



Zum Entwurf komplexer digitaler integrierter Schaltungen sind spezielle formale Sprachen verfügbar, die im Gegensatz zu herkömmlichen Programmiersprachen insbesondere in der Lage sein müssen, die übliche massiv parallele Datenverarbeitung in Hardware zu beschreiben. Die am weitesten verbreiteten **Hardwarebeschreibungssprachen (HDLs = hardware description languages)** zur Beschreibung digitaler Schaltungen sind **VHDL (very high speed integrated circuit HDL)** und **Verilog HDL**.

Der Umsetzung von Programmcode in einer Hochsprache in die Maschinsprache des Zielrechners durch einen **Compiler** entspricht beim Entwurf digitaler Schaltungen mit HDLs die Umsetzung des HDL-Code in die Fertigungsdaten für die integrierte Schaltung. Diese Umsetzung erfolgt weitgehend automatisiert mittels spezieller Software und wird auch **silicon compilation** genannt. Bild 1 zeigt schematisch den Ablauf dieses Prozesses.

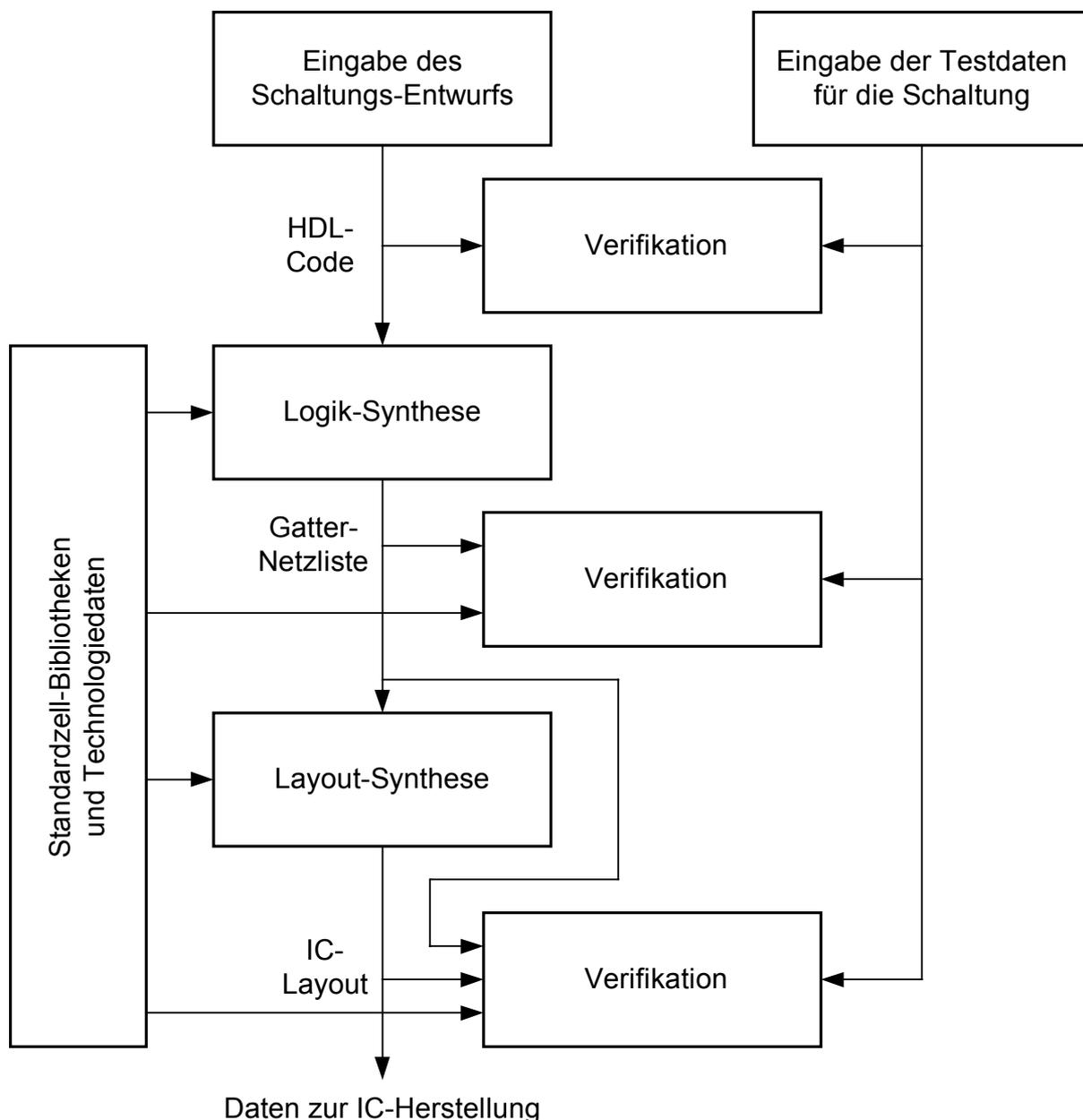


Bild 1: Schematische, vereinfachte Darstellung eines HDL-basierten Synthese-Designflows



## Entwurfsablauf (*design flow*) eines digitalen IC bei Verwendung einer HDL

1. **Eingabe** der Beschreibung der zu realisierenden Schaltung (*design entry*). Der HDL-Code wird im primitivsten Fall mit einem beliebigen Text-Editor eingegeben. In der Praxis werden spezielle Text-Editoren verwendet, die den HDL-Code erkennen und z.B. Schlüsselworte hervorheben und bestimmte Fehler schon bei der Eingabe markieren. In manchen Fällen werden auch Entwurfs-Werkzeuge (*design tools*) verwendet, die andere Beschreibungsformen für digitale Schaltungen (Wahrheitstabellen, Zustandsdiagramme, Petri-Netze, ...) unterstützen. Aus diesen Beschreibungsformen wird dann automatisch HDL-Code erzeugt. Dieser ist in der Regel für Menschen aber kaum lesbar.
2. **Eingabe** der Beschreibung von Eingangssignalen, mit denen das richtige Verhalten der Schaltung untersucht werden kann (*stimuli*).
3. **Verifikation** der HDL-Beschreibung. Dieser Schritt erfolgt meist durch **Simulation** der logischen Funktion der durch den HDL-Code beschriebenen Schaltung bei Anregung mit den eingegeben Stimuli und Vergleich der vom Simulationsprogramm berechneten Ausgangssignale mit den bekannten „richtigen“ Ausgangssignalen.
4. **Umsetzung** des HDL-Code in eine (prinzipiell ebenfalls lesbare) **Gatternetzliste** unter Verwendung von Zellbibliotheken. Diese **Logiksynthese** setzt die Beschreibung der Schaltung in HDL-Code um in eine Netzliste, die nur noch vorgegebene Funktionsblöcke, die sogenannten **Standardzellen**, verwendet. Dieser Schritt erfolgt meist weitgehend automatisch in mehreren Teilschritten. Hierbei wird im Rahmen **umfangreicher Optimierungsschritte** u.a. eine Minimierung der zu realisierenden Schaltfunktionen vorgenommen. Weiterhin werden zusätzliche Gatter und Leitungen eingefügt, um sicherzustellen, dass die gefertigten ICs einfach auf Fertigungsfehler getestet werden können.
5. **Verifikation** der nun durch die Netzliste beschriebenen Schaltung unter Verwendung der vorhandenen Stimuli und der **Simulationsmodelle** und **Simulationsparameter** für die Standardzellen aus den Zellbibliotheken. Hierbei kann die **Signalverzögerung** durch die **Schaltzeiten** der Standardzellen berücksichtigt werden.
6. **Umsetzung** der Netzliste in das Layout der herzustellenden Schaltung. Diese **Layoutsynthese** platziert unter Verwendung von Layoutdaten aus den Zellbibliotheken die geometrischen Strukturen, die zur Herstellung der in der Netzliste enthaltenen Standardzellen notwendig sind, auf der zur Verfügung stehenden Chipfläche (*placement*) und stellt die erforderlichen elektrischen Verbindungen her (*routing*).
7. **Verifikation** der durch die Layoutdaten beschriebenen Schaltung. Nun ist eine Simulation des Schaltungsverhaltens unter Berücksichtigung der **Signalverzögerung** durch die **Verbindungsleitungen** zwischen den Standardzellen möglich.
8. Die verifizierten Layoutdaten dienen zur Herstellung der Belichtungsmasken (*reticles*) für den photolithographischen Herstellungsprozess der ICs.

Der oben geschilderte *design flow* gilt mit Änderungen auch für die Realisierung von ICs mittels **Gate Arrays** und **FPGAs** (*field programmable gate arrays*). Im ersten Fall werden nur Layoutdaten für die Verbindungsleitungen zwischen den vorgefertigten Gattern erzeugt. Im zweiten Fall wird statt der Layoutdaten eine Datei mit Konfigurationsinformationen für das ausgewählte FPGA erzeugt, die in das FPGA geschrieben werden.