# Was ist VHDL? 07.11.2023-10.11.2023

VHDL steht für **V**HSIC (Very High Speed Integrated Circuits) **H**ardware **D**escription **L**anguage.

Es ist vor allem in Europa verbreitet

# Wofür wird es eingesetzt? 10.11.2023

Es wird für die Entwicklung von Analogen & Digitalen Schaltungen verwendet.

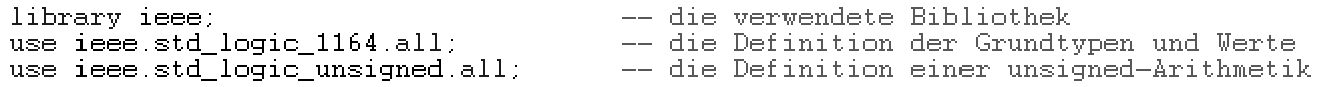
# Aufbau 10.11.2023

## Header (Bibliothek- und Packageeinbindungen)

Legt Definitionen fest.

Gilt nur für die unmittelbar folgende Schaltung.

Wird durch die Schnittstelle definiert.



Der Header ist vergleichbar mit dem Befehl #include aus C.

## Entity (Schnittstellendefinition)

Definiert die Schaltungsschnittstelle.

Hier wird der Name vom Block definiert, welcher die Schaltung und angehängte Signale darstellt.

Dieser Block ist mit der Deklaration einer Funktion in „C“ vergleichbar.

Der Funktionsname, die Typen und die Zahl der übergebenen Parameter werden definiert.

Es gibt an einem Block Eingangs-, Ausgangs- und Bidirektionale Signale.

Ein Bild, das Text, Screenshot, Schrift, Reihe enthält.

Automatisch generierte Beschreibung

Formale Syntax für die Deklaration einer Schnittstelle (entity):

Ein Bild, das Text, Schrift, Reihe, Screenshot enthält.

Automatisch generierte Beschreibung

Der Blockname und der Signalname müssen genau gleich dem Namen im Schaltplan sein, auch bei Groß- und Kleinschreibung. Dies muss man machen, da bei anderen Umgebungen z.B.: Schaltplan zwischen Groß- und Kleinschreibung unterschieden wird, und dadurch Fehler entstehen können.

Signale vom gleichen Typen können sinnvoll in Gruppen deklariert werden.

Hier dürfen die Signale keine Vektoren sein, da runde Klammern im Namen fehlen. Der Block „Zaehler“ erlaubt es, aufgrund des gewählten Signalnamen.

Ein Bild, das Text, Quittung, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

In der Entity deklarierte Signale können innerhalb als auch außerhalb des Blockes verbunden werden. Innerhalb der Schaltungsbeschreibung und Außerhalb mit anderen Blöcken.

## Architecture (Funktionsbeschreibung)