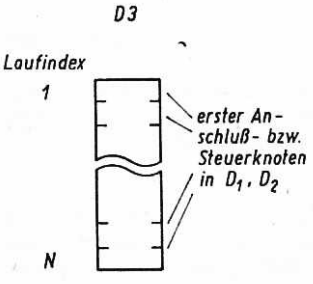
$$D3(1) = 1$$
$$D3(N) = D4(N-1) + D3(N-1) \qquad (2)$$

Die Zweigtypliste (D5, Bild 5) und die Dimensionierungsliste (D6, Bild 5) haben als Laufindex eine Zweignummer und enthalten die Codezahl des Zweigtyps (z.B. 1 = R, 2 = C, 3 = ET, 4 = DT) bzw. die b/L-Verhältnisse der Transistoren und die Widerstands- bzw. Kapazitätswerte.
Wie aus Bild 5 hervorgeht, kann man mit diesen Listen für jeden gewünschten Knoten k über Zeiger- und Zeigerlängenliste die Anschlußknoten $i \cdots (i-1+Z_k)$ in der Anschlußknotenliste und deren Eigenschaften (Steuerknoten, Zweigtyp, Parameterwert) finden.
Grundsätzlich wird zwischen folgenden Knotentypen unterschieden:
Betriebsspannungs- und Masseknoten
Funktionseingangsknoten
Knoten mit bestimmter Belegung
Reguläre (innere) Knoten.
In eine weitere Liste (D9) werden alle regulären Knoten eingetragen, die von einem Knoten gesteuert werden können. In einer Hilfsliste werden dann die Knoten spezifiziert.
Darüber hinaus ist im Bild 5 noch eine weitere Liste D7 sichtbar, die alle Knotenkapazitäten enthält.

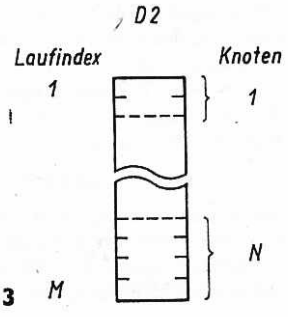
2.2. *Statische Logiksimulation, statische Anfangslösung*

Der Ablauf der Logiksimulation in LSINET ist im Bild 6 dargestellt. Zu Beginn werden allen Eingängen die logische Pegel entsprechend der anliegenden Eingangsfunktion zugeordnet.
Danach erfolgt das Eintragen aller Knoten, außer Betriebsspannungs-, Masse- und Funktionsknoten in eine aktuelle Knotenliste. Die aktuelle Knotenliste ist ein eindimensionales Feld, in dem die aktuellen Knoten in aufsteigender Reihenfolge notiert sind. Formal wird das Feld mit einer Länge, das der Gesamtzahl der Knoten entspricht, angelegt. Außer diesem Feld wird noch ein Hilfsfeld ange-

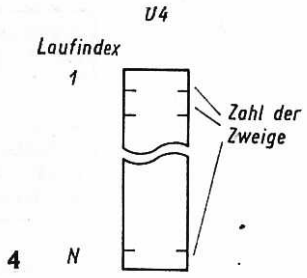




2

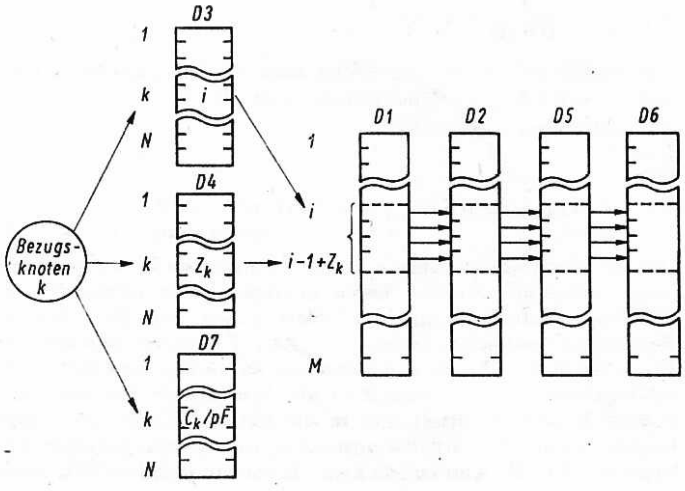


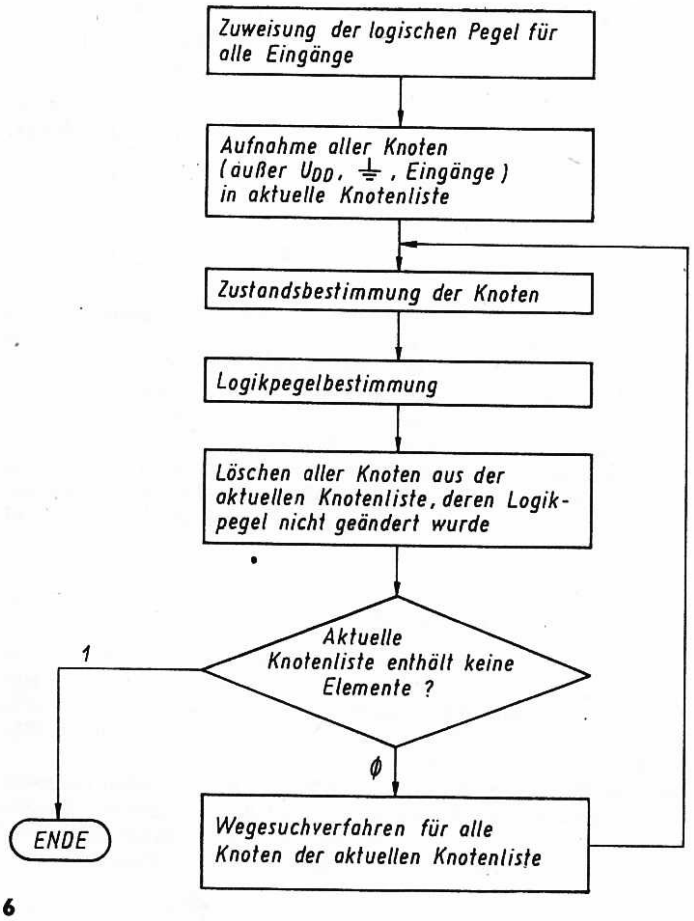
3



4

5





6

legt, in das sofort neue aktuelle Knoten eingetragen werden, wenn es die Rechnung ergibt. Die Übertragung dieses Hilfsfeldes in die aktuelle Knotenliste erfolgt stets nur am Ende eines Rechenzyklus.

Als nächstes erfolgt die Zustandsbestimmung der Knoten der aktuellen Knotenliste. Man unterscheidet 4 Zustände eines Knotens:

— Zustand 1: Knoten ist isoliert und behält seinen Zustand
— Zustand 2: Knoten wird geladen (hat nur leitfähige Pfade zu U_{DD})
— Zustand 3: Knoten wird entladen (hat nur leitfähige Pfade nach Masse)
— Zustand 4: Knoten wird geladen und entladen (hat leitfähige Pfade nach Masse und Betriebsspannung U_{DD})

Die Zustandsbestimmung erfolgt durch Pfadermittlung (s. Abschnitt 2.4.) und zusätzlicher Prüfung der Steuerknoten (ist der Steuerknoten eines Pfades Ø, so ist der Pfad nichtleitend). Wird ein leitfähiger Pfad gefunden, so wird dessen Leitfähigkeit aus der Dimensionierungsliste D6 bestimmt. Da in D6 „normierte Leitwerte" eingetragen sind (z.B. für Transistoren ausgedrückt durch das b/L-Verhältnis), gilt für die Leitfähigkeit eines zum Knoten K führenden Pfades

$$\frac{1}{D6(K)} = \frac{1}{D6(1)} + \cdots \frac{1}{D6(M)} \tag{3}$$

Die Logikpegelbestimmung erfolgt nach der Zustandsbestimmung einfach durch folgende Zuordnungsvorschrift

Z1: Logikpegel ungeändert
Z2: „1"
Z3: „Ø"
Z4: $D6(K)_{Masse}/D6(K)_{U_{DD}} = \alpha > 8$?, wenn ja: Ø
wenn nein: 1

(Bei Z4 wird also gewissermaßen ein Spannungsteiler ausgewertet). Nun werden alle Knoten, deren Logikpegel sich während dieses Simulationsdurchlaufs nicht geändert haben, aus der aktuellen Knotenliste gestrichen (solche mit Z1). Verbleiben danach noch Elemente in der aktuellen Knotenliste, so werden über den Wegesuchalgorithmus (s. Abschn. 2.4.) alle von diesem Knoten beeinflußten Knoten ermittelt und in die aktuelle Knotenliste eingetragen, womit der Simulationszyklus von neuem beginnt. Erst wenn die aktuelle Knotenliste keine Elemente mehr enthält, haben alle Knoten ihren richtigen Logikpegel entsprechend der von außen anliegenden Eingangsfunktion, und die Rechnung ist beendet.

Aus den Logikpegeln kann nun der Anfangszustand der Knotenspannungen durch die folgenden Zuweisungen bestimmt werden:

Z1: UK = OV (isolierte Knoten entladen sich durch Leckströme!)
Z2: UK = MAX (Anschlußknotenspannungen, Anschlußknotenspannungen minus Schwellspannung)
Z3: UK = OV
Z4: $\alpha > 8$) UK = OV
$\alpha > 8$) iterative Berechnung von UK so, daß die Differenz zwischen zu- und abfließenden Strömen unter einer Schranke bleibt.

2.3. *Dynamische Netzwerksimulation*

Der Ablauf dieses Algorithmus ist im Bild 7 gezeigt. Nach Festlegung der Eingangsfunktionsänderung erfolgt in einem Wegesuchverfahren (s. Abschn. 2.4.) die Ermittlung aller Knoten, die von dieser Eingangsfunktionsänderung betroffen sein könnten. Sie werden zunächst in ein Hilfsfeld eingetragen. In einem Lauf der Logiksimulation werden a) alle Knoten ermittelt, an denen bei der betreffenden Eingangsfunktion etwas geschieht (vorausschauend) und b) die Anfangslösung (s. Abschn. 2.2) berechnet. Die ermittelten Knoten, in denen etwas geschieht, können vom Hilfsfeld in die aktuelle Knotenliste eingetragen werden. In diesen Knoten K erfolgt die Berechnung des ersten Teilzeitschrittes nach der folgenden Integrationsformel [s. auch Abschn. 1., Gl. (1)];

$$UK(T) = UK(T-1) + \frac{I}{CK} \times S(T-1) \frac{\text{DELTA}}{\text{DUKMAX}(T-1)}$$

mit DUKMAX(T) = MAX (UK(T)-UK(T-1)), der maximalen Knotenspannungsänderung. Es wird geprüft, ob sie kleiner als eine vorgegebene Schranke DELTA ist. Ist das der Fall, so wird der nächste Teilzeitschritt berechnet, wenn nicht, so wird die Zeitschrittweise S gemäß

$$S(T) = \frac{\text{DELTA} \times S(T)}{\text{DUKMAX}(T)} \tag{5}$$

korrigiert und damit $UK(T)$ erneut berechnet. Dieser Mechanismus der Zeitschrittweitensteuerung ist im Bild 8 gezeigt.

Nun erfolgt erneut eine Prüfung der Knoten auf Aufnahme bzw. Löschen aus der aktuellen Knotenliste mit einem Test auf Über- oder Unterschreiten einer Schranke und Wegesuchverfahren (s. Abschn. 2.4.). Danach wird der nächste Teilzeitschritt berechnet usf. bis der stationäre Zustand erreicht ist (aktuelle Knotenliste ist dann leer).

2.4. *Wegesuchverfahren*

Hiermit werden Teilnetzwerke bestimmt, in dem zu einem Knoten K alle Pfade aus Zweigen und Knoten gesucht werden, die zu Betriebsspannung oder Masse führen, bzw. am Gate eines MOS-Transistors enden.

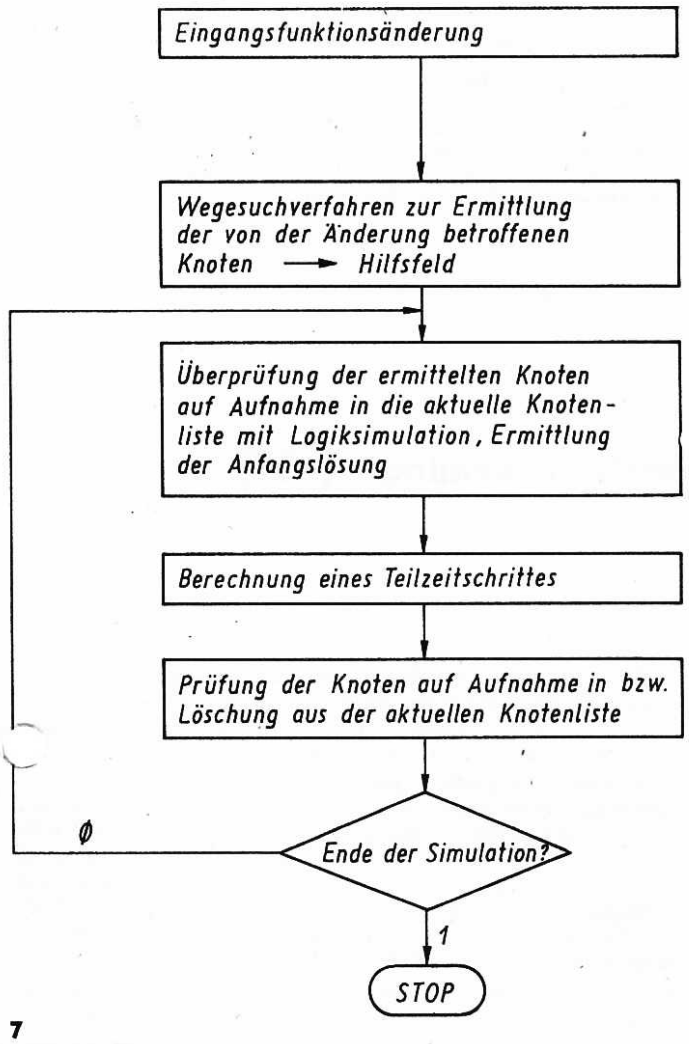
Vom Bezugsknoten K ausgehend wird über den Zeiger (s. Bild) der erste Anschlußknoten (Anschlußknoten 1. Ordnung) ermittelt. Dafür wird wieder über den Zeiger der nächste Anschlußknoten (2. Ordnung) ermittelt usf. Der Suchalgorithmus wird abgebrochen, wenn Masse, Betriebsspannung oder Gate (freier Steuerknoten) als nächster Knoten gefunden wird. Damit ist der erste Pfad gefunden. Endet dieser Pfad z.B. an einem Anschlußknoten n-ter Ordnung, so wird nun zunächst zu einem Knoten n-1-ter Ordnung zurückgegangen und der nächste Knoten n-ter Ordnung geprüft. Sind alle vom Knoten n-1-ter Ordnung ausgehenden Pfade (Teilpfade) ermittelt, wird zum Knoten n-2-ter zurückgegangen und in gleicher Weise verfahren. Dann wird zu den Knoten n-3-ter Ordnung zurückgegangen usf. bis man wieder beim Ausgangsknoten K angelangt ist. Dann sind alle von K ausgehenden und sich u.U. verzweigenden Pfade gefunden.

Das Ergebnis dieser Pfadermittlung wird in Matrizen notiert (s. Abschn. 2.5.).

Ein weiterer Teil des Wegesuchverfahrens ist die Kettenverfolgung. Hiermit können die Teilnetzwerke ermittelt werden, die von einem Knoten eines anderen Teilnetzwerkes beeinflußt werden.

2.5. *Grafische Gatterdarstellung*

Die grafische Gatterdarstellung ermöglicht die zeichnerische Ausgabe eines Gatters auf einem Bildschirm oder einem Drucker. Sie ermöglicht damit die Kontrolle auf Richtigkeit durch Vergleich mit der Schaltbildzeichnung.

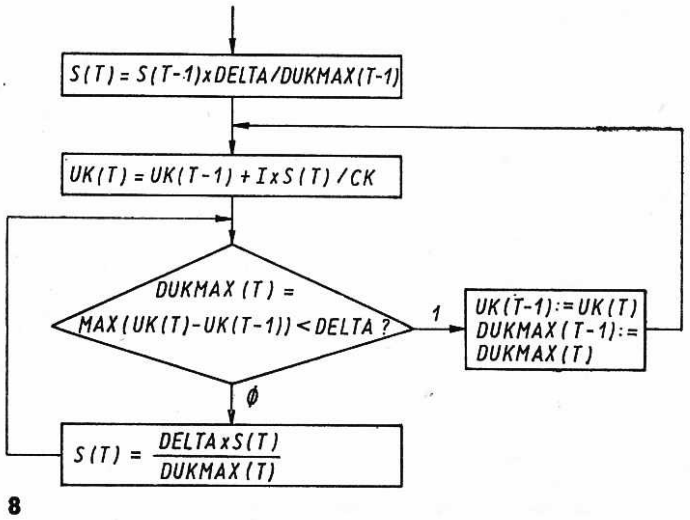


7

Die grafische Gatterdarstellung setzt eine Gatterdefinition voraus.
NMOS-Gatter: An jedem Knoten, der nur einen Zweig zur Betriebsspannung hat.
Die Gattergenerierung an diesem erfolgt durch Ermittlung aller Pfade zu einem Endknoten (Masse, Betriebsspannung, Transistorgates)
CMOS-Gatter: Alle Knoten, an denen p- und n-Kanal-Transistoren angeschlossen sind.
Die Gewinnung der Gatterstruktur erfolgt mit dem Wegesuchverfahren (Pfadermittlung). Das Ergebnis der Pfadermittlung wird in Matrizen eingetragen:

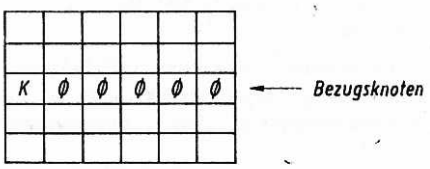
Anschlußknotenmatrix

Ausgehend von der Bezugsknotenzeile K werden nach oben alle Pfade zur Betriebsspannung, nach unten alle Pfade zu Masse oder Transistorgates (freie Steuerknoten) eingetragen (Bild 9).



8

9



Steuerknotenmatrix

Sie ist ebenso aufgebaut wie die Anschlußknotenmatrix, nur enthält sie in der Bezugsknotenzeile nur Ø, nach oben werden die Steuerknoten der Pfade nach Betriebsspannung, nach unten die der Pfade nach Masse bzw. freie Steuerknoten eingetragen.

Zweigtypmatrix

Sie ist wiederum so aufgebaut wie die Anschlußknotenmatrix mit Ø in der Bezugsknotenzeile und den Codezahlen des Zweigtyps (s. Abschn. 2.1.) für die Pfade zur Betriebsspannung (nach oben) bzw. zu Masse und freien Steuerknoten (nach unten). (Weiterhin kann man noch Matrizen für die Dimensionierung der Gatterelemente und Logikpegel aller Anschlußknoten aufstellen.)
Nun erfolgt die grafische Ausgabe, in dem die Matrizen zeilenweise ausgewertet werden.

PRINT
1. Zeile Anschlußknotennummer
Verbindungssymbol
1. Zeile Steuerknotennummer
1. Zeile Bauelementetypmarkierung
2. Zeile Anschlußknotennummer
⋮
K-te Zeile Druck einer Verbindungslinie (Gatterausgang)
K + 1. Zeile Anschlußknotennummer
Verbindungssymbol
⋮

Im Bild 10 ist ein Beispiel einer solchen grafischen Gatterdarstellung gezeigt.

3. Eingabesprache

Die Eingabesprache ermöglicht eine interaktive Kommunikation mit dem Rechner. Es sind folgende Zuweisungen möglich:
Betriebsspannung = Spannungswert : Knotennummer;
(z.B. UB = 5:3;)
Masse: Knotennummer
(z.B. MA:5)
Statische Eingangsfunktion = Spannungswert : Knotennummer
(z.B. UE = 4:17, 18)
Dynamische Eingangsfunktion: Knotennummer;
(z.B. F3:7)
Für die dynamischen Eingangsfunktionen muß eine Funktionsliste angefertigt werden:

Logisch

Bei festem Zeitraster werden eine Taktzeit T und die Spannungswerte bei High- und Lowpegel angegeben
z.B.
UEH = 5;
UEL = 0;
T = 100
F2:HHL
F3:5HL
F4:5HHL

Bei variablem Zeitraster wird der High- und Lowpegel, z.B.
UEH = 5;
UEL = 0;
und

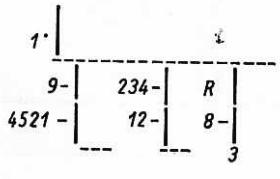


Bild 10. Zeichenerklärung
n- Enhancementtransistor
n Depletiontransistor
R Widerstand

Funktionsnummer: Zeit 1, Log. Pegel 1, Zeit 2, Log. Pegel 2, ...
(z.B. F1:0, H, 150, L, 200, H)
angegeben.

Elektrische Netzwerksimulation

Hier werden Rampenfunktionen in folgender Form eingegeben
Funktionsnummer: Zeit 1, Spannungswert 1, Zeit 2, Spannungswert 2, ...
(z.B. 0,5, 100,5, 110,0, 200,0, 210,5)
Diese Spezifikationslisten werden über einen Eingabeinterpreter aufgenommen und weiter verarbeitet.

Eingegangen am 6. Mai 1983 NaA 9100

Literatur

[1] *Rößler, F.*, u. a.: LSISIMULATOR – ein leistungsfähiges Programmsystem zur Funktionsverifikation hoch- und höchstintegrierter Schaltkreise. Nachrichtentech., Elektron., 34 (1984) 6, S. 213 u. 214

Dr.-Ing. *Wolfgang Hecker*, Dr. sc. techn. *Franz Rößler*, VEB Mikroelektronik Karl Marx Erfurt, 5020 Erfurt, Rudolfstr. 47

Prof. Dr. habil. *Albrecht Möschwitzer*, Technische Universität Dresden, Sektion Informationstechnik, 8027 Dresden, Mommsenstr. 13

Multimikroprozessorsystem zur Steuerung eines Intelligenten Grafischen Terminals (IGT) Technische Realisierung

H. Leich; F. V. Levchanovsky; V. I. Prikhodko, Dubna

In [4] wurde die Architektur eines Multimikroprozessorsystems (MMPS), das als Steuerkern für ein intelligentes grafisches Terminal eingesetzt wird, vorgestellt. Nachfolgend wird die technische Realisierung des MMPS beschrieben.

1. Interface eines Moduls mit dem Systembus

Das Systembus-Interface eines Moduls besteht aus drei Baugruppen:

— den Busdrivern zur Ankopplung der internen Daten-, Adreß- und Steuerleitungen an den Systembus
— einer Steuerschaltung, die ausgehend von der am internen Bus anliegenden Adresse eine Systembus-Anforderung stellen sowie die Bussteuerung übernehmen kann
— einem Steuerblock (control-memory), über den die Kommunikation zwischen den Modulen abgewickelt wird.

Bild 1 zeigt die Schaltung der genannten Baugruppen. Bei Adressierung externer Speicherbereiche wird mit dem Status-Strobe-Signal (STSTB) der CPU ein Busanforderungs-Flipflop gesetzt (BR-FF) und eine Anforderung (/BR) an den Busarbiter gestellt. Gleichzeitig wird das Bereitsignal (RDY) für die CPU ausgeschaltet. Bei Busgewährung (BG) und nicht mehr belegtem Bus (BUSY inaktiv) wird die Bussteuerung mit der Rückflanke des Bustaktes vom Modul übernommen, d.h., es werden die Driver für Adreß-, Daten- und Steuerleitungen aktiviert. Die Übertragungsrichtung für die Datendriver wird durch das Lesesignal (/MRD) gesteuert. Der adressierte Slavemodul quittiert die erkannte Busoperation durch das Signal /ACK, das das Bereitsignal der CPU wieder einschaltet. Durch Ausschalten von /MRD bzw. /MWR durch die CPU wird /BR zurückgenommen und der Systembus wieder freigegeben.

Der Aufbau des als Steuerblock dienenden Speichers mit seinem Zweitor-Interface erfolgt mit Schaltkreisen vom Typ KR1802IR1 (16 × 4 bit Dual-Port-Registerfile [9]). Damit können Steuerblöcke mit 16, 32 oder 48 Byte einfach aufgebaut werden.

Ist ein Modul eine gemeinsame Ressource für andere, so sind im Systembusinterface hardwaremäßig spezielle Flags implementiert, die eine Test-and-Set-Funktion (TAS) verwirklichen. Mit Hilfe eines weiteren Flags (BUSY) wird die Auftragsbereitung synchronisiert. BUSY wird gesetzt, sobald ein Auftrag in den Steuerblock eingetragen wurde. Nach Beendigung der Bearbeitung wird BUSY durch die CPU rückgesetzt. Das TAS-Flag einer Ressource muß vom aktuellen Auftraggeber rückgesetzt werden.

Die schaltungstechnisch realisierten TAS- und BUSY-Flags erleichtern die Organisation der Kommunikation auf Softwareebene.

2. Speichermodul

Um einen für verschiedene Anwendungen einsetzbaren Modul zu erhalten, wurde der Speicher so implementiert, daß er in Portionen zu 16 K Byte aufrüstbar ist und daß außerdem die Möglichkeit besteht, diesen Modul in größeren Systemen einzusetzen. Er stellt dann eine Seite von 64 K Byte dar. Die Seitenauswahl (durch das Steuersignal SEL) muß jedoch extern erfolgen, d.h. die Speichersteuerung decodiert keine Adressen über 64 K hinaus. Möglichkeiten zum Anschluß des Speichermoduls an andere Bussysteme (z.B. CAMAC) sind in Form von freien Bausteinplätzen mit Universalverdrahtung zur Realisierung zusätzlicher Interfacelogik vorgesehen. Im betrachteten MMPS wird ein Speichermodul mit einer Kapazität von 48 K Byte eingesetzt (Busadressen von 16 K bis 64 K).

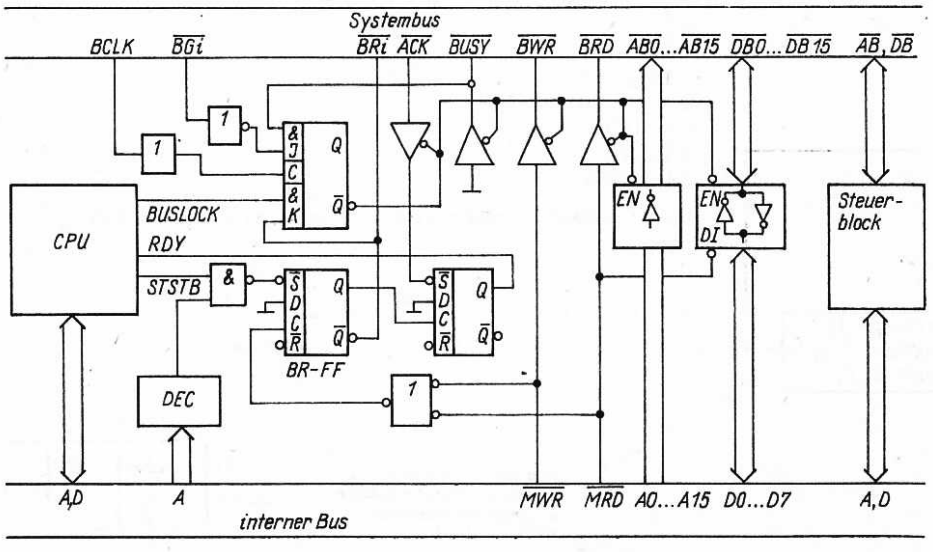


Bild 1. Systembusinterface