

## Projekt 1: VHDL

Lukas Märdian

[maerdian@in.tum.de](mailto:maerdian@in.tum.de)

- Alternativen
- Algorithmus
  - Beschreibung
  - Beispiel
- Implementierung
  - Vorgehensweise
  - Probleme
  - Ausblicke
- Fragen

## Alternative1: Double-Dabble

Shift-Befehle, Addition → Effizienz

## Alternative2: Bin-Dec-BCD

Addition, Multiplikation → Anschaulichkeit

## Alternative1: Double-Dabble

Shift-Befehle, Addition → Effizienz

## Alternative2: Bin-Dec-BCD

Addition, Multiplikation → Anschaulichkeit

## Entscheidung: Double-Dabble

## Double-Dabble:

Algorithmus zur Bin  $\rightarrow$  BCD Konversion

verdoppeln & pfuschen

## Ausgangspunkt:

Ein Vektor, der sich aus einer bestimmten Anzahl an 4 Bit BCD-Ziffern und dem binären Eingabevektor zusammensetzt.

z.B.:

10er	1er	Eingabe(42)
0000	0000	00101010

verdoppeln:

Linksshift des Vektors verdoppelt den Wert.  
Shifte so oft, wie es Bits in der Eingabe gibt.

verdoppeln:

Linksshift des Vektors verdoppelt den Wert.  
Shifte so oft, wie es Bits in der Eingabe gibt.

pfuschen:

Addiere 3 auf eine BCD-Ziffer, falls diese  $\geq 5$  ist.

$$5 * 2 = 10 \leftrightarrow (5 + 3) * 2 = 16$$



10er	1er	Eingabe(42)
0000	0000	00101010

	10er	1er	Eingabe(42)
	0000	0000	00101010
<	0000	0000	01010100

	10er	1er	Eingabe(42)
	0000	0000	00101010
<	0000	0000	01010100
<	0000	0000	10101000

	10er	1er	Eingabe(42)
	0000	0000	00101010
<	0000	0000	01010100
<	0000	0000	10101000
<	0000	0001	01010000

	10er	1er	Eingabe(42)
	0000	0000	00101010
<	0000	0000	01010100
<	0000	0000	10101000
<	0000	0001	01010000
<	0000	0010	10100000

	10er	1er	Eingabe(42)
	0000	0000	00101010
<	0000	0000	01010100
<	0000	0000	10101000
<	0000	0001	01010000
<	0000	0010	10100000
<	0000	0101	01000000

	10er	1er	Eingabe(42)
	0000	0000	00101010
<	0000	0000	01010100
<	0000	0000	10101000
<	0000	0001	01010000
<	0000	0010	10100000
<	0000	0101	01000000
+	0000	1000	01000000
<	0001	0000	10000000

	10er	1er	Eingabe(42)
	0000	0000	00101010
<	0000	0000	01010100
<	0000	0000	10101000
<	0000	0001	01010000
<	0000	0010	10100000
<	0000	0101	01000000
+	0000	1000	01000000
<	0001	0000	10000000
<	0010	0001	00000000



	10er	1er	Eingabe(42)
	0000	0000	00101010
<	0000	0000	01010100
<	0000	0000	10101000
<	0000	0001	01010000
<	0000	0010	10100000
<	0000	0101	01000000
+	0000	1000	01000000
<	0001	0000	10000000
<	0010	0001	00000000
<	0100	0010	00000000

## VHDL:

Very High Speed Integrated Circuit

Hardware Description Language

## Vorgehensweise:

- interne Repräsentanten der BCD-Ziffern
- interner Zähler
- shifte bei rising\_edge(clk)
- addiere bei falling\_edge(clk)
- weise Overflow-Bit und BCD-Ziffern zu

## Problem:

Mehrere Signalzuweisungen in VHDL auf einem großen Vektor.

## Problem:

Mehrere Signalzuweisungen in VHDL auf einem großen Vektor.

## Lösung:

Großen Vektor aus Teilvektoren zusammensetzen.

## Problem:

Mehrere Signalzuweisungen in VHDL auf einem großen Vektor.

## Lösung:

Großen Vektor aus Teilvektoren zusammensetzen.

## Ausblick:

Testen der VHDL Schaltung auf einem FPGA.

# Fragen?

Vielen Dank für Ihre Aufmerksamkeit!