

Schaltungsentwurf und -Layout mit Aktonalgebra

Hermann von Issendorff
D-21745 Hemmoor, Hauptstrasse 40
Email: hviss@issendorff.de

Zusammenfassung:

Prinzipiell benötigt der Aufbau jeder physikalischen Struktur genauso eine zeitlich halbgeordnete Menge von Handlungen wie die Ausführung einer DV-Aufgabe. Während sich aber die Ausführung einer DV-Aufgabe in einer Programmiersprache formal und exakt beschreiben lässt, war das bisher für den Aufbau physikalischer Strukturen nicht möglich. Was ist der Grund dafür? Offenbar liegt er darin, dass die Objekte der DV aus Daten bestehen und damit abstrakt sind, die Objekte physikalischer Strukturen dagegen konkret sind und u.a. räumliche Abmessungen haben. Die räumlichen Beziehungen zwischen den physikalischen Objekten müssen bei der Festlegung der Aufbauordnung zusätzlich berücksichtigt werden. Eine Programmiersprache, mit der sich räumliche Beziehungen beschreiben lassen, die also nicht nur wie alle Programmiersprachen eine Zeitsemantik, sondern eine Raumzeitsemantik hat, gab es aber bisher nicht.

Aktonalgebra, oder kurz AA, über die im Laufe ihrer Entwicklung hier schon mehrfach berichtet wurde, ist -soweit bekannt- die einzige Programmiersprache, die diese Eigenschaft hat. Ihre Raumsemantik ist erstaunlicherweise sehr einfach und dabei sehr mächtig: Sie beschreibt den Raum genauso, wie ein Beobachter ihn wahrnimmt, nämlich durch Unterscheidung zwischen links/rechts, oben/unten und vorne/hinten. Dies genügt für die topologische Beschreibung jeder physikalischen Struktur. Erklärt man zusätzlich die Abmessungen eines bestimmten Objektes als Masseinheiten in den drei Dimensionen, dann lässt sich ein Raster über den Raum legen, in dem jeder Abstand durch Abzählen bestimmbar ist, d.h. man erhält eine Metrik.

In z.B. [1] wurde bereits gezeigt, dass AA so allgemein und elementar ist, dass herkömmliche Programmiersprachen und mathematische Algebren in ihr ausgedrückt werden können. Mit einfachen Konversionsregeln, die eine Raumsemantik per Default hinzufügen oder von ihr abstrahieren, lassen sich DV-Programme oder mathematische Ausdrücke in AA-Ausdrücke überführen bzw. aus AA-Ausdrücken gewinnen.

Dieser Vortrag gilt in der Hauptsache der AA-Programmierung digitaler Hardware. Der Schaltungsentwurf geschieht in topologischer Strukturbeschreibung, die dann beim Layout in eine metrische überführt wird. Die in der topologischen Strukturbeschreibung enthaltenen Nachbarschaftsbeziehungen können direkt in das Layout übernommen werden. Insbesondere liefert die topologische Strukturbeschreibung die räumliche Darstellung von Kreuzungen und Rückkopplungsschleifen. Das Layout kann daher im Wesentlichen auf Drehungen und Faltungen der Schaltungskomponenten zur Anordnung der Schaltung auf einer vorgegebenen Fläche beschränkt werden.

Die klassischen Hardware-Programmiersprachen, wie z.B. VHDL oder Verilog, beschreiben demgegenüber ausschliesslich die Datenverarbeitung und sind zudem auf die relativ grobe Beschreibung des Registertransfer-Levels beschränkt. Für das Layout stehen nur die Datenbeziehungen zwischen den Hardware-Objekten aber keinerlei Strukturinformation zur Verfügung. Das Layout muss daher in einem völlig unabhängigen Entwicklungsschritt erfolgen, in dem in exponentiell aufwändigen Verfahren unter bestimmten Optimierungskriterien nach einer geeigneten Schaltungsstruktur gesucht wird.

[1] von Issendorff, H.: Algebraic Description of Physical Systems. Eurocast 2001, 8. International Conference on Computer Aided Systems Theory, Lecture Notes in Computer Science, Vol. 2178. Springer-Verlag Berlin Heidelberg New York (2001)

