

Rechnersimulation digitaler Schaltungen reduziert Labormessungen

Gessner, E.; Schalldach, H.-H.; Werrmann, G.;
Zech, K.-A.

Zum Entwurfsprozeß einer digitalen Schaltung gehört die Analyse der Synthesergebnisse, mit welchen Mitteln auch immer diese erreicht worden sind. In der Regel wird zu diesem Zweck eine Laborschaltung aufgebaut und unter typischen Umweltbedingungen durchgemessen. Bei kritischen Zeitbedingungen erfolgt zusätzlich eine Analyse von Teilen der Schaltung unter worst-case-Bedingungen, die z. B. manuell, per Bleistift und Millimeterpapier, durchgeführt werden kann. Bei steigender Komplexität der Schaltung und steigendem Integrationsgrad der Bauelemente wächst die Wahrscheinlichkeit, daß die Analyseergebnisse eine Schaltungsänderung zur Folge haben. Nach der Korrektur muß sich wieder ein Analyseschritt anschließen, der bezeichnete Weg wird erneut durchlaufen.

Der Aufbau einer Versuchsschaltung kann dann eingespart werden, wenn der Entwurf einer Korrektur unterzogen werden muß und die zu beseitigenden Mängel auf andere Weise feststellbar sind. Ein Mittel für eine Schaltungsanalyse, die den Versuchsaufbau umgeht, ist die Simulation auf einem Rechner. Im Institut für Nachrichtentechnik wurde zu diesem Zweck das Programmpaket SIMPER [1] geschaffen, das in diesem Beitrag vorgestellt worden soll. Es unterscheidet sich von anderen bekanntgewordenen Simulationsprogrammen [8, 9, 10] durch die Aufgabenstellung, den Lösungsweg und das Leistungsvermögen.

Bei der Simulation wird der physische Aufbau der Schaltung ersetzt durch die Erfassung der Schaltungsdaten und die rechnerinterne Erzeugung eines Schaltungsmodells. Das Simulationsexperiment als Möglichkeit der Modellierung des Zusammenspiels Umwelt - Schaltung tritt dann an die Stelle des Durchmessens der elektrischen Potentiale der Schaltung bei angelegten Eingangsimpulsen.

Der Rationalisierungseffekt dieser Vorgehensweise liegt in dem wesentlich verkürzbaren Zeitaufwand für die Analyse.

Nach Schätzungen, denen Erfahrungen aus dem INT zugrundeliegen, können bei diesem Entwurfsabschnitt ca. 80 Prozent Arbeitszeit eingespart werden. Darüber hinaus sind ohne große Mühe auch solche Eigenschaften eines Schaltungsentwurfs erfaßbar, die bisher nicht oder nur mit großen Schwierigkeiten gemessen oder abgeleitet werden konnten. Hierzu gehört das sichere Funktionieren aller Schaltungsexemplare, die sich bei einem gegebenen Bauelementesortiment aus der Streuung der zeitlichen Bauelementeeigenschaften ergeben.

Mit SIMPER kann außerdem – fast nebenher – das wichtige Problem der Testschrittabellenerzeugung für die Fertigungsprüfung gelöst werden.

Bild 1 zeigt schematisch den Entwurfsprozeß unter Einbeziehung eines Rechners zur Schaltungsanalyse.

Schaltungsentwurf
EDV, Anwendung

1. Aufgabenstellung

Aufgabe war es, dynamische Simulationsprogramme zu entwickeln, die

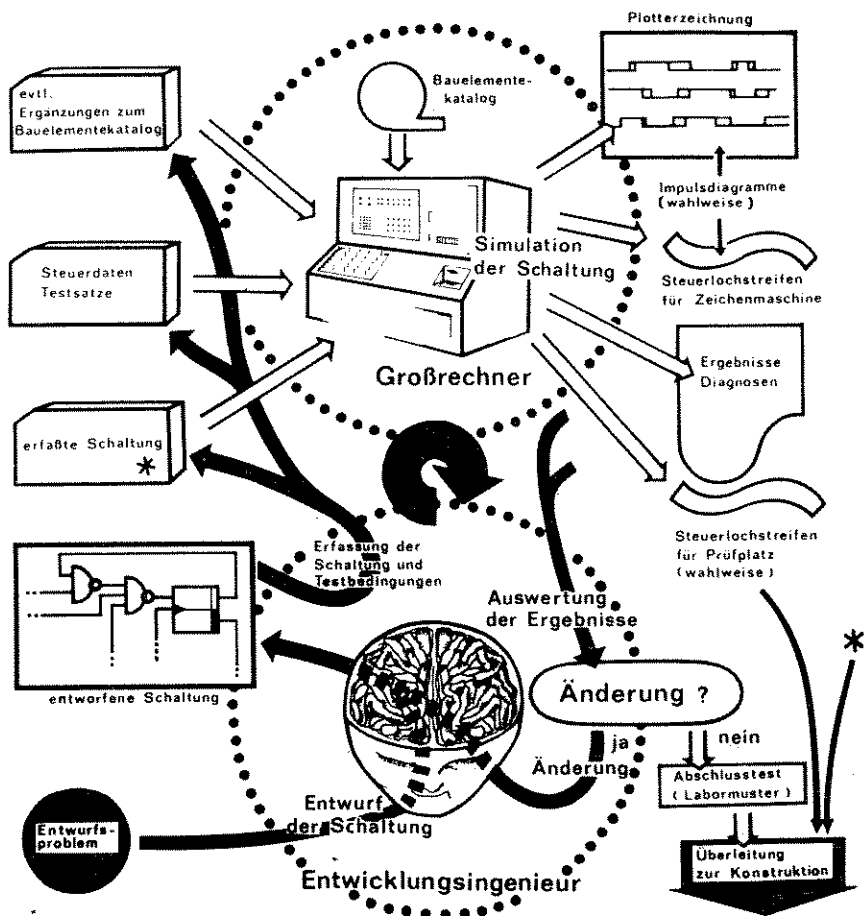
- eine weitgehend wirklichkeitsgetreue Nachbildung der in realen Schaltungen ablaufenden dynamischen Prozesse auch bei hohen Takt- oder Betriebsfrequenzen vornehmen
- außer Aussagen über das typische Schaltungsverhalten auch worst-case-Aussagen liefern
- möglichst für alle vorkommenden Schaltungstypen (synchron getaktet, synchron mit beliebiger schaltungsinterner Umformung der Taktfrequenz, asynchron getaktet, asynchron) geeignet sind
- auch die Simulation großer bis sehr großer Schaltungen (mehrere hundert Funktionselemente) erlauben
- eine Anpassung an Auflösungsforderungen durch einen innerhalb gewisser Grenzen wählbaren Zeitmaßstab ermöglichen
- das gegenwärtig und in naher Zukunft eingesetzte Sortiment an integrierten Schaltkreisen berücksichtigen und darüber hinaus auf neue Bausteinreihen jederzeit erweiterbar sind
- einen möglichst geringen Datenerfassungs- und Rechenzeitaufwand erfordern
- und Ergebnisdaten liefern, die leicht zu interpretieren sind und sich nur wenig von denen unterscheiden, die der Schaltungsentwickler benutzt.

Bei der dynamischen Simulation spielen die Zeitparameter der Funktionselemente, insbesondere die realen Verzögerungswerte, eine dominierende Rolle, im Gegensatz zur statischen (oder synchronen) Simulation, bei der sie völlig vernachlässigt werden, und wo lediglich das Zeitraster der Informationsübernahme und -abgabe bei synchron getakteten Schaltungen durch den Takt und bei asynchronen durch die zeitliche Folge der Eingangsbedingungen festgelegt ist. Die statische Simulation kann deshalb nur die Aussage liefern, ob eine Schaltung im Prinzip logisch richtig entworfen ist, aber nicht, ob sie unter den vorgesehenen Betriebsbedingungen auch richtig funktioniert. Die Verwendbarkeit der Programme für alle möglichen Schaltungstypen erfordert, daß alle Eingangssignale der Schaltung, z. B. auch Takt-, Setz- und Rücksetzsignale sowie die Eingangssignale aller Speicher als gleichberechtigt hinsichtlich ihrer Behandlung im Simulationsalgorithmus anzusehen sind. Schaltungsintern gesehen liegt dann ein asynchrones Simulationsmodell vor. Das bietet die Möglichkeit, auch Schaltungen zu behandeln, bei denen die Setz-, Rücksetz- oder Takteingänge einiger Funktionselemente logisch beschaltet sind oder der Takt logisch beliebig ungeformt ist.

Bild 1
Rolle der Simulation im Entwurfsprozeß digitaler Schaltungen

Für die Zeitparameter der Funktionselemente, insbesondere die Verzögerungen, sind in der Praxis keine definierten Werte, sondern nur Toleranzbereiche angebar, deren Größe für jeden Funktionselementtyp von solchen Einflußgrößen wie fertigungsbedingten Streuungen, parasitären Kapazitäten, Betriebsspannungsschwankungen, Belastungsschwankungen und Temperaturänderungen abhängt. Von den genannten Einflußgrößen, die zum Teil zufälligen Charakter tragen, dominieren die fertigungsbedingten Streuungen vor allen anderen.

Von den digitalen Schaltungen wird fast ausnahmslos verlangt, daß sie auch bei der für das geforderte Schaltungsverhalten ungünstigsten Relation der extremsten Verzögerungszeiten aller Funktionselemente zueinander noch sicher arbeiten. Sie müssen deshalb einer von den Verzögerungstoleranzbereichen ausgehenden worst-case-Analyse unterzogen werden. Deren Ergebnisse liefern Aussagen über die Bereiche mit bestimmten und unbestimmten Signalwerten, Verzögerungsbereiche von Signalfanken und das Auftreten von Hasards. Eine von konstanten Verzögerungswerten, beispielsweise den Mittel- oder Erwartungswerten der Verteilungsdichtefunktionen der Verzögerungen innerhalb der Toleranzbereiche (typische Werte in den Herstellerdaten) ausgehende Analyse liefert zumindest bei „schnellen“ Schaltungen nur wenig verlässliche Aussagen über das dynamische Schaltungsverhalten, da die zugrundeliegenden Voraussetzungen praktisch nicht erfüllbar sind. Eine derart vereinfachte Analyse hat aber gewisse Vorteile bei der Untersuchung von Schaltungen, bei denen die Verzögerungen eine untergeordnete Rolle spielen (sog. „langsame“ Schaltungen) oder zur Berechnung stationärer Signalwerte in Testschrittabellen für die Fertigungsprüfung, wenn der einwandfreie Schaltungsentwurf schon vorher nachgewiesen worden ist. Aus diesem Grunde wurden drei Programmvarianten entwickelt: für Mittelwertsimulation, für Toleranzsimulation (als Form der worst-case-Simulation) und für sogenannte Testschrittsimulation als spezielle Art der Mittelwertsimulation zur Erzeugung von Testschrittabellen. Hauptanwendungsgebiete der Programme sind gegenwärtig die Analyse von Schaltungen im Entwurfsprozeß und die Herstellung von Prüfunterlagen (Testschrittabellen) zur Fertigungsprüfung. Weitere Anwendungen, die allerdings Änderungen bzw. Erweiterungen der bestehenden Programme bedingen, zeichnen sich auf dem Gebiet der Fehlerlokalisierung ab.



2. Eingabedaten

Aus Bild 1 ist ersichtlich, daß für einen Simulationslauf drei Arten von Eingabedaten benötigt werden. Es sind dies der Bauelementekatalog, die Schaltungsdaten und die Steuerdaten.

Der Bauelementekatalog beinhaltet alle Angaben über die Bauelemente, die für die Simulation relevant sind, also die minimalen, typischen (nominalen) und maximalen Verzögerungszeiten für $1 \rightarrow 0$ - und $0 \rightarrow 1$ -Übergänge, Vorbereitungs- und Halte-Zeiten, Stiftreferenzen, Kennzeichnung des logischen Verhaltens usw. Diese Daten sind für die am häufigsten verwendeten Bauelementetypen (die Reihen SN 74-N, -L, -H und -S von TI, die Reihen D 10, und D 20 von HWF bzw. AMD) erfaßt, so daß der Anwender diese Daten nicht bereitstellen muß. Für andere logische Bauelemente bzw. bei besonderen Anwendungsfällen – z. B. bei Verwendung von fiktiven logischen Elementen, RC-Hoch- und -Tiefpässen sowie Verzögerungsgliedern, die als black boxes geführt werden und näher spezifiziert werden müssen – kann der Katalog mittels Lochkarten erweitert bzw. modifiziert werden.

Die Schaltungsdaten werden für jede Schaltung gesondert erfaßt. Durch sie werden

- die verwendeten Bauelementetypen spezifiziert (Bauelementeliste)
- die Schaltungstopologie beschrieben (Verbindungsliste) und
- die zum Steckverbinder und zu festen Meßpunkten geführten Leitungen angegeben (Verbindungsliste).

Mit Hilfe der Steuerdaten hat der Anwender die Möglichkeit, seine Testbedingungen vorzugeben. Neben der Angabe des Zeitquants (z. B. 1 ns), Standardzeiten für die Übergänge des Taktsignals (falls vorhanden) und der Informationseingänge, Forderungen an Ausgabe und Diagnosen u. ä. gehören hierzu die Initialbelegungen bestimmter Funktionselemente und der Schaltungseingänge sowie zusätzliche innere Meßpunkte, deren Potentiale mit ausgedruckt oder gezeichnet werden sollen (bei Nichtangabe treten Standardbedingungen in Kraft). Ein wichtiger Bestandteil der Steuerdaten ist die Folge der Eingangsbefehle bzw. der Testsatz. Ersterer enthält die Nummer der Taktperiode und den Zeitpunkt in diesem Takt, in denen sich die angegebene Befehlsänderung vollziehen soll. Durch Ausnutzung der oben erwähnten Standardzeiten kann der Erfassungsaufwand extrem niedrig gehalten werden. Zur Steuerung des Zeichen- und Plotterprogramms sind noch weitere Angaben notwendig. Alle abzulochenden Daten können vom Anwender auf Formblätter notiert werden, da sie formatgebunden sind und dadurch eine leichte Handhabung gewährleisten.

3. Modellierung der Signale und Schaltungen

Die Modellierung der Schaltungen und der in ihnen ablaufenden Informationsverarbeitungsprozesse entscheidet letztlich über den Wert und die Aussagekraft der Simulationsergebnisse und den zu treibenden Aufwand. Da die Genauigkeit der Ergebnisse durch den vertretbaren Programmier- und Rechenzeitaufwand begrenzt ist, wurde ein möglichst günstiger Kompromiß zwischen beiden Forderungen angestrebt.

3.1. Signalbeschreibung

Die Signalbeschreibung geschieht – wie gegenwärtig bei der Simulation digitaler Schaltungen allgemein üblich – auf der Basis von logischen, nicht von Spannungswerten. Das Flankenübertragungsverhalten der Funktionselemente wird idealisiert, d. h. die Flankendauer wird vernachlässigt (unendlich steile Flanken) und damit der Übergangsbereich mit einem nicht definierten logischen Wert aus den Betrachtungen eliminiert. Werden den Signalflanken Zeitwerte zugeordnet, sind es immer die Zeitwerte der realen Flanken bei der halben Impulsamplitude. Diese Idealisierung ist vertretbar, da die sich innerhalb einer aus Schaltkreisen einer Reihe aufgebauten digitalen Schaltung einstellende „systemeigene“ Flanke von unbedeutender Dauer ist. Bei der Mittelwertsimulation sind zwei Signalzustände (0,1) ausreichend (zweiwertige Simulation), bei der Toleranzsimulation sind mindestens drei Zustände erforderlich (0, x, 1). Der dritte Signalzustand x bezeichnet hierbei Signalwerte in unbestimmten Signalbereichen, unbekannt Initialzustände von Speichern oder unbekannt Werte beliebiger Variabler. Innerhalb der unbestimmten Signalbereiche, die als Folge der Toleranzbereiche der Verzögerungen entstehen, ist der exakte Signalwert unbekannt, er kann konstant 0 oder 1 sein oder auch beliebig oft zwischen diesen beiden Werten wechseln. Demzufolge schließt diese Beschreibung auch Hasards mit ein. Die unbestimmten Signalbereiche haben entweder die Bedeutung von Signalübergangsbereichen zwischen Signalbereichen mit definierten aber komplementären Werten oder von unbestimmten 0- oder 1-Impulsen zwischen zwei Signalbereichen des gleichen definierten Wertes. Zur richtigen Interpretation eines X-Signalbereiches ist bei der dreiwertigen logischen

Beschreibung also immer noch Information aus den angrenzenden Bereichen eines Impulszuges erforderlich. Durch mehr als drei Signalzustände (z. B. sechs in [1], neun in [7]) ließen sich zwar alle der mittels X bezeichneten Fälle separieren, die Algorithmmierung und Programmierung würden jedoch viel aufwendiger. Die dreiwertige Signalbeschreibung besitzt außerdem den Vorteil, daß eine eindeutige Abbildung auf die Werte eines dreiwertigen (nicht-Booleschen) Logiksystems existiert. Das erlaubt die formale Anwendung der Rechenregeln, die für das aus den Funktionen der Inversion (verallgemeinerte Negation), der Konjunktion und Disjunktion (in der Fassung für mehrwertige Logik) bestehende unvollständige Funktionssystem [6] gelten und die den Grundoperationen entsprechen, die für die Verknüpfung der Signalzustände auf Grund der Wirkungsweise der Funktionselemente gültig sind. Während es bei der Mittelwertsimulation also um die als Ereignisse aufgefaßten Signalübergänge und die ihnen zugeordneten Zeitpunkte geht, erfordert die Toleranzsimulation die Behandlung von Signalübergangsbereichen, die im Extremfall zu unbestimmten Impulsen entarten. Beim Durchgang von Signalen durch Funktionselemente mit toleranzbehafteten Verzögerungswerten vergrößern sich in der Regel die unbestimmten und verkleinern sich die bestimmten Signalbereiche. Ein Kriterium für die Brauchbarkeit einer Schaltung ist dadurch gegeben, daß die bestimmten Signalbereiche unter normalen Betriebsbedingungen nirgendwo eine gewisse für die einwandfreie Funktion notwendige Mindestbreite unterschreiten dürfen. Die der Toleranzsimulation auch aus Gründen der Einfachheit zugrundegelegten Voraussetzungen, nämlich das Fehlen jeglicher determinierter Abhängigkeiten zwischen den tatsächlichen Verzögerungswerten der beiden unterschiedlichen Signalübergänge eines Funktionselements, zwischen den Verzögerungen der verschiedenen Eingangssignale eines Funktionselements und zwischen ihnen sichern, daß jede Schaltung unter den für sie denkbar ungünstigsten Bedingungen analysiert wird. Die Tatsache, daß in realisierten Schaltungen immer Abhängigkeiten der genannten Art vorhanden sind, wodurch sich die Unbestimmtheit der Signale verringert, kann dann allerdings nach sich ziehen, daß manchen Schaltungen, die praktisch einwandfrei funktionieren, ein unzureichendes worst-case-Verhalten bescheinigt wird. Zur Interpretation der Simulationsergebnisse kann also zusammenfassend festgestellt werden, daß einerseits nur die mittels Toleranzsimulation nachgewiesene Richtigkeit einer Schaltung eine unbedingte Garantie für die richtige Funktion aller Schaltungsexemplare ist, und daß andererseits ein bei Mittelwertsimulation erkannter Fehler in vielen Schaltungsexemplaren auftreten wird. Die Dimensionierung einer Schaltung anhand der Ergebnisse einer Toleranzsimulation kann aus den genannten Gründen zu übertriebenen Forderungen an die Zeitparameter der Funktionselemente führen und den Einsatz von eigentlich entbehrlichen Hochgeschwindigkeitsbauelementen notwendig machen; die Dimensionierung nach den Ergebnissen einer Mittelwertsimulation kann dagegen Fehlfunktionen in einigen Schaltungsexemplaren zur Folge haben.

3.2. Schaltungsmodell

Das Schaltungsmodell besteht aus dem Strukturmodell der Schaltung und den logischen und zeitlichen Verhaltensmodellen der Funktionselemente. Das Struktur-

modell enthält die Informationen über die Typen der eingesetzten Funktionselemente bzw. Bausteine sowie ihre Verbindungen untereinander. Komponenten der digitalen Schaltungen sind gegenwärtig hauptsächlich Gatter und kleinere Funktionsblöcke (Flip-Flop, Addierer, Dekoder, Register, Zähler, Matrixspeicher), die dem Integrationsgrad nach entweder zur Kleinintegration (SSI) gehören oder die Grenze zur Mittelintegration (MSI) bilden. Die Modellierung der Schaltung und die Simulation werden je nach Art der Funktionselemente auf Gatter- und Blockniveau durchgeführt.

Die detaillierte Ermittlung des logischen und zeitlichen Verhaltens eines Funktionsblocks muß allerdings immer von der Gatterebene ausgehen, da die vom Hersteller angegebenen Daten nur als Makromodelle verwendbar sind. Diese Verhaltensmodelle werden bei der hier gewählten anwenderfreundlichen Vorgehensweise im Programm als Prozeduren fixiert, wobei funktionell ähnliche Elemente in einer Prozedur zusammengefaßt sind. Jeder hinzukommende Bausteintyp mit neuer logischer Funktion erfordert deshalb entweder die Erweiterung einer bereits vorhandenen oder eine weitere Prozedur. Eine andere, sogar flexiblere Möglichkeit, die Modellierung jedes Funktionselements aus standardisierten Grundelementen in jedem Anwendungsfall durch den Anwender selbst, wurde wegen des unzumutbaren Arbeitsaufwandes für den Anwender und der komplizierten Erfassung des exakten Zeitverhaltens verworfen.

Die logischen Verhaltensmodelle beschreiben unter Weglassung aller Zeiteigenschaften die rein logischen Beziehungen zwischen Eingangs-, Ausgangs- und Zustandsvariablen eines Funktionselements in zwei- bzw. dreiwertiger Logik je nach Simulationsart. Sequentielle Funktionselemente erfordern häufig eine detaillierte Analyse ihrer Gatter-Innenschaltung anhand der beschreibenden Übergangsfunktionen, um nicht nur die in den Datenblättern verzeichneten Normalfälle, sondern auch alle wahrscheinlichen Spezialfälle der Ansteuerung mit zu erfassen.

Die zeitlichen Verhaltensmodelle der kombinatorischen Funktionselemente beinhalten lediglich die von den Eingängen zu den Ausgängen auftretenden typischen, minimalen und maximalen Zeitverzögerungen, die der sequentiellen Funktionselemente die Zeitverzögerungen für Ansteuerung vom Takt-, Setz- oder Rücksetzeingang zu den Ausgängen, einhaltende Zeitdifferenzen zwischen unterschiedlichen Eingangssignalen (Vorbereitungs- und Haltezeit), die vorgeschriebenen Mindestimpulslängen der Takt-, Setz- und Rücksetzsignale und die schaltende Taktflanke. Die Verzögerungswerte sind allg. für die steigende und fallende Flanke des Ausgangssignals unterschiedlich (nichtlineare Verzögerungen). Als Folge dessen treten in der Praxis Verlängerungs-, Verkürzungs- oder Auslöschungseffekte beim Durchgang von Impulsen durch Funktionselemente auf, die auch vom Modell wiedergegeben werden.

Die Auswertung des logischen und zeitlichen Verhaltensmodells im Simulationsalgorithmus und die Verknüpfung ihrer Aussagen entsprechend der realen Wirkungsweise ergeben das Gesamtverhalten des betreffenden Funktionselements, das schon bei relativ einfachen sequentiellen Elementen ziemlich kompliziert sein kann. Eine einfache schematische Darstellung des Gesamtverhaltens eines Funktionselements, wie sie in der Literatur (z. B. [2]) manchmal verwendet wird, ist nur bei Gattern möglich.

4. Programme

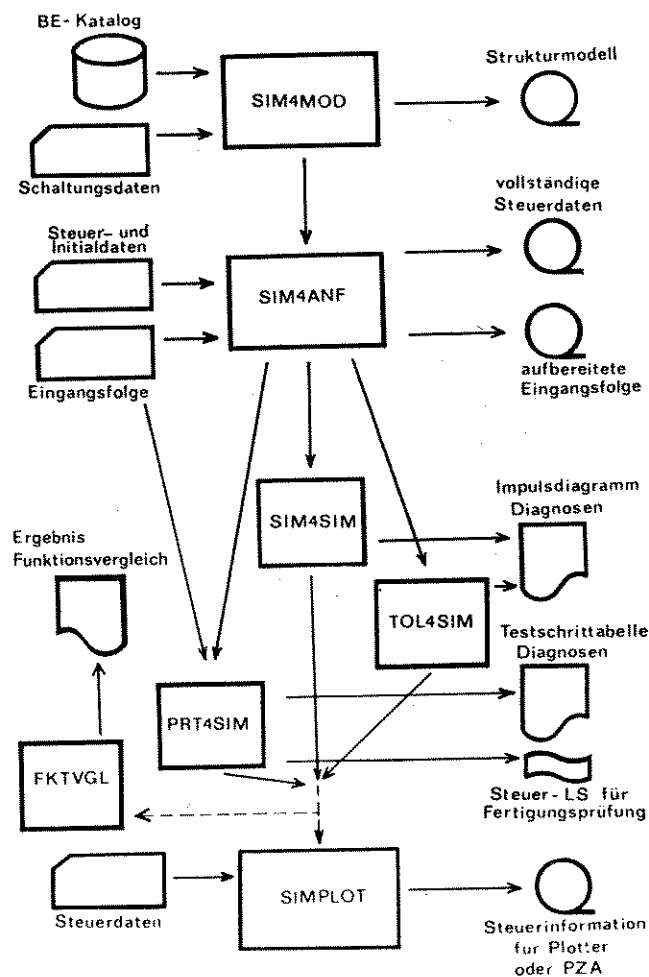
4.1. Simulationsalgorithmus

Von den bekannten Simulationsmodellen für digitale Schaltungen [3] erfüllt nur das asynchrone Simulationsmodell mit Zeitskala (Time Mapping Asynchronous Simulation Model) prinzipiell die Anforderungen, die eine echte dynamische Simulation stellt, während die anderen (z. B. Compiled-Code Model, Next-Event Synchronous Simulation) sich nur für statische Simulationen eignen. Der gewählte Simulationsalgorithmus ist eine modifizierte Variante des in [4, 5] beschriebenen rechenzeitgünstigen active path-Algorithmus, der den speziellen Gegebenheiten der Mittelwert- und Toleranzsimulation angepaßt wurde.

Die Simulation geschieht dabei ereignisorientiert unter Berücksichtigung der aktiven Schaltungspfade in einem diskreten Zeitmaßstab. Ein Ereignis ist ein Signalübergang, aktiv ist ein Schaltungspfad (Verbindung), wenn im Zeitpunkt der Beobachtung auf ihm ein Ereignis stattfindet.

Der Zeitmaßstab ist in gleich große Zeitschritte Δt (Zeitquanten) unterteilt. Die maximale Größe des Zeitquants darf gleich dem größten gemeinsamen Teiler aller erfaßten Zeitparameter der Schaltung sein. Die Abarbeitung des Algorithmus geschieht in einer sogenannten Δt -Schleife (Δt -loop), deren diskreten Zeitpunkten Ereignisse

Bild 2
Übersicht über das Programmpaket SIMPER



entsprechend ihrem Auftreten zugeordnet und mit ihrer Auswertung wieder gelöscht werden. Die Länge der Schleife ist bei synchron getakteten Schaltungen zweckmäßig gleich der Taktperiode, bei asynchronen mindestens gleich der größten Funktionselementeverzögerung zu wählen; ein vollständiger Durchlauf entspricht einem Simulationszyklus. Vor Beginn des Simulationslaufs müssen aus gegebenen Initialdaten (z. B. für die Eingangssignale und Speicherzustände) kompatible Initialen für die Ein- und Ausgänge aller Funktionselemente der Schaltung bestimmt werden.

4.2. Die Programme des Programmpaketes SIMPER

Das Programmpaket umfaßt mehrere Programme, die nacheinander bzw. wahlweise zu rechnen sind (Bild 2).

Das Programm **SIM4MOD** liest die Schaltungsdaten und erstellt daraus mit Hilfe eines Bauelementekataloges das Strukturmodell der zu simulierenden Schaltung. Es umfaßt im wesentlichen die

- Leitungs- und Verbindungsliste,
- Funktionselementliste,
- Funktionselementtypenliste,
- Schaltzeitenliste.

Dieses Programm wird auch einzeln als Datenprüfprogramm verwendet. Es erkennt einen großen Teil aller möglichen formalen Schaltungsfehler.

Das Programm **SIM4ANF** erfüllt zwei Funktionen: Es wertet die Steuer- und Initialdaten aus und ergänzt sie. Unter Verwendung des Strukturmodelles wird dabei ein statischer Anfangszustand für die Simulation erzeugt. Das ist eine vollständige Signalbelegung der Steckverbinderstifte und aller Ein- und Ausgangsstifte der einzelnen Funktionselemente der Schaltung.

Für die Mittelwert- und die Toleranzsimulation wird außerdem die Folge von Eingangssignalen gelesen und - ggf. unter Hinzufügen der Taktimpulse - in eine für die weitere Verarbeitung günstige Form umgesetzt.

Mit den Daten, die durch die vorstehend genannten Programme bereitgestellt werden, kann eine oder können mehrere der verschiedenen Varianten des Simulationsprogrammes gerechnet werden:

Die Programme **SIM4SIM** (Mittelwertsimulation) und **TOL4SIM** (Toleranzsimulation) liefern als Simulationsergebnis gedruckte Impulsablaufdiagramme; das Programm **PRT4SIM** (Testschrittsimulation) erstellt zum gegebenen Testsatz die vollständige Testschritttabelle und den Steuerlochstreifen für ein Leiterplattenprüfprogramm auf dem KRS 4200.

Das Programm **SIMPLOT** erzeugt ausgehend von einem zwischengespeicherten Simulationsergebnis (Impulsablaufdiagramme) die Steuerinformationen zum Zeichnen von Impulszeitdiagrammen (wahlweise für den Plotter IBM 1627 II oder die Präzisionszeichenanlage MK III [PZA]).

Das Programm **FKTVGL** vergleicht die zwischengespeicherten Simulationsergebnisse zweier Lösungsvarianten der gleichen Schaltung, die sich in der Struktur, den eingesetzten Bauelementtypen und ihren Zeitparametern unterscheiden können.

4.3. Programmierung

Für das Programmpaket SIMPER wurde PL/1 als Programmiersprache verwendet, einige besonders häufig aufgerufene Unterprogramme sind auch in Assemblersprache umgesetzt. Als Rechner wird gegenwärtig noch das ESER-

Vergleichsmuster IBM/360-40 (Betriebssystem OS) verwendet; die Umstellung auf das System ES 1040 wird vorbereitet.

Bei der Festlegung des rechnerinternen Schaltungsmodells und bei der Programmierung wurde dem Aspekt des Aufwandes an Rechenzeit große Bedeutung beigemessen. Besonderes Augenmerk galt dabei den zyklisch durchlaufenen Teilen des Simulationsprogrammes.

So konnte bei einer Überarbeitung u. a. durch Vermeiden bestimmter sehr aufwendiger PL/1-Befehle der Verbrauch an Rechenzeit für den Simulationszyklus trotz erweiterter Leistungen gegenüber einer älteren Version um über 80 % gesenkt werden. Gleichzeitig verringerte sich der Kernspeicherbedarf für das Simulationsprogramm um ca. 20 %, so daß auch auf eine Überlagerung der Unterprogramme verzichtet werden konnte.

Die verschiedenen Varianten des Simulationsprogrammes benutzen teilweise die gleichen Unterprogramme. Somit kann der Aufwand zur Erweiterung der Anwendbarkeit des Programmpaketes für Bauelementtypen mit weiteren logischen Funktionen relativ gering gehalten werden.

4.4. Funktionselementebasis

Mit dem gegenwärtigen Stand der Simulationsprogramme können Funktionselemente mit folgenden logischen Funktionen verarbeitet werden:

- beliebige NAND, AND, NOR und OR mit bis zu 8 Eingängen
- AND-NOR (bis zu 10 Eingänge in maximal 5 Gruppen) mit oder ohne Erweiterung durch EXPANDER
- Negation und reine Verzögerung
- aus R und C zusammengesetzte Hoch- und Tiefpässe
- JK- und RS-Master-Slave-Flip-Flop
- D-Flip-Flop und 8-Bit-Serienschieberegister.

Eine Erweiterung um andere Gatter und Funktionsblöcke ist möglich und vorgesehen (z. B. für ANTIVALENZ und verschiedene Zähler).

5. Ausgabedaten

Zu den Ausgabedaten gehören die Ergebnisdaten der Simulation und Diagnosen zum Ablauf der Simulation. Die Ergebnisdaten sind gedruckte Impulsablaufdiagramme und/oder gezeichnete Impulszeitdiagramme, für Prüfzwecke auch Testschrittabellen als Liste gedruckt und als Lochstreifen. Die Diagnosen sind Mitteilungen, Warnungen und Fehlerhinweise, die Aufschluß über Ort, Zeitpunkt und Art des vom Normalen abweichenden Schaltungsverhaltens geben, beurteilt anhand der Verhaltensmodelle der Funktionselemente.

5.1. Gedruckte Impulsablaufdiagramme

Aus den Impulsablaufdiagrammen (Tafel 1) lassen sich bei sachgerechter Interpretation alle benötigten Informationen über die simulierte Schaltung ablesen. Der Kopf wird nur einmal ausgedruckt und kennzeichnet die Bedeutung der einzelnen Druckspalten. Das sind Periodennummer, Zeitpunkt innerhalb einer Periode, Bezeichnung der Schaltungsein- und -ausgänge sowie der inneren Meßpunkte. Der Zeilenvorschub entspricht der Zeitachse. Es wird immer nur dann eine Zeile ausgedruckt, wenn sich der Signalwert an einem Ein- oder Ausgang oder einem Meßpunkt ändert. Befindet sich die Schaltung vor einem Eingangssignalwechsel in einem stationären Zustand, wird die betreffende Zeile mit einem Punkt markiert. Weitere Markierungen durch Sterne kennzeichnen

DNR.	TP	ZQT	BS/FE	KURZDIAGNOSE	SEITE 1
F 12	0	79	1/2	HALTEZEIT NLR	0 ZQ
F 12	0	91	3/2	HALTEZEIT NLR	0 ZQ
F 12	2	79	2/1	HALTEZEIT NLR	0 ZQ
F 12	2	97	3/2	HALTEZEIT NLR	0 ZQ
F 12	3	92	1/1	HALTEZEIT NLR	0 ZQ
F 11	3	109	8/1	AN O IMPULS:	28 ZQ
F 12	4	73	1/2	HALTEZEIT NLR	0 ZQ
F 12	4	98	1/1	HALTEZEIT NLR	0 ZQ
F 12	5	73	2/1	HALTEZEIT NLR	0 ZQ
F 12	6	79	1/2	HALTEZEIT NLR	0 ZQ
F 12	7	79	2/1	HALTEZEIT NLR	0 ZQ
F 12	8	92	1/1	HALTEZEIT NLR	0 ZQ
F 11	9	31	5/4	AN O IMPULS:	13 ZQ
F 11	9	31	5/2	AN O IMPULS:	13 ZQ
S 02	9	31	3/2	P I C FLANKE	ZL DICHT
F 12	9	73	1/2	HALTEZEIT NLR	0 ZQ

Tafel 2
Diagnosen

6. Erfahrungen mit den Programmen

Die Leistungsfähigkeit der Programme wurde anhand zahlreicher Schaltungen aus den Gebieten der PCM-Übertragungstechnik, der Vermittlungstechnik sowie der industriellen Steuerungstechnik getestet. Vergleiche anhand typischer Beispiele der PCM-Technik ergaben, daß durch Anwendung der Simulationsmethode der Arbeitszeitbedarf für eine Schaltungsanalyse im Entwurfsstadium bezogen auf Labormessungen einschließlich Laboraufbau auf mindestens 20 % sinkt. In den Zeitbedarf für die Simulation sind hierbei auch die Erfassung der Schaltungs- und Steuerdaten, die eventuelle Korrektur ihrer formalen Fehler und die Interpretation der Ergebnisdaten einbezogen. Ähnliche Werte ergaben sich für die rechnergestützte statt manuelle Erzeugung von Testschrittabellen der Prüftechnik.

Dem Umfang der zu simulierenden Schaltung und der Menge der Ein- und Ausgabedaten sind nur durch die Größe des verfügbaren Kernspeicheranteils, vom Programm her jedoch keine Beschränkungen auferlegt. Der Kernspeicherbedarf ist für das Programm SIM4MOD (Bildung des Strukturmodells) am höchsten. Er beträgt für eine Schaltung mit 50 IC-Bausteinen ungefähr 90 K Byte und erhöht sich für jeden weiteren Baustein um ca. 200...300 Byte.

Der Gesamtrechenzeitbedarf für die Mittelwertsimulation einer der oben erwähnten PCM-Schaltungen mit 18 Funktionselementen (5 Typen) für 12 Taktperioden von je 120 ns Dauer und einem Zeitquant von 1 ns einschließlich eines meist erforderlichen Korrekturlaufs für die Schaltungs- und Steuerdaten betrug ca. 165 s CPU-(Zentral-einheits-)Zeit, für eine andere Schaltung mit 81 Funktionselementen (5 Typen) für 32 Taktperioden von je 136 Zeitquanten der Länge 30 ns ca. 360 s CPU-Zeit. Das entspricht einem Preis von 130 Mark bzw. 280 Mark bei dem gegenwärtig festgelegten Preis-Leistungs-Verhältnis des verwendeten Rechners. Die Rechenzeiten für die drei Programmvarianten unterscheiden sich nur ganz geringfügig. Ganz allgemein steigt der Rechenzeitbedarf für eine bestimmte Schaltung von einem schaltungsabhängigen konstanten Wert an etwa linear mit der Länge der Eingangsfolge. Von wesentlichem Einfluß auf die absolute Höhe der Rechenzeit ist die Zahl der in der Schaltung stattfindenden Signalwechsel. Die Zahl der Funktionselemente ist weniger bedeutsam als die Zahl der unterschiedlichen Typen in einer Schaltung.

Die Rechenzeit zur Erzeugung des Steuerlochstreifens für die PZA ist etwas geringer als die zum Anfertigen der gleichen Plotterzeichnung. Sie ist abhängig von der Zahl der Impulsdigramme auf einem Blatt, dem Blattformat und der Zahl der zu zeichnenden Blätter. Für den letztgenannten Parameter wurde erwartungsgemäß eine fast lineare Abhängigkeit festgestellt. Beispielsweise betrug die Rechenzeit für eine Zeichnung mit 34 Impulsdigrammen auf zwei Blättern A 2 75 s CPU-Zeit, auf 4 Blättern A 2 140 s CPU-Zeit.

7. Zusammenfassung

Der Beitrag informiert über das Programmpaket SIMPER zur dynamischen Simulation digitaler Schaltungen, das im INT Berlin entwickelt wurde. Je nachdem, ob die typischen Werte der Zeitparameter oder ihre Toleranzbereiche zugrundegelegt werden, läßt sich eine Mittelwert- oder Toleranzsimulation durchführen, die Aussagen über das worst-case-Verhalten ermöglicht. Mit einer weiteren Programmvariante sind Testschrittabellen erzeugbar. Die Programme sind für fast alle Typen digitaler Schaltungen einsetzbar und gewährleisten aufgrund der relativ genauen Modellierung des Zeitverhaltens der Funktionselemente eine weitgehend wirklichkeitstreuere Nachbildung der in realen Schaltungen ablaufenden Prozesse. Die Nachnutzung des Programmpakets ist möglich.

Erfahrungsaustauschträger

Institut für Nachrichtentechnik
116 Berlin, Edisonstr. 63

Literatur

- [1] Abschlußbericht des Themas „Simulation II“. INT Berlin, 31. 5. 1975
- [2] Roginsky, W. N. Zur Theorie dynamischer diskreter Automaten. Nachrichtentechnik 17 (1967) 11, S. 411—413
- [3] Breuer, M. A. Design Automation of Digital Systems: Theory and Techniques. Vol. 1. Englewood Cliffs, N. J., Prentice Hall 1972
- [4] Ulrich, E. G. Time-sequenced Logical Simulation Based on Circuit Delay and Selective Tracing of Active Network Paths. Proc. of the 20 th Nat. Conf. of ACM, 1965, pp 437—448
- [5] Ulrich, E. G. Exclusive Simulation of Activity in Digital Networks. Comm. of ACM 12 (1969) 2, S. 102—110
- [6] Pospelov, D. A. Analyse und Synthese von Schaltsystemen. Berlin, Verlag Technik, 1973
- [7] Fantauzzi, G. An Algebraic Model for the Analysis of Logical Circuits. IEEE Trans. vol. C-23 (1974) 6, S. 576—581
- [8] Bochmann, D.; Kieser, M. Erfahrungen mit einer dialogfähigen Sprache zur Analyse logischer Schaltnetzwerke. Nachrichtentechnik 23 (1973) 4, S. 132
- [9] Wagner, S.; Voigt, H.-M. Digitale Simulation digitaler Schaltungen. msr 16 (1973) 7, S. 276—277
- [10] Jorke, G. Rechnersimulation logischer Schaltungen mit hochintegrierten Grundelementen. ZKI-Informationen 2 (1973), S. 36—40

Sozialistische

Rationalisierung

in der Elektrotechnik/Elektronik

Nachrichtentechnik - DDR

Übertragungstechnik



INSTITUT FÜR RATIONALISIERUNG DER ELEKTROTECHNIK/ELEKTRONIK

