

Zusammenfassung: Hardware Modelling

Inhaltsverzeichnis

Hardware Design	1
Hardwarebeschreibungssprachen (HDLs)	1
VHDL	2
Hardware Modelling	2

An dieser Zusammenfassung kann gerne auf [Github](#) mitgewirkt werden!

Hardware Design

Wann eine Hardware-Implementierung sinnvoll ist:

- wenn Off-the-shelf-Hardware die Erfordernisse nicht erfüllt (Performance, Effizienz, Sicherheit, Größe)
- wenn lange Entwicklungszeiten und hohe Kosten toleriert werden können
- wenn der mögliche Gewinn die Kosten übersteigt

Tabelle 1: Hardware vs. Software

	Hardware	Software
Ausführung	gleichzeitig	sequentiell
Aktive Teile	alle	nur verwendete Code-Zeile
Mehr Instruktionen führen zu. . .	mehr Hardware	mehr Rechenzeit
Begrenzende Größe	Fläche	Zeit

Hardwarebeschreibungssprachen (HDLs)

HDLs auf dem Y-Diagramm:

- meistens auf der RTL- oder Logic-Ebene des Y-Diagramms (siehe VO Digital Design)
- meistens auf der Verhaltens- oder Strukturachse des Y-Diagramms

- in seltenen Fällen auch auf algorithmischer Ebene
- tieferliegende Ebenen werden durch Tools realisiert

Aufgaben von HDLs:

- höhere Abstraktion → geringere Produktivitätslücke
- Dokumentation (ursprünglicher Zweck von VHDL)
- Kommunikation

Produktivitäts-/Verifikations-Lücke:

1. Durch bessere Technologie könnten immer mehr Gatter auf einen Chip passen
2. Es können aber nur begrenzt komplexe Schaltungen entworfen werden
3. und nur noch weniger komplexe Schaltungen verifiziert werden

Der Unterschied zwischen (1) und (2) ist die Produktivitätslücke, der Unterschied zwischen (1) und (3) die Verifikationslücke.

VHDL

Hardware Modelling