

2 Grundlagen Digitalentwurf und Prototyping

In diesem Kapitel wird auf Komponenten und Prinzipien eingegangen, auf die diese Arbeit aufbaut. Zunächst werden die Etappen beim Entwurf von digitalen Schaltungen erläutert. Daran schließt sich die Vorstellung der Arbeitsschritte beim Entwurf wie Synthese, Simulation und Timinganalyse an. Dabei werden die genutzten Designtools vorgestellt. Das realisierte Funktionsmuster basiert auf der PROMetheus Prototyper Familie. Diese wird abschließend vorgestellt.

2.1 Top-Down-Entwurf

Der Entwurf einer integrierten Schaltung aus einer umgangssprachlichen Funktionsbeschreibung läuft in mehreren Etappen ab. Jede dieser Etappen befindet sich auf einer anderen Abstraktionsebene und benutzt verfeinerte Modelle der digitalen Schaltung. Dies macht die Komplexität des Entwerfens beherrschbar, da Prüfung und Entwurf vereinfacht werden. Es ist jeweils nur ein Teilbereich der Forderungen zu erfüllen. Vor dem Erstellen der nächste Hierarchiestufe wird die Beschreibung auf Fehler geprüft und notfalls korrigiert. Die Überführung in die nächste Abstraktionsebene wird als Synthese bezeichnet.

Für die Modellierung der Hierarchiestufen sind verschiedene Beschreibungsmittel entwickelt worden. Dazu zählen die Hardwarebeschreibungssprachen (Hardware Description Language = HDL). Die zur Zeit am häufigsten genutzten sind Verilog und VHDL. In dieser Arbeit wurde VHDL verwandt.

2.1.1 Algorithmische Ebene

Neben dem Verhalten der Ein- und Ausgänge werden nebenläufige Algorithmen angegeben, mit deren Hilfe die gewünschte Funktion erreicht wird. Es gibt keinen Bezug zur späteren Hardwarestruktur der Realisierung. Als Datentypen können Fließ- und Festkommadarstellungen gewählt werden. Zu den Elementen der algorithmischen Ebene zählen Prozesse, Funktionen und Prozeduren.

2.1.2 Register-Transfer Ebene

Bei der Beschreibung auf Register-Transfer Ebene (kurz RT-Ebene, RT-Level oder RTL) wird eine Schaltung durch grundlegende Operationen und den Transfer von Daten und Steuersignalen zwischen Registern dargestellt. Typische Beschreibungselemente sind Register, Multiplexer, Zähler und Addierer. Eine RTL Beschreibung kann meist automatisch durch Softwaretools zu einer Gatterbeschreibung synthetisiert werden.

2.1.3 Gatterebene

Auf der Gatterebene werden Schaltnetze und Schaltwerke betrachtet. Der einzige Datentyp beschreibt mehrwertige Logik, Vektoren sind in einzelne Signale aufgespalten. An Grundelementen stehen kombinatorische Gatter (OR, XOR, AND, INV) und Register zur Verfügung. Verzögerungen zwischen den Gattern und Schaltzeiten rücken stärker in den Mittelpunkt der Betrachtung. Synonym wird die Gatterebene auch als Logikebene oder als GTL bezeichnet.

2.1.4 Technologieebene

Mit Technologie ist die konkrete Realisierung des Designs als Chip gemeint. Zwei Hauptklassen unter den IC's sind die anwendungsspezifischen integrierten Schaltungen (Application Specific Integrated Circuit = ASIC) und die programmierbare Logik (Programmable Logic Device = PLD). Zu den PLD's zählen die konfigurierbaren Gatternetze (Field Programmable Gate Array = FPGA). Für ASIC's muss der Designer zumindest die Verbindungen zwischen den einzelnen Zellen als Masken entwerfen, die Erstellung des Designs erfordert einen Hersteller mit entsprechender Ausrüstung (Reinraum, Belichter etc.). Bei den PLD Technologien sind die Verbindungskanäle als Schaltmatrizen vorgefertigt. Die Verdrahtung der Zellen erfolgt durch konfigurierbare Grundzellen. Eine Einführung in die verschiedenen FPGA Familien bietet [5].

2.2 Entwurfstätigkeiten und -werkzeuge

2.2.1 Zieltechnologie

Für diese Diplomarbeit wurden ausschließlich FPGA's der Serie Altera Flex10K genutzt [11]. Das entwickelte Design ist in einem EPF10k40 IC realisiert. Dieser bietet ein Gatteräquivalent von 40000.

2.2.2 RTL Synthese

Die RTL Synthese setzt die Beschreibung des Designs aus der Register-Transfer Ebene in eine Gatternetzliste um. Diese beschreibt alle benötigten Grundgatter und die elektrischen Verbindungen zwischen diesen. Die RTL Synthese setzt sich zusammen aus:

- a) **Syntax- und Konsistenzcheck**
- b) **Ausflachung und Optimierung**
- c) **Fitting und Mapping**

Der Syntax- und Konsistenzcheck analysiert die HDL Beschreibung. Dabei führen zum Abbruch unter anderem Verstöße gegen den Syntax der HDL, die Verwendung nicht synthesesfähiger Konstrukte und Fehler in der Beschreibung der Zustandsautomaten. Mit einer Warnung werden funktionslose Elemente quittiert, beispielsweise blind endende Signalwege. Anschließend wird die HDL Beschreibung in eine interne Datenbank überführt.

Das folgende Ausflachen des Designs (Hierarchie Flattening) erleichtert die Optimierung. Dabei werden die Grenze der Funktionsblöcke aufgelöst. Die Optimierung erzeugt boolesche Gleichungen.

Fitter und Mapper passen das Design an die Zieltechnologie an. Für FPGA's ordnet der Fitter den einzelnen Segmenten boolesche Ausdrücke zu, bei PAL oder GAL's erzeugt er die OR und AND Matrizen. Er benötigt im Gegensatz zum Mapper keine typspezifischen Synthesebibliotheken, sondern lediglich Kenntnis über die interne Struktur. Der Mapper setzt die booleschen Ausdrücke in eine Netzliste für eine bestimmte FPGA um.

Eine detaillierte und an CPLD's und FPGA's ausgerichtete Darstellung der Synthese findet sich in [21].

2.2.3 Place and Route

Mit diesem Schritt vollzieht sich der Übergang zur realen Hardware. Zuerst werden die Grundschaltungen auf dem Chip platziert, Place, und anschließend elektrisch verbunden, Route. Dies

erfordert die Kenntnis der Herstellungstechnologie, man spricht vom technology mapping. Beim ASIC Entwurf sind die Transistorschaltungen der Grundelemente auszuwählen und die Metallisierungsebenen vorzugeben. Die Umsetzung der Gatter bedeutet bei einer FPGA Realisierung die Belegung der Logikzellen. Die Verbindung dieser erfolgt durch Zuweisung der konfigurierbaren Leitungskanäle.

Das Place and Route (kurz: P&R) wird von Randbedingungen gesteuert, engl.: constraint driven. Dazu zählen:

- Taktfrequenz
- Chipfläche
- kapazitive Lasten und Fan-Out der Ausgangspins
- Laufzeit kritischer Signale

Zum Teil beeinflussen diese Randbedingung auch die dem P&R vorangestellten Syntheseschritte wie beispielsweise das Mapping. Das P&R versucht die Randbedingungen zu erfüllen und bricht die Optimierung beim Erreichen dieser ab. Somit ist der fehlerfreie Betrieb mit dem geforderten Takt garantiert, das erzeugte Routing muss aber nicht das Optimale sein. Ergebnisse mit der höchsten Taktfrequenz werden durch overconstraining ermittelt. Dazu wird im ersten Schritt eine nicht erreichbare Taktperiode vorgegeben, die zu einem Abbruch des P&R führt. Die bei diesem erfolglosen Lauf vom Synthesetool ermittelte Taktfrequenz wird schrittweise in beide Richtungen variiert. Die höchste Taktvorgabe mit einem erfolgreichen P&R ist dann die höchste für dieses Design und Technologie.

Bei dem computergestützten Place and Route kommen heuristische Verfahren, zum Beispiel simulated annealing, zur Anwendung. Dadurch können auch bei gleichen Eingangsdaten unterschiedliche Ergebnisse entstehen.

Nach dem Place and Route steht die konkrete Realisierung der elektrischen Verbindungen fest. Daraus lassen sich die Verzögerungszeiten für alle Pfade ermitteln. Das ist die Voraussetzung für den folgenden Arbeitsschritt.

2.2.4 Statische Timinganalyse

Bei der Statischen Timing Analyse (STA) wird die maximale Taktfrequenz des synchronen Designs ermittelt. Ausgangspunkt der Berechnung ist der Signalfluss zwischen zwei Register. Nach Eintreffen der Taktflanke schaltet der Ausgang des Quell-Registers mit Ablauf der Registertransferzeit t_{Transfer} . Nach Durchlaufen der Verbindungsstrecke in $t_{\text{Kombinatorik}}$ erscheint das Signal am Eingang des Ziel-Register. An diesem Knoten muss es mindestens t_{SETUP} lang stabil anliegen, bevor die nächste Taktflanke das Register schalten lässt.

Die zulässige Verzögerungszeit eines Pfades ist somit [Bild 1 , oberer Teil]:

$$t_{\text{Transfer}} + t_{\text{Kombinatorik}} + t_{\text{Setup}} < \text{Taktperiode}$$

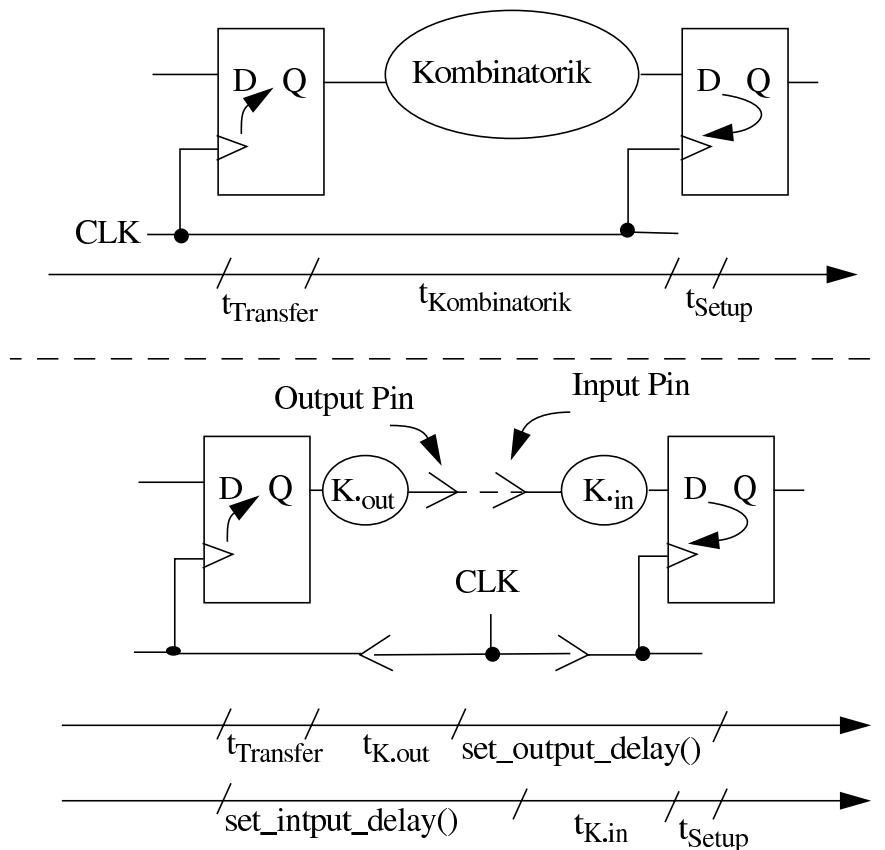


Bild 1 Bestimmung maximaler Takt

Die hochste zulassige Taktfrequenz einer integrierten Schaltung ist die des langsamsten Pfades. Dieser wird auch als kritischer Pfad bezeichnet.

Der Berechnung von Pfaden, die ber Chipanschlsse verlaufen, dienen die Randbedingungen output- und input-delay. Mit diesen wird der STA die Signallaufzeit auf dem auerhalb des Designs verlaufenden Pfades mitgeteilt. Diese ist nicht nur von FF- und Kombinatorik Parametern abhangig, die Verzogerung an den Anschlusstreibern und die Verbindung zwischen den IC's tragen ebenfalls zur Gesamtlaufzeit bei (Bild 1 , unterer Teil). Die zulassige Verzogerungszeit eines Outputpfades ist:

$$t_{\text{Transfer}} + t_{\text{KombinatorikOut}} + \text{set_output_delay}() < \text{Taktperiode}$$

und fur einen Inputpfad gilt:

$$\text{set_input_delay}() + t_{\text{KombinatorikIn}} + t_{\text{Setup}} < \text{Taktperiode}$$

2.2.5 Simulation

Mit der Simulation wird die jeweilige Implementierung und deren Spezifikation fur diesen Schritt auf gleiches Verhalten uberpruft. Sie benotigt Stimuli, eine Beschreibung des Designs und einen Vergleich mit Referenzwerten. Stimuli und Vergleich werden auch als Testbench bezeichnet. Diese wird aus der Spezifikation abgeleitet. Stimmt die aus den Stimuli erzeugte Antwort der Designbeschreibung nicht mit den Referenzwerten uberein, erzeugt die Simulation eine Fehlermeldung. Zusatzlich dazu ist der zeitliche Verlauf in- und externer Signale auch als

Wavediagramm darstellbar. Stimuli, Vergleiche und Design werden meist in einer Hardwarebeschreibungssprache erstellt.

Abhängig von der Abstraktionsstufe der Designbeschreibung ist eine Unterteilung in die funktionale und in die Timingsimulation (Post-Layout-Simulation) möglich. Die funktionale Simulation testet Modelle aus der RTL und GTL Ebene. In der Post-Layout-Simulation liegt das Design als backannotierte Gatterliste vor. Für diese Liste werden die Gatterverzögerungen, die Laufzeiten der Verbindungen und das Timingverhalten der Register aus dem Place and Route in die Gatterliste zurückgeschrieben (engl. backannotation). Die Timing-Simulation testet somit nicht nur die Funktion der Gatter, sondern auch ihre Antwortzeiten und die Leitungsverzögerungen. Auf diese Weise wird das genaue Zeitverhalten der gesamten Schaltung überprüft. Die nötige Rechenzeiten für eine RTL, GTL und Post-Layout-Simulation steigen jeweils um zirka eine bis mehrere Größenordnungen. Beispielsweise benötigt die Post-Layout-Simulation des Moduls E1IF aus Abschnitt 7.2 auf einer Sun Ultra2 für die Berechnung der ersten 400 ms Echtzeit zirka 9 Stunden [18].

Die Nutzerschnittstelle der HDL Simulatoren ist dem von Debuggern in der Softwareentwicklung nachempfunden. Sie erleichtert durch die komfortable Festlegung von Unterbrechungspunkten und den Zugriff auf interne Zustände die Fehlersuche. Diese Funktionalität ist im Abschnitt 3.3.1 detailliert dargestellt.

2.2.6 Designflow

Der bei dieser Arbeit genutzte Designflow entspricht weitgehend dem eben beschriebenen. Er ist typisch für den Entwurf einer FPGA Realisierung. Die genutzten CAE Tools und die einzelnen Arbeitsschritte stellt Bild 2 dar.

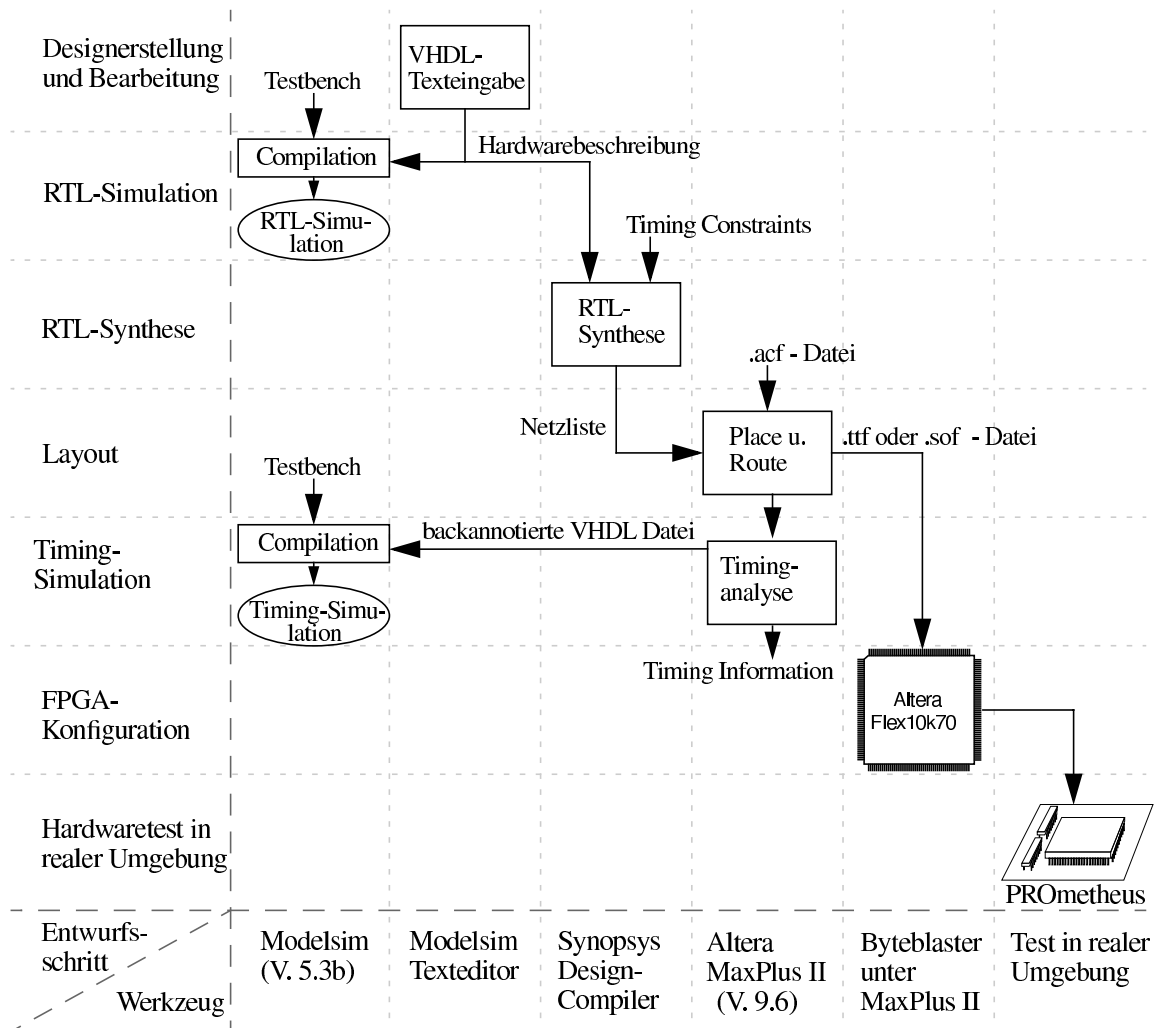


Bild 2 Designflow, nach [15]

Der Entwurf beginnt mit der Erstellung oder Modifikation der Verhaltensbeschreibung in VHDL. Anschließend wird die Systemumgebung als Testbench in VHDL modelliert und mit dem Simulator Modelsim V 5.4 der Firma Model Technology Incorporated simuliert. An die Prüfung auf Korrektheit schließt sich die automatische RTL-Synthese an.

Diese Synthese erfordert Vorgaben zum Zeitverhalten (Timing constraints) und der zu erzeugenden Struktur (synthesefähige Hardwarebeschreibung). Da bereits die Verhaltensbeschreibung ausschließlich synthesefähige VHDL Konstrukte benutzt, kann diese unverändert für RTL-Simulation und RTL-Synthese benutzt werden. Als Synthesetool kam der Synopsys Designcompiler zur Anwendung.

Jetzt folgt das P&R mit dem Backendtool MaxPlus II der Firma Altera. Dieses benötigt eine .acf Datei und die von der Synthese erzeugte Netzliste. Die .acf-Datei (Assignment and Configuration File) kontrolliert unter anderem die nicht durch die Designfunktionalität vorgegebenen Eigenschaften der FPGA. Das sind beispielsweise die Signalbelegung der Anschlüsse und Verhalten der I/O-Treiber (z.B. slew rate [Flankensteilheit]). Beim Place and Route entstehen die .tft und .sof Dateien sowie die technologieabhängigen Timinginformationen (backannotierte VHDL Datei). Daran schließt sich die Timing-Simulation an. Diese gestattet die Überprüfung

Literaturverzeichnis

- [1] Thomas Kropf: VLSI-Entwurf: Vorgehen, Methoden, Automatisierung. International Thomson Publishing, Bonn, 1995
- [2] Prof. Dr. Ing. A. Auer, Ralf Kimmelman: Schaltungstest mit Boundary-Scan. 1996
- [3] Altera Corporation: High Speed Board Designs - Application Note 75. Dez. 1999
- [4] Altera Corporation: Quartus SignalTap User's Guide.
- [5] Markus Wannemacher: Das FPGA-Kochbuch. International Thomson Publishing, Bonn, 1998
- [6] J. Haufe, C. Fritsch, M. Gulbins, V. Lück, P. Schwarz: Echtzeit-Debugger für digitale integrierte Schaltungen. Elektronik H. 22/2000 S. 132 ff.
- [7] Xilinx Corporation: ChipScope software and ILA cores user manual v2.0, 2000
- [8] Alfred L. Crouch: Design-for-test for digital IC's and embedded core systems. Prentice Hall PTR, 1999
- [9] Xilinx Corporation: Using the XC4000 Readback Capability - Application Note XAPP015.
- [10] Xilinx Corporation: Virtex FPGA Series Configuration and Readback - Application Note XAPP138. Oct. 2000
- [11] Altera Corporation: Flex 10K Data Sheet. Mai 2000
- [12] Fraunhofer Gesellschaft, Institut für Integrierte Schaltungen: PROMetheus SimConnect - User Manual. Dresden, 1999
- [13] Chip Design Dresden: Prototypingboard PROMetheus I. Dresden, 1999
- [14] H. Herzog: PROMetheus 4 - Prototyping nicht nur für Mikrocontroller. 5. Design&Elektronik Entwicklerforum Programmierbare Logik, München 23. Juni 1998, Begleittexte, 134-139
- [15] Andre Schulze: Entwicklung eines PCI-Interfaces für ein FPGA-basiertes Prototypingboard unter Benutzung eines Standard-PCI-Controllers. Dresden, 2000
- [16] Atmel Corporation: EMC Design Considerations - Application Note AVR040.2000

-
- [17] Galvantech Inc.: Asynchronous SRAM GVT 75312A8 - Datasheet. 1999
- [18] Rainer Ulrich, Wolfgang Grafen, Jürgen Haufe, Jens Grosse: Debugging of FPGA based Prototypes - A Case Study. 2000
- [19] G. Koch, U. Kebschul, W. Rosenstiel: Breakpoints and Breakpointdetection in Source Level Emulation. ACM Transactions on Design Automation of Electronic Systems (TODAES), Vol. 3, No. 2, April 1998
- [20] J. Haufe; P. Schwarz, T. Berndt, J. Große: Accelerated Logic Simulation by Using Prototype Boards. Proc Design, Automation and Test in Europe Conference, Paris 1998, Designer Track, 183-189
- [21] P. Heusinger, K. Ronge, G. Stock: Handbuch der PLD's und FPGA's. Poing 1994
- [22] J. Haufe: Aufbau eines Hardware-Debuggers, Realisierungsstudie. Dresden 2000
- [23] Martin Speitel: Verifizieren statt Simulieren. Elektronik H. 12/1998, S. 86 ff
- [24] Altera Corporation: Apex 20K Data Sheet. März 2000
- [25] Altera Corporation: Metastability in Altera Devices- Application Note 42. Juni 1996
- [26] <http://www.ikos.com/products/virtualwires.pdf>
- [27] Xilinx Corporation: Application Note XAPP 045 - Voltage/Current Characteristics of XC4000-Family Outputs. November 1997
- [28] Xilinx Corporation: FPGA Products. November 1997
- [29] Georg Archer: Entwicklung eines SCI-Knotens zur Kopplung von PCI-basierten Arbeitsplatzrechnern mit Hilfe von VHDL. Diplomarbeit TU München, Institut für Informatik 1996