

Bi-Dekomposition von Funktionsmengen in mehrwertiger Logik

Christian Lang

christian.lang@imms.de

Abstract: Logiksynthese für neue Schaltungstechniken und Data Mining sind zwei Anwendungsgebiete für die Synthese von Funktionen in mehrwertiger Logik. Bi-Dekomposition ist eine Methode der Logiksynthese, bei der die gegebene Funktion in zwei einfachere Teilfunktionen zerlegt wird. In dieser Arbeit werden Algorithmen zur Darstellung und Bi-Dekomposition von Funktionsmengen in mehrwertiger Logik entwickelt. Als Ergebnis der Implementierung dieser Algorithmen ist das modulare Programmsystem YADE entstanden, das mehrwertige Funktionsmengen in ein Netzwerk von mehrwertigen Gattern dekomponiert. Vergleiche mit anderen Dekomposern zeigen, dass durch Bi-Dekomposition erzeugte Gatternetzwerke eine geringere Komplexität aufweisen.

1 Einsatzgebiete der Bi-Dekomposition in mehrwertiger Logik

Teile und herrsche ist ein Prinzip, das in vielen Bereichen der Informatik sehr erfolgreich eingesetzt wird. Im digitalen Schaltungsentwurf ist die Bi-Dekomposition ein bekanntes Verfahren, um eine Schaltfunktion in zwei einfachere Funktionen zu zerlegen, wobei beide Teilfunktionen über ein logisches Gatter verknüpft werden. Durch rekursive Bi-Dekomposition der Teilfunktionen kann eine beliebige Schaltfunktion durch ein Netzwerk von Gattern dargestellt werden. Obwohl die Idee mehrwertiger Schaltungstechnik nicht neu ist, so ist doch die überwiegende Mehrheit der heute eingesetzten diskreten Schaltungstechnik zweiwertig. Verfahren zur Synthese von Booleschen Funktionen sind dementsprechend gut erforscht.

Fortschritte in der Halbleitertechnik verstärken das Interesse an neuen Schaltungstechnologien mit mehr als zwei Werten. Die Zahl der Verbindungsleitungen wird zunehmend ein Problem in modernen VLSI Chips. Übertragungssysteme in der Telekommunikation benutzen mehrwertige Signale, um die vorhandene Bandbreite optimal zu nutzen. Diese Verfahren werden zukünftig auch in der Mikroelektronik eingesetzt. Schnelle Speicherinterfaces nutzen vierwertige Logik, um zwei Bit pro Takt zu übertragen [Ram]. Mittels mehrwertiger Schaltungstechnik können Signale ohne Konvertierung direkt verarbeitet werden. Für die Implementierung mehrwertiger Logik gibt es eine Reihe von Schaltungstechniken, z. B. Current-Mode CMOS [TB01] und Neuron MOS [HCK01].

Bei der Miniaturisierung herkömmlicher Elektronik zeichnen sich physikalische Grenzen

ab. Zukünftige elektronische Systeme werden Quanteneffekte, wie die Supraleitung oder den Tunneleffekt nutzen [UB01, WHT01]. Derartige Schaltungstechniken sind von Natur aus mehrwertig. Für die Entwicklung solcher Schaltungen werden effektive Methoden zur mehrwertigen Logiksynthese benötigt.

Bi-Dekomposition ist die Zerlegung einer Funktion $f(A, B, C)$ in zwei Teilfunktionen $g(A, C)$ und $h(B, C)$ nach der Formel

$$f(A, B, C) = \pi(g(A, C), h(B, C)), \quad (1)$$

wobei A , B und C disjunkte Variablenmengen und $\pi(x, y)$ ein binärer Operator sind. Im Schaltungsentwurf wird der Operator $\pi(x, y)$ so gewählt, dass er direkt in der jeweiligen Technologie realisierbar ist. Es entsteht eine Schaltung, die das Gatter π sowie die Teilfunktionen $g(A, C)$ und $h(B, C)$ enthält, siehe Abbildung 1. Die vollständige Schaltung

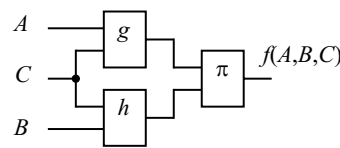


Abbildung 1: Durch Bi-Dekomposition erzeugte Schaltungsstruktur.

wird durch rekursive Bi-Dekomposition der Teilfunktionen erzeugt.

Eine zweite Anwendung findet Bi-Dekomposition in mehrwertiger Logik im Data Mining [Zu97]. Neben anderen Verfahren, wie statistischen Methoden oder neuronalen Netzen, ist Bi-Dekomposition ein Verfahren, um Klassifizierungsprobleme zu lösen. Für eine Menge von Lernbeispielen, die durch bestimmte Attribute gekennzeichnet sind, sei die Zuordnung zu einer bestimmten Klasse bekannt. Die Bi-Dekomposition kann jetzt angewendet werden, um aus diesen Beispielen die Zuordnung von neuen, in den Beispielen nicht vorkommenden Attributkombinationen abzuleiten. Die Grundlage für diese Anwendung ist das Occam Prinzip, welches besagt, dass unter einer Menge von Erklärungen für einen bestimmten Sachverhalt die einfachste Erklärung die wahrscheinlich richtige ist. Die Menge von Lernbeispielen und ihre Zuordnung zu Klassen wird als partiell definierte mehrwertige Funktion mit den Attributen als Argument und der Klasse als Funktionswert dargestellt. Per Bi-Dekomposition wird jetzt die einfachste vollständig definierte Funktion ermittelt, die in ihren Werten mit der partiell definierten Funktion übereinstimmt. Nach dem Occam Prinzip sind in dieser Funktion die Klassen mit hoher Wahrscheinlichkeit richtig zugeordnet.

Das Verfahren soll an einer vereinfachten Version eines bekannten Data Mining Beispiels demonstriert werden [BM98]. Ausgehend von bekannten Kriterien, siehe Tabelle 1, soll der richtige Typ der Kontaktlinse für den jeweiligen Patienten ermittelt werden. Die Zuordnung des Kontaktlinsentyps zu den Kriterien kann jetzt als eine Funktion in mehrwertiger Logik dargestellt werden

$$Kontakt = f(Alter, Brille, Tränenrate, Kontakt). \quad (2)$$

Tabelle 1: Kodierung der Kriterien zur Verschreibung von Kontaktlinsen.

Variable	Bedeutung	Kode	Wert
<i>Alter</i>	Alter des Patienten	0	jung
		1	keine Altersweitsichtigkeit
		2	Altersweitsichtigkeit
<i>Brille</i>	Brillenverschreibung	0	kurzsichtig
		1	weitsichtig
<i>Tränenrate</i>	Menge an Tränenflüssigkeit	0	reduziert
		1	normal
<i>Kontakt</i>	Type der Kontaktlinse	0	hart
		1	weich
		2	keine

Aus experimentellen Studien sei die Funktion f teilweise bekannt, siehe f_1 in Abbildung 2(a). Durch Bi-Dekomposition wird jetzt f in die möglichst einfachen Teilfunktionen $g()$ und $h()$ zerlegt, siehe Abbildung 2(b). Aus der Komposition dieser beiden Funktionen ergibt sich jetzt die vollständig definierte Funktion f_2 , aus der die vorher unbekanntenen Werte für die Verschreibung von Kontaktlinsen abzulesen sind.

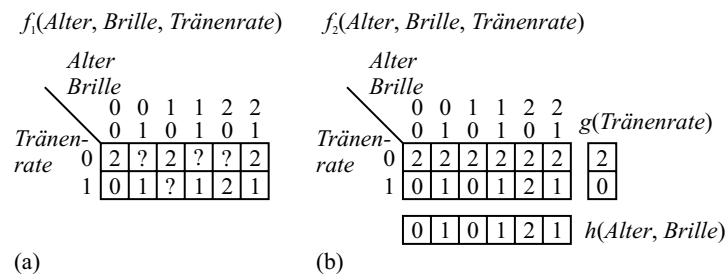


Abbildung 2: Darstellung der Regeln zu Kontaktlinsenverschreibung als mehrwertige Funktion. (a) Teilweise bekannte Funktionswerte. (b) Dekomposition der Funktion f_1 und Komposition zur Funktion f_2 .

2 Funktionsmengen

Für die Bi-Dekomposition einer Funktion gibt es in der Regel mehrere Lösungen mit unterschiedlichen Teilfunktionen. Beispielsweise existieren für die Bi-Dekomposition der Funktion $f(a, b)$, siehe Abbildung 3(a), bezüglich des modsum-Operators \oplus_3 ($x \oplus_3 y = x + y \pmod 3$) die Teilfunktionen $g_i(a)$ und $h_i(b)$, $i = 1, 2, 3$. Abbildung 3(b) zeigt die resultierenden Schaltungsstrukturen.

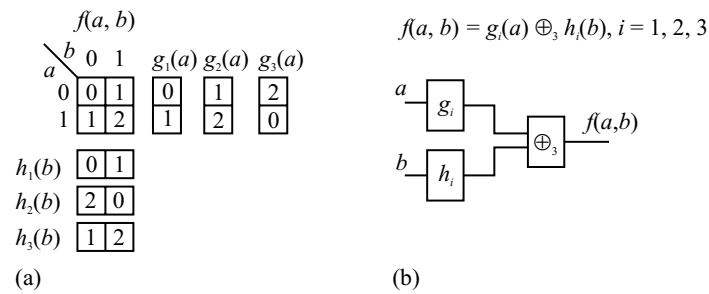


Abbildung 3: Dekomposition der Funktion $f(a, b)$ bezüglich des Operators \oplus_3 . (a) Funktion $f(a, b)$ und Teilfunktionen $g_i(a)$ sowie $h_i(b)$, $i = 1, 2, 3$. (b) Schaltungsstruktur der Bi-Dekomposition.

Wenn man nur die Bi-Dekomposition von einzelnen Funktionen betrachten würde, müsste man für die weitere rekursive Dekomposition aus der Menge der Teilfunktionen einen Vertreter auswählen, der dann weiter dekomponiert würde. Diese Auswahl ist aber problematisch, weil sie großen Einfluss auf die Komplexität der weiteren Bi-Dekomposition hat, aber keine effektiven Auswahlalgorithmen bekannt sind. Die Alternative, alle Funktionen zu dekomponieren, verbietet sich durch die mit der Dekompositionstiefe exponentiell steigende Zahl von Varianten.

Um das Auswahlproblem zu umgehen, wurde die Bi-Dekomposition für Funktionsmengen untersucht. Statt einen Vertreter auszuwählen, wurde die gesamte Menge von Funktionen dekomponiert und auf die Teilmenge der dekomponierbaren Funktionen eingeschränkt. Damit wird der durch die Funktionsmenge gegebene Optimierungsspielraum in jeder Dekompositionsstufe nur soweit, wie unbedingt notwendig, eingeschränkt. Die Auswahl der optimalen Funktion aus der Funktionsmenge wird damit verschoben, bis mehr Information verfügbar ist.

Aufgrund der hohen Komplexität von allgemeinen Funktionsmengen, sind keine effektiven Algorithmen zu deren Bi-Dekomposition bekannt. Bei der Bi-Dekomposition bezüglich eines bestimmten Operators treten nur spezielle Klassen von Funktionsmengen auf, die besondere Eigenschaften aufweisen und deshalb effektiv bi-dekomponierbar sind. Im Rahmen dieser Arbeit [La03] wurden folgende Klassen von Funktionsmengen identifiziert und untersucht:

- In *partiell definierten Funktionen* ist ein Funktionswert entweder vollständig spezifiziert (care) oder kann einen beliebigen Wert aus dem Wertebereich der Funktion annehmen (don't care). Sie sind bereits aus der Booleschen Logik bekannt und wurden in Kapitel 1 im Zusammenhang mit Data Mining Anwendungen vorgestellt.
- *Funktionsintervalle* verfeinern die Kontrolle über die Funktionswerte, indem für die möglichen Funktionswerte Intervalle von Werten zugelassen werden. Funktionsintervalle entstehen bei der Bi-Dekomposition bezüglich der Minimum- und Maximum-Operatoren.
- *Relationen* besitzen als Funktionswerte beliebige Teilmengen des Wertebereichs.

Relation finden Anwendungen beim Data Mining, wenn die Klassenzuordnung nicht eindeutig ist.

- *Funktionenverbände* zeichnen sich dadurch aus, dass Minimum und Maximum von jeweils zwei Verbandselementen wiederum Elemente des Funktionenverbands sind. Funktionenverbände sind ein wichtiges theoretisches Hilfsmittel zur Entwicklung von Bi-Dekompositionsalgorithmen.
- *Kombinierte partiell definierte Funktionen* beschreiben Mengen von Funktionen, die durch Addition einer Konstanten auseinander hervorgehen. Diese Funktionsmenge entsteht bei der Modsum-Bi-Dekomposition von partiell definierten Funktionen.

Die beschriebenen Klassen von Funktionsmengen ordnen sich in eine Hierarchie ein, wobei die allgemeine Funktionsmenge schrittweise spezialisiert wird, siehe Abbildung 4. So ist z. B. jedes Funktionenintervall gleichzeitig eine Relation, d. h. Relationen verallgemeinern die Funktionenintervalle.

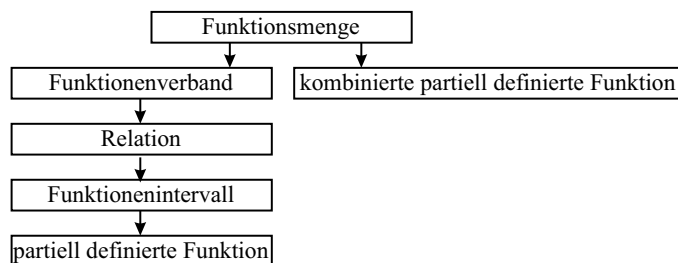


Abbildung 4: Hierarchie Klassen von Funktionsmengen.

3 Bi-Dekompositionsalgorithmen

Bi-Dekompositionsalgorithmen werden nach dem Typ des Operators und nach der dekomponierten Funktionsmenge unterschieden. Ausgangspunkt für die Entwicklung der Algorithmen waren bekannte Algorithmen für die Bi-Dekomposition Boolescher Funktionen [BDS91]. In dieser Arbeit [La03] wurden Algorithmen für folgende Operatorklassen entwickelt:

- *Monotone Operatoren* weisen nicht fallende Operatorwerte für steigende Argumentwerte auf. Sie haben besondere Bedeutung im Data Mining, da viele natürliche Funktionen, z. B. Summe oder Produkt, monoton sind. Der entwickelte Bi-Dekompositionsalgorithmus dekomponiert Funktionenintervalle bezüglich beliebiger monotoner Operatoren. Unter Ausnutzung der Monotonie werden die Funktionswerte für die Teilfunktionen schrittweise eingeschränkt, bis eine Bi-Dekomposition gefunden ist.

- Ein *Simple Operator* ist durch eine Schachtelung von if-then-else Konstrukten definiert. Durch die Auswahl und Reihenfolge der if-then-else Konstrukte ist es möglich, die Operatoren an die jeweilige Anwendung anzupassen. Relationen werden bezüglich der einfachen Operatoren bi-dekomponiert, indem die Teilfunktionswerte schrittweise aus den einzelnen if-then-else Konstrukten berechnet werden.
- Die *Minimum-* und *Maximum-Operatoren* sind mehrwertige Verallgemeinerungen der Booleschen AND- und OR-Operatoren. Sie sind sehr gut in Halbleitertechnologien realisierbar. Ein besonders effektiver Dekompositionsalgorithmus für Funktionenintervalle benutzt die mehrwertige Verallgemeinerung des Booleschen Differentialkalküls [BP81].
- Der *Modsum-Operator* ist eine mehrwertige Verallgemeinerung des Booleschen EXOR-Operators durch die aus der Zahlentheorie bekannte Addition modulo einer Konstanten. Auch im mehrwertigen Fall ist er linear, weshalb sich partiell definierte Funktionen durch das Lösen von linearen Gleichungssystemen bi-dekomponieren lassen.

Die Bi-Dekomponierbarkeit ist eine Eigenschaft einer Funktionsmenge, d. h. es gibt Funktionsmengen, die nicht bi-dekomponierbar sind. Für diese Funktionsmengen wurden Algorithmen entwickelt, die es dennoch ermöglichen, diese Funktionsmengen in einfachere Teilfunktionen zu zerlegen:

- *Multi-Dekomposition* ist eine Zerlegung einer nicht bi-dekomponierbaren Funktionsmenge in $n \geq 2$ bi-dekomponierbare Teilfunktionsmengen. Um die Zahl n der Teilfunktionen zu minimieren, wurden Graphenfärbungsalgorithmen benutzt.
- *Funktionsmengen-Separation* zerlegt eine nicht bi-dekomponierbare Funktionsmenge in eine größere (und damit einfachere) Funktionsmenge und in eine direkt bi-dekomponierbare Funktionsmenge.

4 Bi-Dekomposer YADE

Um die entwickelten Bi-Dekompositionsalgorithmen zu testen, wurde das Programmsystem YADE (Yet Another DEcomposer) erstellt. YADE dekomponiert eine gegebene Funktionsmenge in eine Netzliste von mehrwertigen Gattern. Die Funktionsmenge wird als Tabelle im ML-Format [Po] an YADE übergeben. Das Programm liefert eine SPICE ähnliche Netzliste zurück. Intern werden die Funktionsmengen als binär kodierte Entscheidungsdiagramme dargestellt [MFP⁺00], wobei das BDD (Binary Decision Diagram) Package BuDDy die Datenverwaltung übernimmt [LN]. Vergleiche mit der ersten Implementierung von YADE haben gezeigt, dass diese Datenstruktur erheblich effektiver ist als eine Implementierung durch mehrwertige Entscheidungsdiagramme. YADE ist objektorientiert in ANSI C++ geschrieben und dadurch leicht zu modifizieren.

YADE ist als Experimentierplattform ausgelegt, um verschiedene Bi-Dekompositionsalgorithmen mit unterschiedlichen Funktionsmengen zu testen. Implementiert wurden alle

in Kapitel 3 angegebenen Algorithmen. Das Programm ist in drei Packages aufgeteilt, siehe Abbildung 5. Das Package YADE enthält die Dekompositionsalgorithmen. Die Ein-

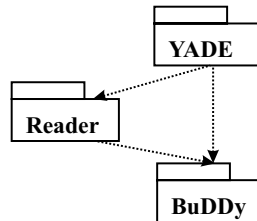


Abbildung 5: Package Struktur von YADE.

und Ausgabe ist zur besseren Modifizierbarkeit getrennt im Package Reader implementiert. Die mehrwertigen Funktionsmengen werden durch das Package BuDDy verarbeitet. Zur Anpassung an unterschiedliche Anwendungen und Optimierungskriterien ist die Bi-Dekomposition in folgenden Punkten konfigurierbar:

- Algorithmus zur Reduktion irrelevanter Variablen
- Methode zur Auswahl der Dekompositionsvariablen
- Strategie zur Selektion der Dekompositionsoperatoren
- Wiederverwendung von Teilschaltungen

Mit vorgefertigten Konfigurationen können verschiedene Optimierungsziele verfolgt werden:

- Gatterzahl
- Schaltungstiefe
- Rechenzeit

YADE kann durch Ableitung von vorhandenen Basisklassen um neue Dekompositionsalgorithmen und Funktionsmengen erweitert werden.

5 Ergebnisse und Ausblick

Die praktische Anwendbarkeit der entwickelten Algorithmen wurde durch die vollständige Dekomposition einer Reihe von Testfunktionen mittels YADE nachgewiesen. Die Testfunktionen wurden der POLO (Portland Logic Optimization Group) Internetseite [Po] entnommen und stammen aus den Bereichen Data Mining und Logikentwurf. Die Ergebnisse wurden hinsichtlich einer Reihe von Komplexitätsparametern (u. a. Gatterzahl

und Schaltungstiefe) mit anderen Dekomposern verglichen. Es zeigte sich, dass durch die Bi-Dekomposition mit YADE im Durchschnitt Netzwerke geringerer Komplexität erzeugt werden.

Die Dekomposition eines Data Mining Beispiels (Klassifizierung von Seeschwämmen [UD]) mit 76 Lernbeispielen á 44 Attributen besteht aus nur 9 Gattern. Damit wurden die Daten für die Interpretation durch den Menschen anschaulich strukturiert und es wurde eine Hypothese für die Klassifizierung weiterer Beispiele automatisch gefunden.

Um die Ergebnisse in praktischen Anwendungen einzusetzen, sind weitere Arbeiten notwendig. Für den Einsatz der mehrwertigen Bi-Dekomposition im Schaltungsentwurf müssen die technologisch realisierbaren Grundgatter bestimmt und in entsprechende Bi-Dekompositionsalgorithmen abgebildet werden. Weitere Untersuchungen sind notwendig, um die Auswahlalgorithmen für Operatoren und Variablenmengen an die gewünschten Optimierungskriterien anzupassen.

Um die vorgestellte Bi-Dekomposition im Data Mining einzusetzen, sind Tests notwendig, die zeigen, unter welchen Bedingungen die per Bi-Dekomposition gewonnene Strukturierung der Daten anderen Methoden überlegen ist. Es ist auch zu untersuchen, welche Operatoren im Data Mining besondere Vorteile bringen (z. B. bei der Interpretierbarkeit der Ergebnisse).

Literatur

- [BDS91] Bochmann, D.; Dresig, F.; Steinbach, B.: A new decomposition method for multilevel circuit design. In: *European Conference on Design Automation*. S. 374–377. Amsterdam, Holland. 1991.
- [BM98] Blake, C.; Merz, C.: *UCI Repository of machine learning databases*. University of California, Irvine, Department of Information and Computer Sciences. Irvine, USA, URL <http://www.ics.uci.edu/~mlearn/MLRepository.html>. 1998.
- [BP81] Bochmann, D.; Posthoff, C.: *Binäre dynamische Systeme*. Oldenbourg Verlag. München. 1981.
- [HCK01] Han, S.; Choi, Y.; Kim, H.: A 4-digit CMOS quaternary to analog converter with current switch and neuron mos down-literal circuit. In: *31th IEEE International Symposium on Multiple-Valued Logic*. S. 67–71. Warschau, Polen. May 2001.
- [La03] Lang, C.: *Bi-Decomposition of Function Sets using Multi-Valued Logic*. Dissertation erschienen bei dissertation.de Berlin. Technische Universität Bergakademie Freiberg. 2003.
- [LN] Lind-Nilsen, J.: *BuDDy—A Binary Decision Diagram Package*. IT-højskolen i København. URL <http://www.itu.dk/research/buddy/index.html>.
- [MFP⁺00] Mishchenko, A.; Files, C.; Perkowski, M.; Steinbach, B.; Dorotska, C.: Implicit algorithms for multi-valued input support minimization. In: *4th International Workshop on Boolean Problems*. S. 9–21. Freiberg. September 2000.
- [Po] Portland Logic and Optimization Group: *POLO Web Page*. Portland State University. URL <http://www.ee.pdx.edu/~polo/>.

- [Ram] Rambus Inc.: *Rambus Signaling Technologies—RSL, QRS� and SerDes Technology Overview*. URL <http://www.rDRAM.com/>.
- [TB01] Teng, H.; Bolton, R.: The use of arithmetic operators in a self-restored current-mode CMOS multiple-valued logic design architecture. In: *31th IEEE International Symposium on Multiple-Valued Logic*. S. 100–105. Warschau, Polen. May 2001.
- [UB01] Uemura, T.; Baba, T.: A three-valued D-flip-flop and shift register using multiple-junction surface tunnel transistors. In: *31th IEEE International Symposium on Multiple-Valued Logic*. S. 89–93. Warschau, Polen. May 2001.
- [UD] Uriz, I.; Domingo, M.: *Marine Sponges, O.HADROMERIDA (DEMOSPONGIAE.PORIFERA)*. Centre d'Estudis Abançats de Blanes (CSIC) Cami de Santa Barbara. Data archived in [BM98].
- [WHT01] Waho, T.; Hattori, K.; Takamatsu, Y.: Flash analog-to-digital converter using resonant-tunneling multiple-valued circuits. In: *31th IEEE International Symposium on Multiple-Valued Logic*. S. 94–99. Warschau, Polen. May 2001.
- [Zu97] Zupan, B.: *Machine Learning Based on Function Decomposition*. PhD thesis. University of Ljubljana. Ljubljana, Slowenia. 1997.

Persönlicher Werdegang

Christian Lang hat an der Technischen Universität Chemnitz Informationstechnik studiert und 1995 mit dem Diplom (Dipl.-Ing.) abgeschlossen. In den Jahren 1993 bis 1994 belegte er an der Western Michigan University in Kalamazoo den Masters Studiengang Electrical Engineering, den er mit einem Masters Degree (MSEE) abschloss.

Von 1995 bis 1999 war er wissenschaftlicher Mitarbeiter des Instituts für Informatik an der Technischen Universität Bergakademie Freiberg, an der er 2003 zum Dr.-Ing. promovierte. Thema der Forschungsarbeit war die dekompositorische Synthese von Booleschen und mehrwertigen Funktionen.

Seit 2000 arbeitet Herr Lang als Themengebieteileiter für Präzisions- und Optoelektronik sowie Entwurfsmethodik am Institut für Mikroelektronik- und Mechatronik-Systeme (IMMS gGmbH) in Erfurt. Hier beschäftigt er sich mit der Entwicklung von Ausleselektronik für Sensoren und schnelle optische Speichermedien. Ein weiterer Schwerpunkt seiner Arbeit liegt in der Anwendung der formalen Schaltungsverifikation auf mixed-signal Schaltungen und Hardware-/Softwaresysteme, sowie in der symbolischen Analyse von analogen Schaltungen.

