

VHDL

Einführung

**B
U
L
M
E**  **Höhere Technische
Bundes-Lehr- und
Versuchsanstalt
BULME Graz – Götting**

V1.0

F. Wolf

Graz, Jänner 2002

Inhaltsverzeichnis

1	<i>Der Begriff VHDL</i>	1
1.1	Beschreibung mit VHDL	2
1.2	Geschichtliche Entwicklung von VHDL	2
1.3	Design-Methodik mit VHDL	3

VHDL

Einführung

1 Der Begriff VHDL

VHDL ist eine Hardwarebeschreibungssprache, mit der Schaltungsdesigner digitale Schaltungen und Systeme durchgängig von Definition, Verifikation bis Konstruktion, praktisch von der Systemdefinition bis zum Schaltungsentwurf, beschreiben und verifizieren können.

VHDL steht für:

Very high speed integrated circuit
Hardware
Description
Language

Warum VHDL ?

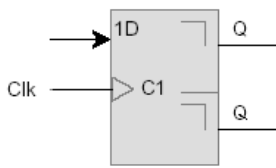
Steigende Integrationsdichte und Komplexität
erfordern neue Entwurfsmethoden

VHDL soll sein:

- Einheitliches und standardisierte Beschreibungsmechanismus für komplexe elektronische Systeme, zwecks Wiederverwendbarkeit, Austauschbarkeit und Archivierung von Modellen.
- Sie soll abstrakt genug sein, um funktionale Spezifikation zu erstellen, und ausführbar sein, um die Funktion mittels Simulation zu verifizieren.
 - VHDL ist eine höhere Programmiersprache mit für die Beschreibung von Merkmalen. (Zeitverhalten, Kommunikation von Prozessen über Signale, paralleles Arbeiten von Prozessen).

VHDL Beschreibung eines Designs besteht meist aus vier Teilen:

Package	Typen, Funktionen, Prozeduren, Komponenten, Konstanten
Entity	Schnittstellenbeschreibung (I/O-Signale und Attribute)
Architecture	Strukturelle Beschreibung oder Verhaltensbeschreibung
Configuration	Zuordnung der Architekturvarianten und der Submodule der

Beispiel:

```
Library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;
use IEEE.std_logic_components.all;
```

```
entity NAND is
port(
    a, b:    in std_logic;
    y      :  out std_logic);
end NAND;
```

```
architecture structural of NAND is
begin
    y <= not (a and b)
end structural;
```

```
configuration NAND_CFG of NAND is
    for structural
    end for;
end structural_CFG;
```

1.1 Beschreibung mit VHDL

Die Entwicklung auf dem Gebiet der elektronischen Systeme wird durch steigende Komplexität und höhere Integrationsdichten gekennzeichnet. Der Entwurf immer komplexer werdender Systeme kann nicht länger mit herkömmlichen Methoden beherrscht werden. Ebenso ist eine Verifikation durch Simulation auf Logikebene mit vertretbarem Zeitaufwand nicht mehr durchführbar. Strukturiertes Top-Down-Design unter Einsatz von Synthese ist wesentliches Hilfsmittel, das anstehende Komplexitätsproblem zu lösen.

Dazu ist es erforderlich, das zu entwerfende System bereits auf einer abstrakten Ebene mit Hilfe einer Hardwarebeschreibungssprache wie z.B. VHDL zu beschreiben.

Die wichtigsten Aufgaben einer Hardwarebeschreibungssprache sind:

- Entwurf komplexer Systeme / Synthese
- Simulation komplexer Systeme
- Wiederverwendung / Dokumentation
- Datenaustausch (Normierung der Beschreibungssprache)

Die Hardwarebeschreibungssprache VHDL erfüllt diese Forderungen.

1.2 Geschichtliche Entwicklung von VHDL

Die Entwicklung von VHDL (VHSIC Hardware Description Language) basiert auf einer Initiative des amerikanischen Verteidigungsministeriums, das durch eine einheitliche Sprache zur Dokumentation die enormen Kosten für Wartung und Weiterentwicklung von elektronischen Systemen reduzieren wollte. Nach Vorversuchen wurde 1983 der Auftrag an Intermetrics, IBM und Texas Instruments erteilt, eine entsprechende Sprache mit ersten Programmen zur Unterstützung zu entwickeln.

Im August 1985 wurde die erste Version vorgestellt und im Februar 1986 an das IEEE zur Normierung übergeben. Im Dezember 1987 wurde mit "VHDL" (IEEE-Norm 1076-1987) der erste IEEE-Standard für Hardwarebeschreibungssprachen festgelegt. Dieser Standard definiert die Syntax und Semantik der Sprache, nicht jedoch die Anwendung!

Die Unterstützung von VHDL durch Softwarehersteller wuchs und mittlerweile hat die Sprache einen solch hohen Verbreitungsgrad erreicht, das sich kaum jemand der damit einhergehenden Wandlung beim Entwurf elektronischer Systeme entziehen kann.

Nach IEEE-Richtlinien muß eine Norm alle 5 Jahre überarbeitet werden. Der für 1992 vorgesehene Prozess zur Neudefinition der Norm hat sich allerdings derart verzögert, das sie die Bezeichnung IEEE 1076-1993 erhalten hat. Noch länger wird es dauern, bis Softwarehersteller die neuen Konstrukte in ihre Programme integriert haben.

1.3 Design-Methodik mit VHDL

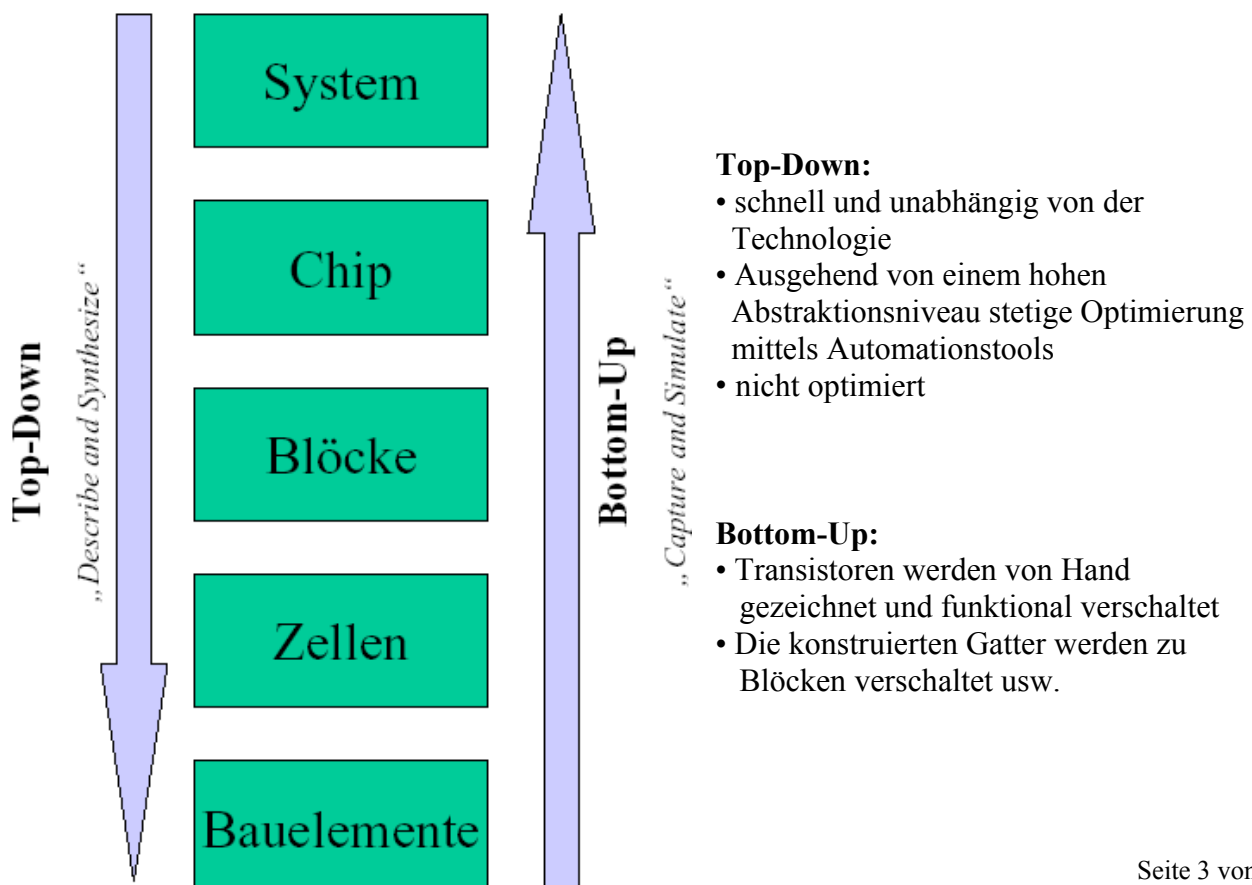
Mit VHDL wurde nicht nur ein neues Datenformat oder eine neue Sprache eingeführt, sondern zusammen mit der Durchgängigkeit der Beschreibung eines Systems von der Spezifikation zur Gatternetzliste mit Hilfe von VHDL konnten viele neue Werkzeuge entwickelt werden, die eine komplett neue Entwurfsmethodik begründet haben.

Einige wesentliche neuen Tools dienen zur:

- Spezifikationserfassung
- Graphischen Systembeschreibung mit Möglichkeiten zur frühzeitigen Simulation
- Generierung von VHDL-Code aus dieser graphischen Beschreibung
- Synthese

Nach der Erfassung der Aufgabenstellung über Ablaufdiagramme und Flusspläne erfolgte bisher der Entwurf manuell bis zur Gatternetzliste, die graphisch über sog. Schematic Entry Tools in den Rechner eingegeben wurde, selten mit einfachen Logikentwurfswerkzeugen.

Der Anfangspunkt der Rechnerunterstützung wurde beim Top-Down Design im Lauf der Zeit weit in Richtung frühe Entwurfsphasen verschoben. Mit Hilfe einer Beschreibungssprache wie VHDL kann das System bereits frühzeitig beschrieben und auch simuliert werden. Fehler und notwendige Korrekturen werden somit viel früher entdeckt.



Es existieren auch Werkzeuge, die eine graphische Eingabe des Entwurfs gestatten (z.B. in Form von Statecharts, erweiterten Automatengraphen). Diese Tools bieten zumeist auch die Möglichkeit zur Simulation. Häufig kann aus einer solchen graphischen Beschreibung dann anschließend (synthetisierbarer) VHDL-Code auf Register-Transfer-Ebene erzeugt werden.

Stehen diese Möglichkeiten der automatischen Code-Generierung nicht zur Verfügung, muss die VHDL-Beschreibung entsprechend manuell modifiziert werden, denn z.Zt. erfordern verfügbare Synthesewerkzeuge eine Systembeschreibung auf Register-Transfer-Ebene.

Bei der sich anschließenden Synthese der VHDL-RTL-Beschreibung wird eine noch technologieunabhängige Beschreibung auf Logikebene erzeugt. Die Entscheidung für das Umsetzen auf eine bestimmte Technologie ("Technology Mapping") und einen bestimmten Hersteller kann somit erst zu einem möglichst späten Zeitpunkt erfolgen. Anschließend Simulationen auf Logikebene können zeigen, ob die Randbedingungen (Chipfläche, zeitlichen Anforderungen, maximale Verlustleistung) auch erfüllt werden.

Danach sind die Testbitmuster für den Produktionstest zu erzeugen. Wenn beim Entwurf entsprechende Teststrukturen noch nicht berücksichtigt wurden, sind diese an dieser Stelle hinzuzufügen.

Auf Basis der technologiespezifischen Netzliste werden das Layout bzw. die Programmierdaten für die Produktion erzeugt. Danach können die notwendigen Informationen für eine sehr genaue Timing-Verifikation in die Logikebene zurückgereicht ("Backannotation"), und somit vom Layout herrührende Fehler detektiert werden.

