

Aufgabe 1

Beantworten Sie folgende Verständnisfragen:

- 1.1 Mit welchen Zeichen werden Kommentare versehen? Wo dürfen die Kommentare im VHDL-Code erscheinen?
- 1.2 Welche selbstdefinierten Bezeichner sind in VHDL erlaubt? Wenn nicht, erläutern Sie weshalb!
 - INTGR9
 - Intgr1_5
 - Intgr1-5
 - Acquire_
 - _On_State
 - Zero_To_3
 - Abc@deF
 - \1 2bcx@- +Name_\
- 1.3 Aus welchen Grundbestandteilen, sogenannten Design-Units, besteht eine elektrische Schaltung, die in VHDL kodiert ist? Erläutern Sie die Aufgaben der Teile?
- 1.4 Erklären Sie den Unterschied zwischen den Datentypen `bit` und `std_logic`?
- 1.5 Erläutern Sie die Schlüsselwörter `signal` und `variable`. Welcher signifikante Unterschied existiert?
- 1.6 Wozu dient das Schlüsselwort `generic`?
- 1.7 Beschreiben Sie die Bedeutung des Konzeptes `configuration`. Geben Sie ein Beispiel (kein VHDL-Code).
- 1.8 Welche Typen von `port` existieren? Können bei allen Typen Werte zugewiesen werden?
- 1.9 Was passiert nach der Abarbeitung des letzten Befehls innerhalb der `process`-Umgebung? Wie kann ein `process` gestoppt werden?
- 1.10 Wo können sequentielle Anweisungen innerhalb einer Schaltungsbeschreibung auftauchen?
- 1.11 Nennen Sie die wichtigsten Abfrage- und Schleifenkonstrukte von VHDL?
- 1.12 Wozu dienen folgende zwei Zeilen VHDL-Code, die typischerweise am Beginn einer VHDL-Beschreibung stehen?

```
library ieee;  
use ieee.std_logic_1164.all;
```
- 1.13 In welcher Weise werden Befehle im Objekt `architecture` abgearbeitet?

Aufgabe 2

Analysieren Sie jede Zeile Code auf ihre Bedeutung hin. Das Beispiel zeigt einen einfachen Zähler. Die Zeilennummerierung dient hier nur zur besseren Orientierung und ist bei den VHDL-Programmierung nicht notwendig !

```
1:  entity COUNTER_NTY is
2:      generic(  MODULUS_G   : Integer := 4;
3:                COUNTDLY_G : Time    := 5 ns;
4:                PERIOD_G_G  : Time    := 100 ns);
5:  port(        COUNT       : out Integer := 0);
6:  end  COUNTER_NTY;
7:
8:  architecture COUNTER_BEH of COUNTER_NTY is
9:      begin
10:         COUNTER_LBL: process
11:             variable COUNT_V : Integer :=0;
12:             begin
13:                 wait for PERIOD_G;
14:                 COUNT_V := (COUNT_V + 1) mod MODULUS_G;
15:                 COUNT   <= COUNT_V after COUNTDLY_G;
16:             end process COUNTER_LBL;
17:         end COUNTER_BEH;
```