https://doi.org/10.20378/irbo-52665

Die formale Validierung einer „Bausteintafel" delay-insensitiver Grundelemente

TILMAN REINHARDT,* Universität Erlangen, Elektrotechnik RS (Prof. K. D. Müller-Glaser)
MICHAEL MENDLER,† Universität Erlangen, Informatik VII (Prof. U. Herzog)
TERRY STROUP,‡ Labor für Grundlagen der Informatik, University of Edinburgh, UK

Beschreibungen der Bausteine in einer CAD-Bibliothek müssen erhöhte Forderungen an ihre Korrektheit erfüllen. Dies bietet Anlaß zum praktischen Einsatz — mit Rechnerunterstützung — mathematischer Theorien der Validierung.

Problemstellung

Um die Validierung von Schaltkreisentwürfen zu unterstützen, enthalten CAD-Bibliotheken ausführliche Spezifikationsdaten für die Grundbausteinen, die sie bereitstellen. Doch wirft die rechnerunterstützte Validierung von *Gesamtentwürfen* wiederum technische, dem Wesen nach mathematische Fragen zur Validierung des gewählten Satzes von *Fundamentalbausteinen* auf:

Konsistenz Sind die Beschreibungen der Einzelbausteine in sich konsistent, erfüllt also die Verhaltensbeschreibung jedes Bausteins seine Funktionsbeschreibung?

Vollständigkeit Beschreibt die Spezifikation jedes Bausteins sein Verhalten bezüglich formaler, mathematischer Maßstäbe auch vollständig?

Kohärenz Decken die Grundbausteine wirklich alle primitiven logischen Funktionen des Schaltkreisentwurfs, und zwar ohne Redundanz, ab?

Ein Satz von Grundbausteinen, der diese Anforderungen erfüllt, weist eine innere Systematik auf, die über die Eigenschaften der Einzelbausteine hinausgeht. Ein so zusammenhängendes Gefüge von Elementen nennen wir deswegen eine *Bausteintafel.* Ein bisher ungelöstes Problem ist, ob die Kernbausteine von CAD-Bibliotheken konsistent, vollständig und kohärent sind, ob sie also in diesem Sinne eine Bausteintafel enthalten.

Eine Bausteintafel für asynchrone Schaltungen

Delay-insensitiv sind diejenigen Schaltungen, deren Gatter und Leitungen beliebige aber endliche Verzögerungszeiten besizten dürfen, ohne die korrekte Funktion des Schaltkreises zu beeinträchtigen [1]. Delay-insensitive Schaltungen übertragen Daten asynchron zwischen ihren Modulen mit Hilfe von *Handshake-Protokollen,* die den Ablauf der Interaktion regeln. Wenn wir die Flanken in einem solchen Protokoll als *Ereignisse* betrachten, lassen sich die zeitlichen Verschränkungen der Handshake-Ereignisse, die das Protokoll tatsächlich zuläßt, in verschiedenen Kalkülen formalisieren.

Zwei günstige Kalküle sind der Prozeßkalkül CCS (Calculus of Communicating Systems), eine Art Automatenbeschreibung, die eine asynchrone Zusammenschaltung der Einzelmaschinen zuläßt, und die temporale Logik HML (Hennessy-Milner Logic), eine Art

*gemeinsame Zuschrift: Dienstagsclub, Informatik VII, Martensstr. 3, 8520 Erlangen.
†Teilweise durch ein Stipendium der Stevenson Foundation sowie durch die DFG, SFB 182, gefördert.
‡Teilweise durch den britischen SERC, Projekt GR/F 38808, sowie durch die DFG, SFB 182, gefördert.

Spezifikationslogik, die das Räsonieren über komplexe zeitliche Verhältnisse gestattet [2]. Mit CCS läßt sich das Verhalten und mit HML läßt sich die gewünschte abstrakte Funktion einzelner Bausteine beschreiben. Da die zwei Formalismen miteinander verträglich sind, kann man dann nachweisen, falls es auch stimmt, daß der mit CCS beschriebener Baustein die mit HML beschriebener Funktion erfüllt. Diese formale Validierung, oder *Verifikation*, läßt sich aufgrund der guten mathematischen Eigenschaften der beteiligten Kalküle am Rechner durchführen [3]. Auch weitere Eigenschaften, wie die Vollständigkeit der Beschreibung nach kanonischen Kriterien, lassen sich am Rechner verifizieren.

Eine Bausteintafel konstruieren wir für den delay-insensitiven Schaltungsentwurf, indem wir die Bausteine jeweils in CCS und HML beschreiben und die Korrektheit und Vollständigkeit nachweisen [4]. Die acht Bausteine bilden aber auch ein kohärentes Gefüge, denn sie entstehen alle durch drei Arten von systematischen Verwandlungen aus einem einzigen Merge-Element (siehe **Bild**).

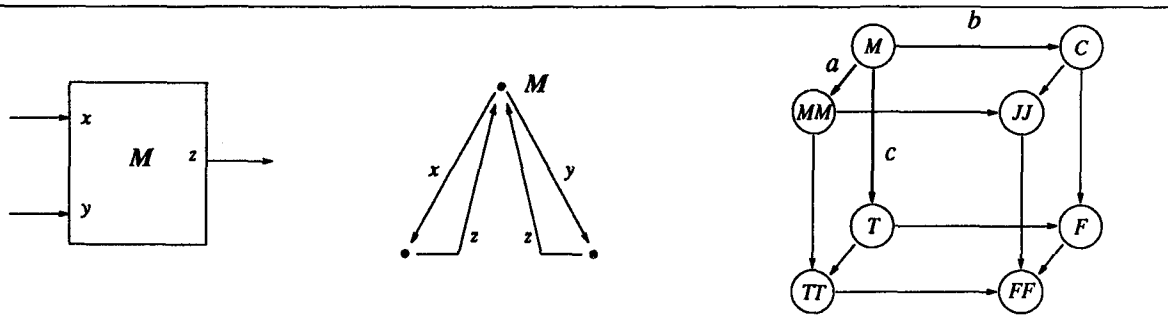

Bild. Ein einfaches Merge-Element, vergleichbar mit einem Oder-Gatter *(links)*, wird intern beschrieben als konkreter Automat mit CCS *(mitte)* und extern beschrieben durch HML-Formeln, wie etwa $\nu X.\langle x\rangle[z]X \wedge \langle y\rangle[z]X$, die die zeitliche Verschränkung von Ein- und Ausgabeereignissen festhalten. Diese zwei Beschreibungen lassen sich systematisch in drei Richtungen transformieren: (*a*) Handshake-Verfeinerung, (*b*) Sequentialität in Paralellität umwandeln, (*c*) Invertierung der Funktion. Aus der Anwendung solcher Transformationen *hintereinander* entsteht ein Satz *(rechts)* von delay-insensitiven Grundbausteinen. Drei Arten von Eigenschaften verbinden die insgesamt 16 Beschreibungen eng mit einander (siehe Text).

Das Ausgangsproblem, einen formal validierten Satz von Grundbausteinen aufzustellen, ist durch diese Untersuchungen teilweise gelöst worden. Die nicht-trivialen Fragen nach Konsistenz und Vollständigkeit konnten — mit wesentlicher Unterstützung des Rechners — bejaht werden. Doch stehen weitere Untersuchungen an. Die rein formale Wahl der Grundelemente und die Gestalt ihrer Beschreibungen müssen sich in einer realistischen Entwurfspraxis, eventuell unter Ergänzung um weitere, aus jenen zusammengesetzten Elementen bewähren. Und die systematische Ableitung aller weiteren Elemente aus dem Merge-Element muß selber formalisiert werden, damit formale Verhältnisse einzelner Bausteine zueinander verifiziert werden können. Eine einfache formale Frage wäre z.B., ob zunächst Verfeinern dann Invertieren stets das gleiche Ergebnis wie umgekehrt liefert. Eine eventuell weitreichende praktische Frage wäre, wie man solche Transformationen bei der Validierung von Gesamtentwürfen ausnutzen könnte.

[1] M. Rem. The nature of delay-insensitive computing. In *BCS Workshops in Computing, IV Higher Order Workshop*. Springer, 1991.
[2] Robin Milner. *Communication and Concurrency*. Prentice Hall, 1989.
[3] R. Cleaveland et al. The Concurrency Workbench: A Semantics Based Tool for the Verification of Concurrent Systems. Report ECS-LFCS-89-83, University of Edinburgh, 1989.
[4] T. Reinhardt et al. Konsistenz, Kohärenz und Vollständigkeit einer Bausteintafel delay-insensitiver Grundelemente. Interner Bericht, Informatik VII, Universität Erlangen, 1992.