

Ausbildungsbetrieb



BITLC Business IT Learning Center GmbH
Im Defdahl 10 G
44141 Dortmund

Praktikumsbetrieb



QUENDRO GmbH
Gröpperstr. 5
58454 Witten

Prüfling

Burcu Arik
Ötztalerstraße 10
44339 Dortmund

Prüflingsnummer:
1138074



FPGA-Konsolenanwendung
FPGA-AUTOMATISCHE
GENERIERUNG VON TEST
BENCH VORLAGEN



Zu meiner Person



Burcu Arik, 36



Aus der Türkei/Wohnsitz Dortmund



Seit 2018 in Deutschland



Verheiratet



Engischlehrerin

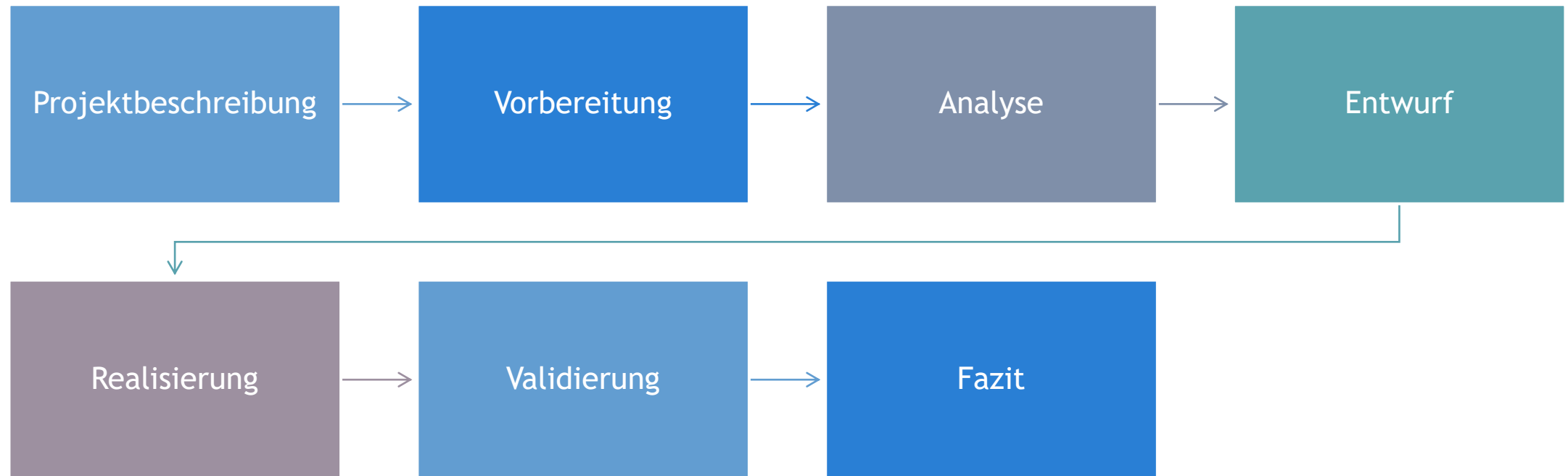


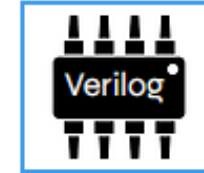
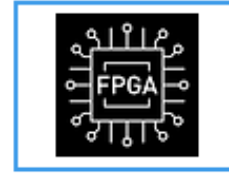
Umschulung zum Fachinformatiker als
Anwendungsentwicklung in BITLC GmbH



Quendro GmbH

AGENDA





Eckdaten



Gegründet 2022



Sitz in Witten



Start-Up/1 Mitarbeiter/Geschäftsführer



Software(C, C++), FPGA(Field Programmable Gate Array) und Electronic HW development.



Weiterbildungen im Bereich Software und Hardware Development

PROJEKT



10.01.2024



Was ist ein FPGA?

Anpassbare Chips, programmierbar für spezifische Logikfunktionen

Nach der Herstellung konfiguriert werden

Ideal für Prototyping und maßgeschneiderte Hardware

Einsatz in Automobiltechnik, Luft- und Raumfahrt, Militär, Robotik und Datenzentren

IoT-Integration

Dynamische Umgebungen

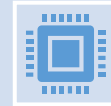
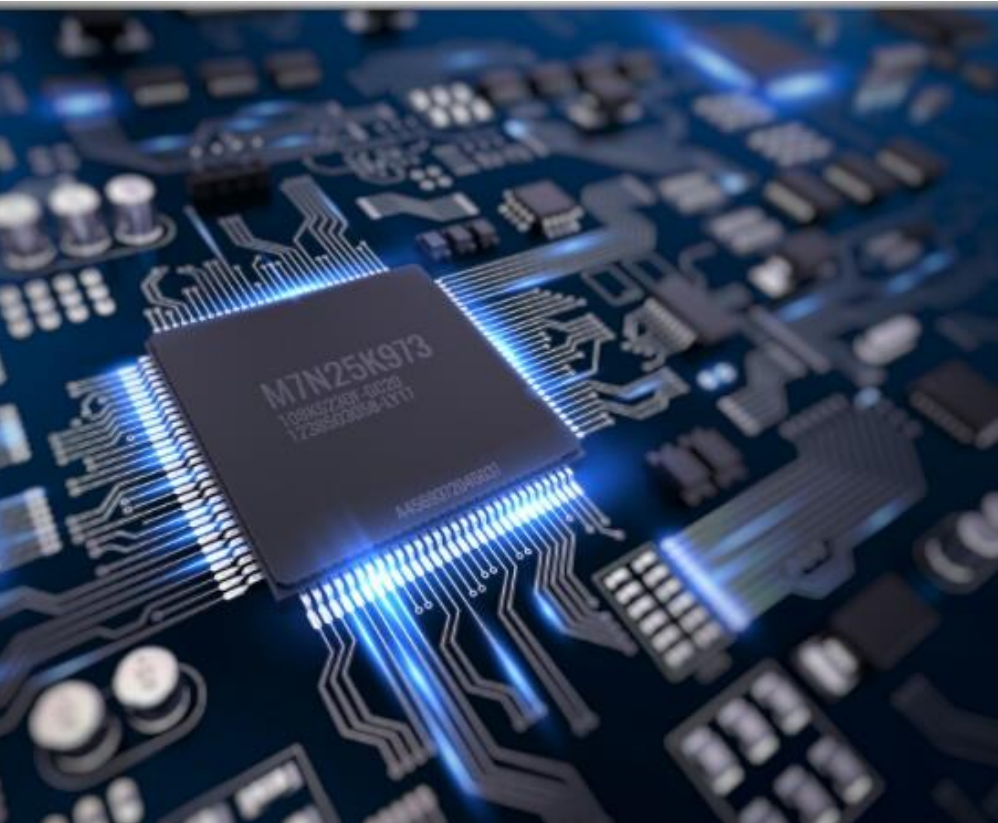


PROJEKT



10.01.2024

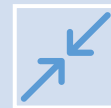
PROJEKTBSCHREIBUNG



**Automatisierung der FPGA-Test
Bench-Erstellung**



**Beschleunigung des
Testverfahrens**



Reduktion der Entwicklungszeit



**Effizienz- und
Produktivitätssteigerung**

BEGRÜNDUNG DES PROJEKTS

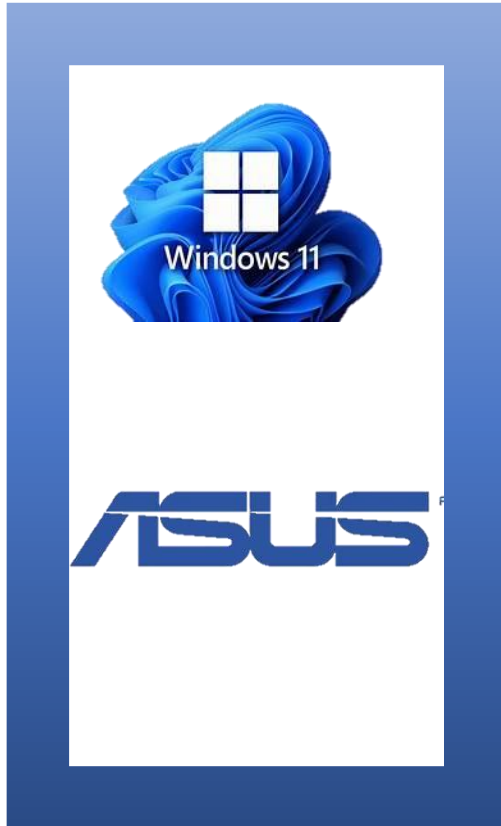
Automatisierung

Zeitersparnis

Kosteneffizienz



RESSOURCEN

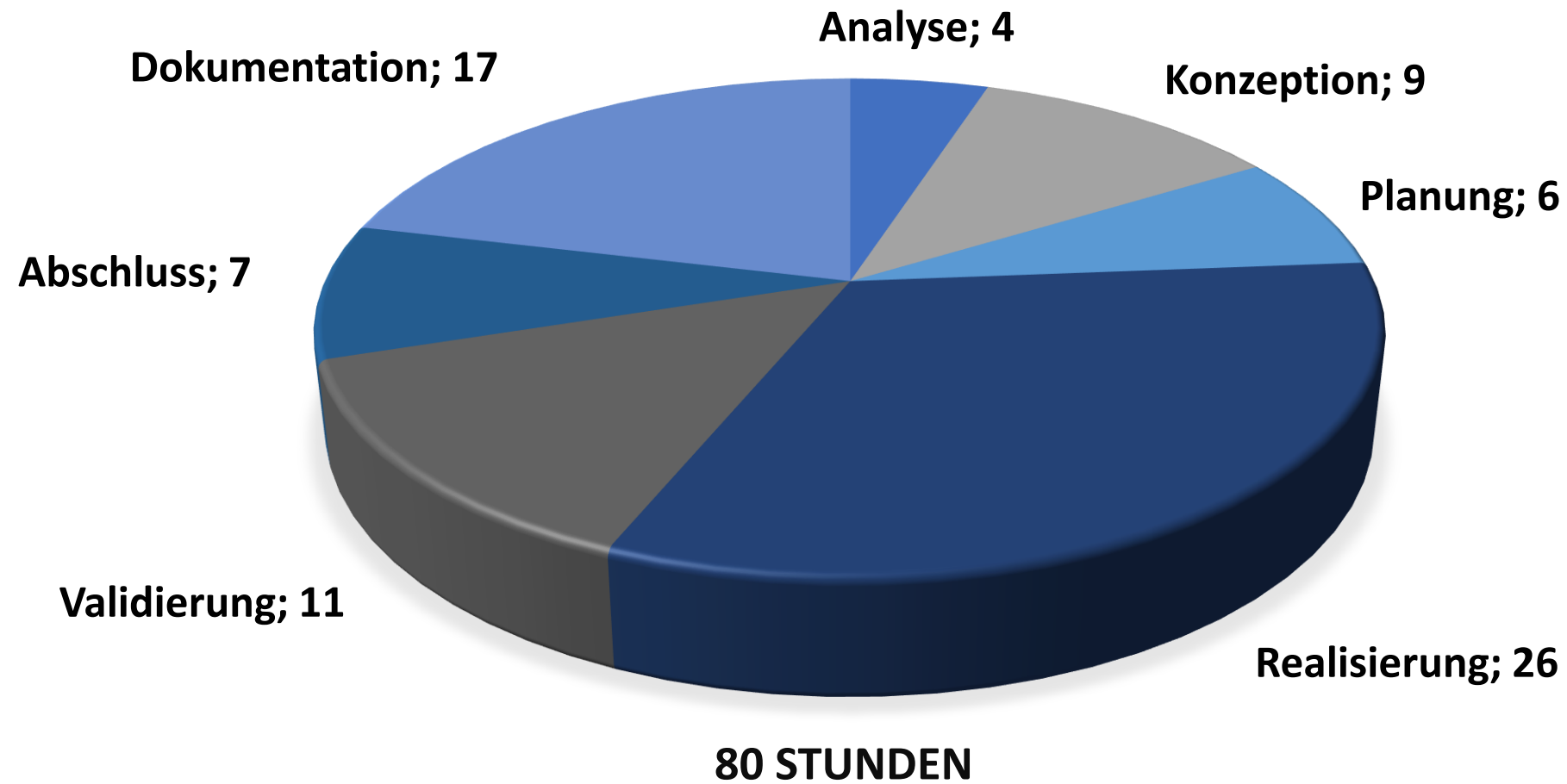


VORBEREITUNG

10.01.2024



ZEITPLANUNG

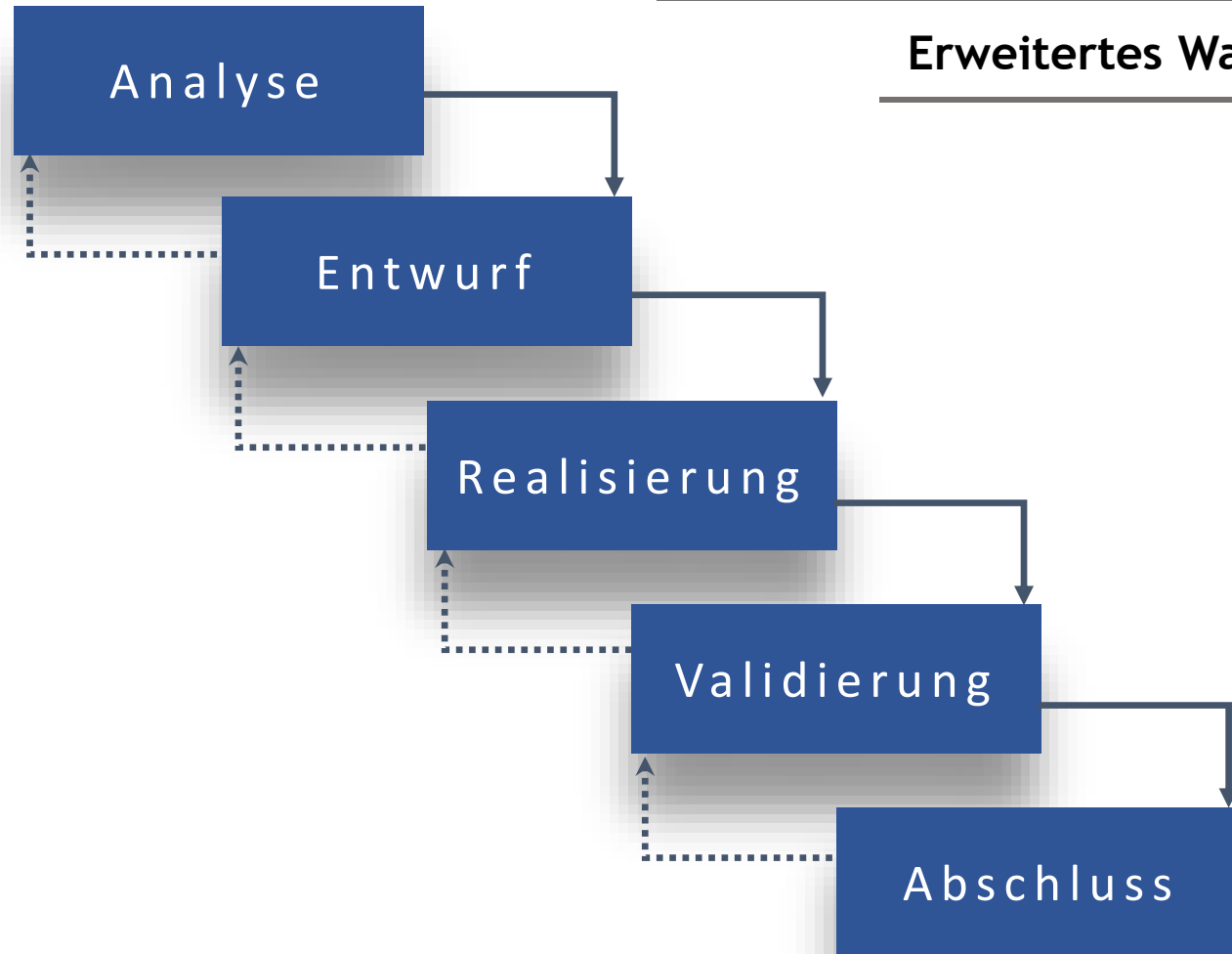


VORBEREITUNG

10.01.2024

VORGEHENSMODELL

Erweitertes Wasserfallmodell



VORBEREITUNG

10.01.2024



Zeitaufwendig bei logischen Fehlern

früher in Excel, ineffizient und fehleranfällig

Zusätzlicher Arbeitsaufwand

Automatisierte Testlösung

Fehlerreduktion

Effizienzsteigerung

Zeitersparnis

PROJEKTKOSTEN

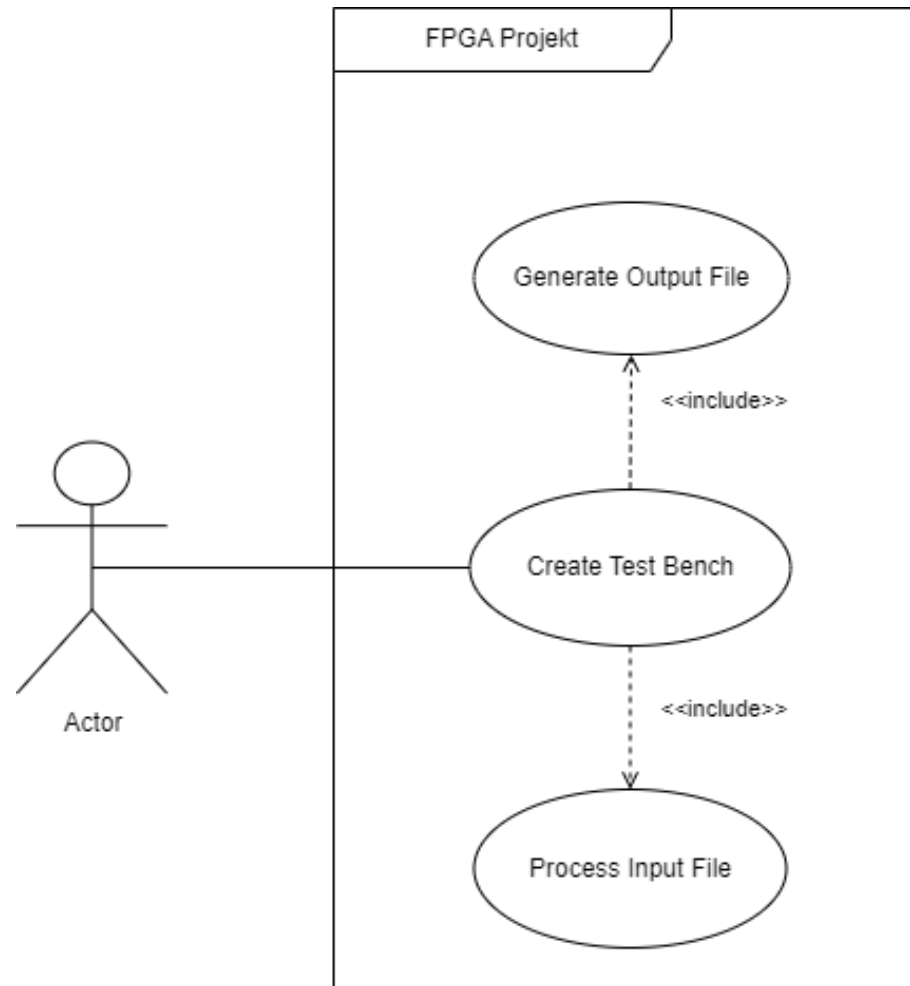
Personalkosten	Sachmittelkosten	Gesamtkosten
Entwicklungskosten, Hilfestellung, Aufsicht bei Projektplanung und Rückfragen, Abnahme, Dokumentation	Material, Energie, Büroraum, Arbeitsmaterialkosten und Software	
1320,00€	146,00€	1466,00€

ANALYSE

10.01.2024



ANWENDUNGSFALL-DIAGRAM

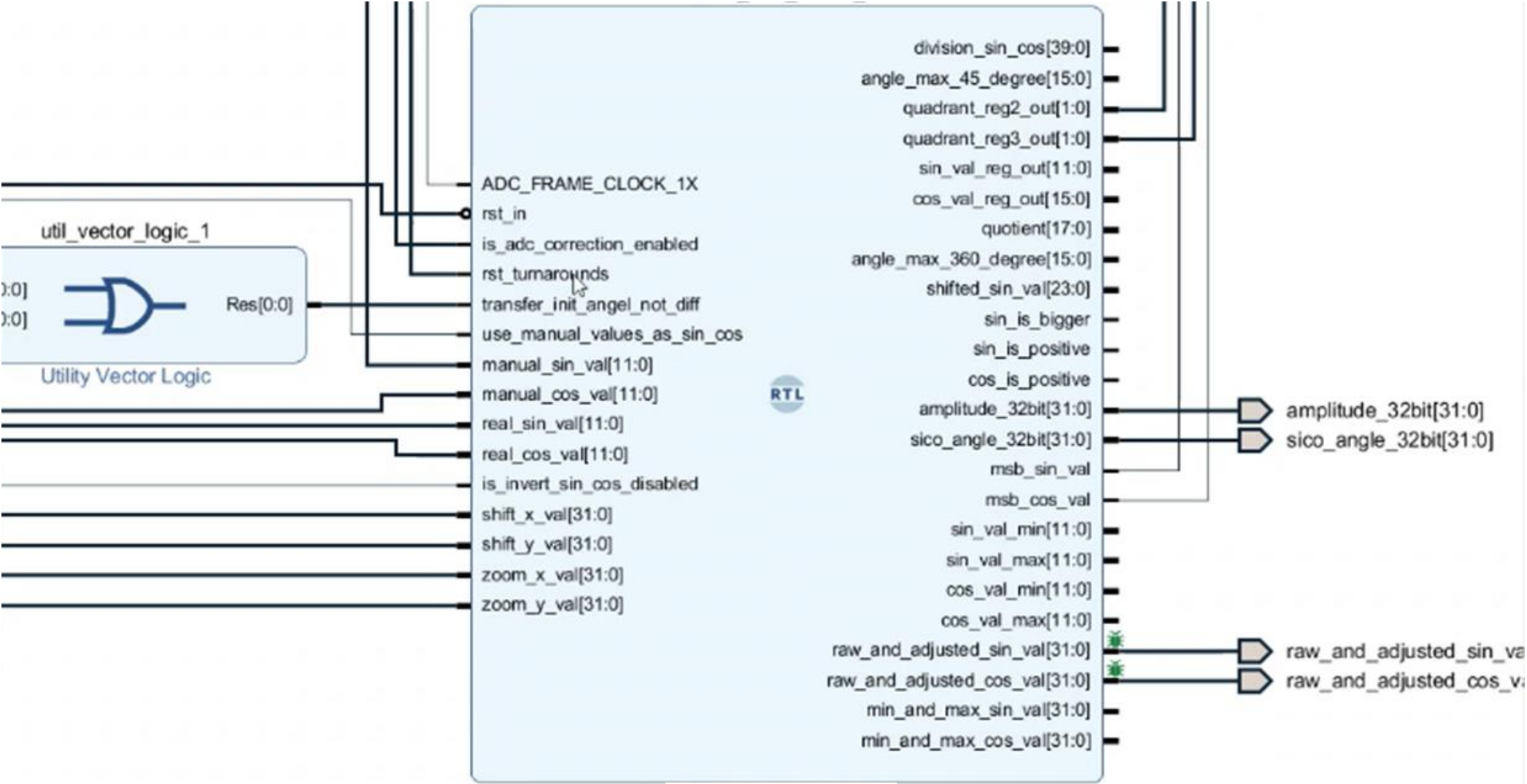


ENTWURF

10.01.2024

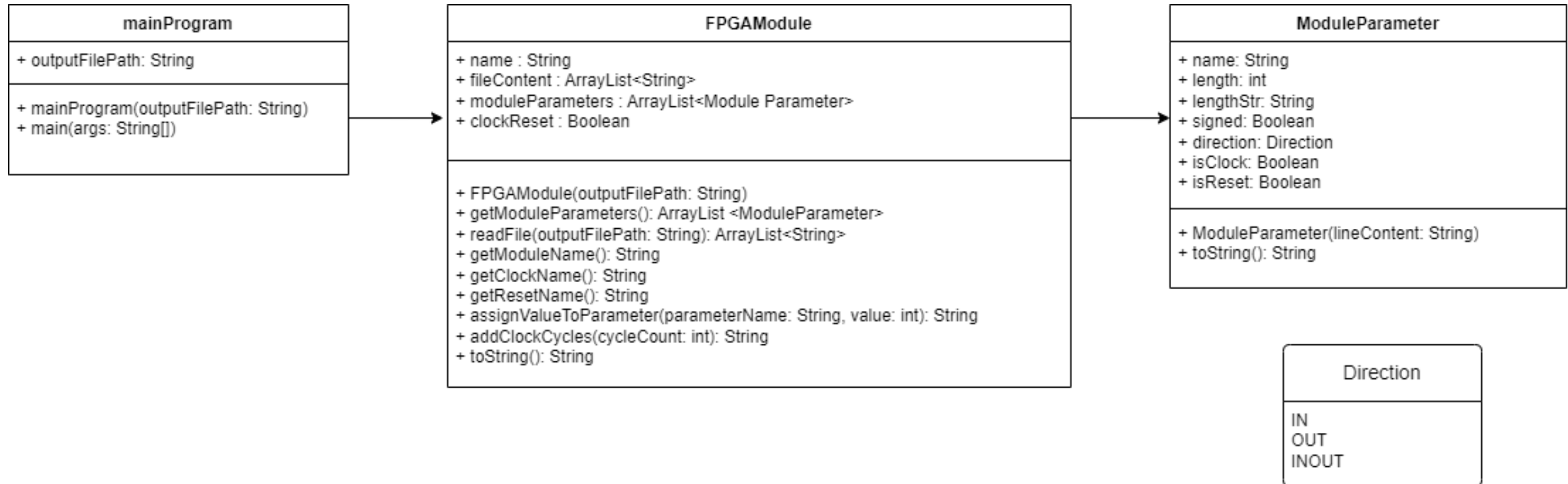


I
N
P
U
T

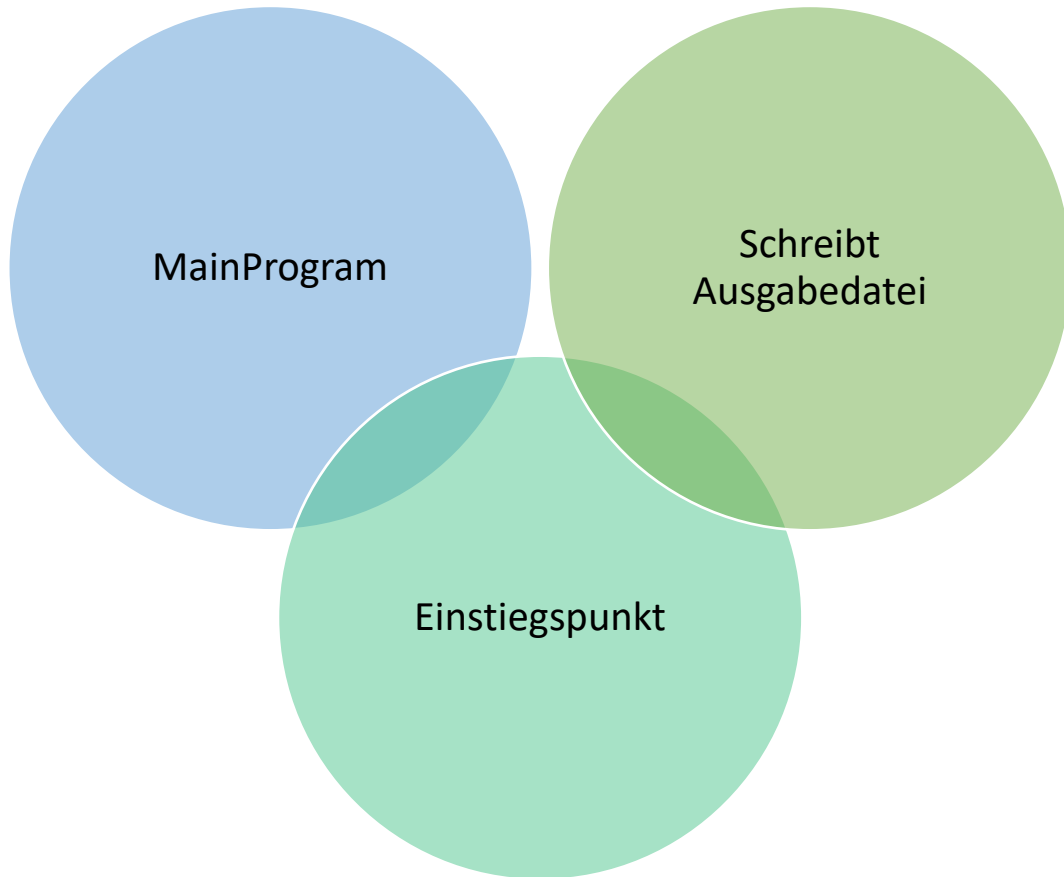


O
U
T
P
U
T

Klassendiagramm



Implementierung der Geschäftslogik - **mainProgram**



```
mainProgram.java x  FPGAModule.java  ModuleParameter.java  get_abs_pos.

1  import java.io.FileNotFoundException;
2  import java.io.PrintWriter;
3  import java.nio.file.Paths;
4  import java.util.ArrayList;
5  import java.util.Scanner;
6
7  public class mainProgram {
8      public static void main(String[] args) throws FileNotFoundException { /
9
10         if (args.length != 1) {
11             System.out.println("Usage: java mainProgram <outputFilePath>");
12             return;
13         }
14
15         String outputFilePath = args[0];
16
17
18         FPGAModule fpgaModule = new FPGAModule(outputFilePath);
19         System.out.println(fpgaModule);
20         try(PrintWriter out = new PrintWriter("./demo_test.v")){
21             out.println(fpgaModule);
22         }
23     }
24 }
```

REALISIERUNG

10.01.2024



Implementierung der Geschäftslogik - Klasse FPGAModule

mainProgram.java FPGAModule.java x ModuleParameter.java get_abs_pos_state_machi

```
10
11 public class FPGAModule {
12
13     String name;
14     ArrayList<String> fileContent;
15     ArrayList<ModuleParameter> moduleParameters;
16     Boolean clockReset;
17
18     public FPGAModule(String outputPath) {
19         fileContent = readFile(outputFilePath);
20         name = getModuleName();
21
22         moduleParameters = getModuleParameters();
23
24     }
25
26     public ArrayList<ModuleParameter> getModuleParameters() {
27         moduleParameters = new ArrayList<>();
28         boolean insideDeclarations = false;
29
30         for (String lineContent : fileContent) {
31             if (lineContent.contains("(")) {
32                 insideDeclarations = false;
33             }
34
35             if (lineContent.startsWith("input")
36                 || lineContent.startsWith("output")
37                 || lineContent.startsWith("inout")) {
38                 ModuleParameter moduleParameter = new ModuleParameter(lineContent);
39
40                 moduleParameters.add(moduleParameter);
41             }
42         }
43     }
44 }
```

```
public FPGAModule(String outputPath) {
    fileContent = readFile(outputFilePath);
    name = getModuleName();
}
```

Liest den Dateiinhalt

```
public ArrayList<ModuleParameter> getModuleParameters() {
    moduleParameters = new ArrayList<>();
    boolean insideDeclarations = false;
}
```

Modulparameter zu extrahieren

REALISIERUNG

10.01.2024



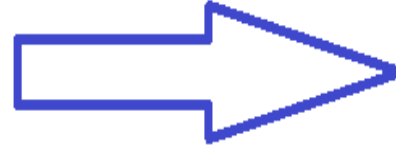
INPUT

```
module get_abs_pos_state_machine(
    input wire clk,
    input wire rst,
    input wire init_state_machine,
    input wire hls_done,
    input wire hls_ready,
    input wire [31:0] axis1_hw_counter,
    input wire [31:0] axis1_set_position_part1,
    input wire [31:0] axis1_set_position_part2,
    input wire [31:0] axis1_counts_per_m,
    input wire [31:0] axis2_hw_counter,
    input wire [31:0] axis2_set_position_part1,
    input wire [31:0] axis2_set_position_part2,
    input wire [31:0] axis2_counts_per_m,

    input wire [63:0] selected_axis_hls_calculated_abs_pos,

    output reg start_hls_calculations, //start absolu
    output reg [2:0] state,
    output reg [31:0] selected_axis_hw_counter,
    output reg [31:0] selected_axis_set_position_part1,
    output reg [31:0] selected_axis_set_position_part2,
    output reg [31:0] selected_axis_counts_per_m,

    output reg [63:0] axis1_hls_calculated_abs_pos,
    output reg [63:0] axis2_hls_calculated_abs_pos
);
```



OUTPUT

```
module tb_get_abs_pos_state_machine;
    reg clk;
    reg rst;
    reg init_state_machine;
    reg hls_done;
    reg hls_ready;
    reg [31:0] axis1_hw_counter;
    reg [31:0] axis1_set_position_part1;
    reg [31:0] axis1_set_position_part2;
    reg [31:0] axis1_counts_per_m;
    reg [31:0] axis2_hw_counter;
    reg [31:0] axis2_set_position_part1;
    reg [31:0] axis2_set_position_part2;
    reg [31:0] axis2_counts_per_m;
    reg [63:0] selected_axis_hls_calculated_abs_pos;
    wire start_hls_calculations;
    wire [2:0] state;
    wire [31:0] selected_axis_hw_counter;
    wire [31:0] selected_axis_set_position_part1;
    wire [31:0] selected_axis_set_position_part2;
    wire [31:0] selected_axis_counts_per_m;
    wire [63:0] axis1_hls_calculated_abs_pos;
    wire [63:0] axis2_hls_calculated_abs_pos;
    get_abs_pos_state_machine ttb_get_abs_pos_state_machine
endmodule
```

Implementierung der Geschäftslogik – Klasse ModuleParameter

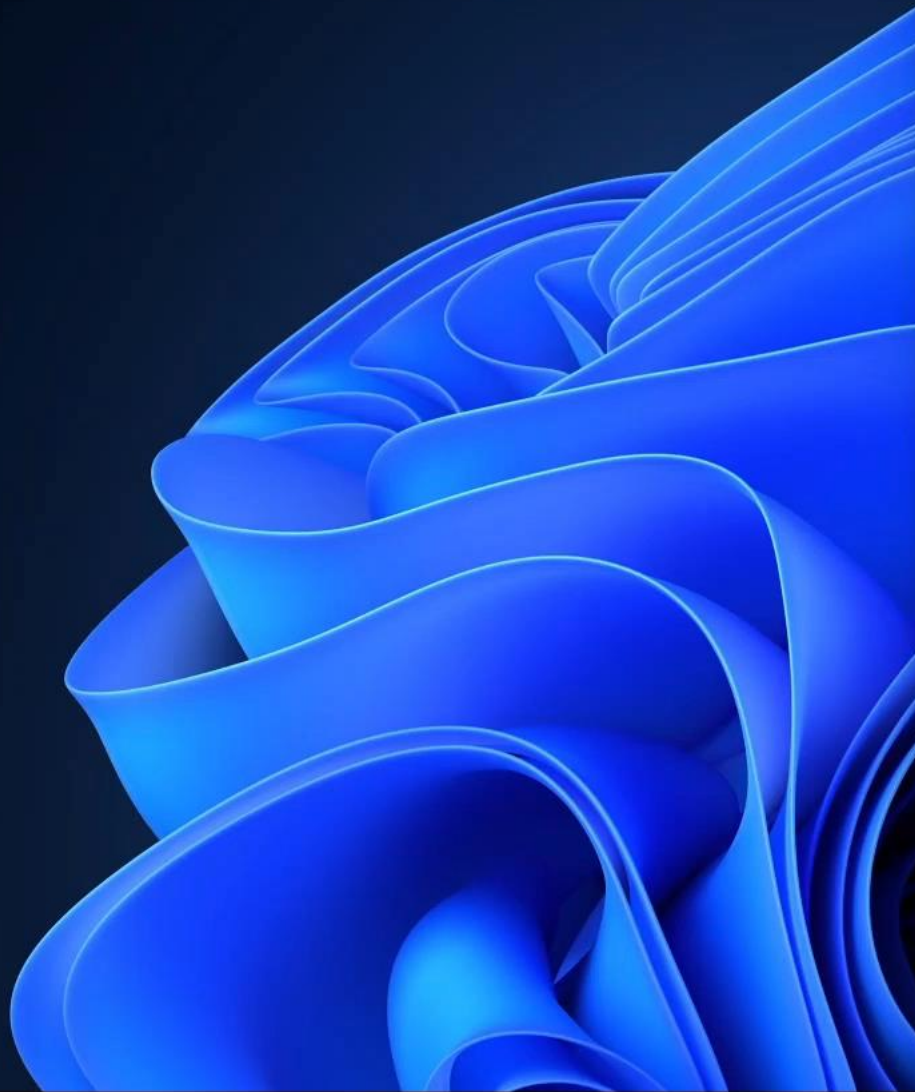
mainProgram.java FPGAModule.java **ModuleParameter.java** get_abs_pos_state_mach

```
5
6 public class ModuleParameter {
7
8     enum Direction {
9         IN,
10        OUT,
11        INOUT
12    };
13
14    String name;
15    int length;
16    String lengthStr = "";
17    boolean signed = false;
18    Direction direction; //it can be one of in, out, inout
19
20    boolean isClock;
21    boolean isReset;
22
23    public ModuleParameter(String lineContent) {
24        //remove comment part of the line
25        if (lineContent.contains("//")) {
26            int beginIndex = lineContent.indexOf("//");
27            lineContent = lineContent.substring(0, beginIndex);
28        }
29
30        //if we split line content, then last part is name of parameter
31        String[] splittedLineContent = lineContent.split(" ");
32        name = splittedLineContent[splittedLineContent.length - 1].replace(",", "");
33
34        if (lineContent.contains("input")) {
35            direction = Direction.IN;
36        } else if (lineContent.contains("output")) {
37            direction = Direction.OUT;
38        } else if (lineContent.contains("inout")) {
39            direction = Direction.INOUT;
40        }
41    }
42}
```

```
enum Direction {
    IN,
    OUT,
    INOUT
};
```

REALISIERUNG

10.01.2024



A7 Benutzer Dokumentation

Einleitung

Diese Dokumentation beschreibt, wie Sie das Test Bench Generator-Tool verwenden können, um Java-Dateien zu kompilieren und das Test Bench Vorlage auszuführen.

Schritt 1: Den Dateipfad öffnen

1. Starten Sie die Eingabeaufforderung auf Ihrem Computer.
2. Navigieren Sie zum Speicherort Ihrer Java-Projekte mithilfe des Befehls „cd“. Zum Beispiel:

```
"cd C:\JavaProjects\testbench2\TestBenchGenerator\src\main\java"
```

```
Komut İstemi
Microsoft Windows [Version 10.0.22621.2283]
(c) Microsoft Corporation. Tüm hakları saklıdır.

C:\Users\Fatih>cd C:\JavaProjects\testbench2\TestBenchGenerator\src\main\java

C:\JavaProjects\testbench2\TestBenchGenerator\src\main\java>
```

Abbildung 23 Datei Öffnen

TESTPHASE

```
module get_abs_pos_state_machine(  
    input wire clk,  
    input wire rst,  
    input wire init_state_machine,  
    input wire hls_done,  
    input wire hls_ready,  
    input wire [31:0] axis1_hw_counter,  
    input wire [31:0] axis1_set_position_part1,  
    input wire [31:0] axis1_set_position_part2,  
    input wire [31:0] axis1_counts_per_m,  
    input wire [31:0] axis2_hw_counter,  
    input wire [31:0] axis2_set_position_part1,  
    input wire [31:0] axis2_set_position_part2,  
    input wire [31:0] axis2_counts_per_m,  
  
    input wire [63:0] selected_axis_hls_calculated_abs_pos,  
  
    output reg                start_hls_calculations, //start absolute pos. calcul.  
    output reg [2:0] state,  
    output reg [31:0] selected_axis_hw_counter,  
    output reg [31:0] selected_axis_set_position_part1,  
    output reg [31:0] selected_axis_set_position_part2,  
    output reg [31:0] selected_axis_counts_per_m,  
  
    output reg [63:0] axis1_hls_calculated_abs_pos,  
    output reg [63:0] axis2_hls_calculated_abs_pos  
);  
  
    reg [2:0] prev_state;
```

```
module tb_get_abs_pos_state_machine;  
    reg clk;  
    reg rst;  
    reg init_state_machine;  
    reg hls_done;  
    reg hls_ready;  
    reg [31:0] axis1_hw_counter;  
    reg [31:0] axis1_set_position_part1;  
    reg [31:0] axis1_set_position_part2;  
    reg [31:0] axis1_counts_per_m;  
    reg [31:0] axis2_hw_counter;  
    reg [31:0] axis2_set_position_part1;  
    reg [31:0] axis2_set_position_part2;  
    reg [31:0] axis2_counts_per_m;  
  
    reg [63:0] selected_axis_hls_calculated_abs_pos;  
    wire start_hls_calculations;  
  
    wire [2:0] state;  
    wire [31:0] selected_axis_hw_counter;  
    wire [31:0] selected_axis_set_position_part1;  
    wire [31:0] selected_axis_set_position_part2;  
    wire [31:0] selected_axis_counts_per_m;  
  
    wire [63:0] axis1_hls_calculated_abs_pos;  
    wire [63:0] axis2_hls_calculated_abs_pos;  
    get_abs_pos_state_machine ttb_get_abs_pos_state_machine  
    (  
        .clk(clk),
```

VALIDIERUNG

10.01.2024



SOLL/- IST-VERGLEICH

Projektphase	Geplant	Tatsächlich	Differenz
Analyse	4h	4h	0h
Konzeption	9h	9h	0h
Planung	6h	6h	0h
Realisierung	26h	26h	0h
Validierung	11h	10h	-1h
Abschluss	7h	7h	0h
Dokumentation	17h	18h	+1h
Gesamtsumme	80h	80h	0h

FAZIT

10.01.2024



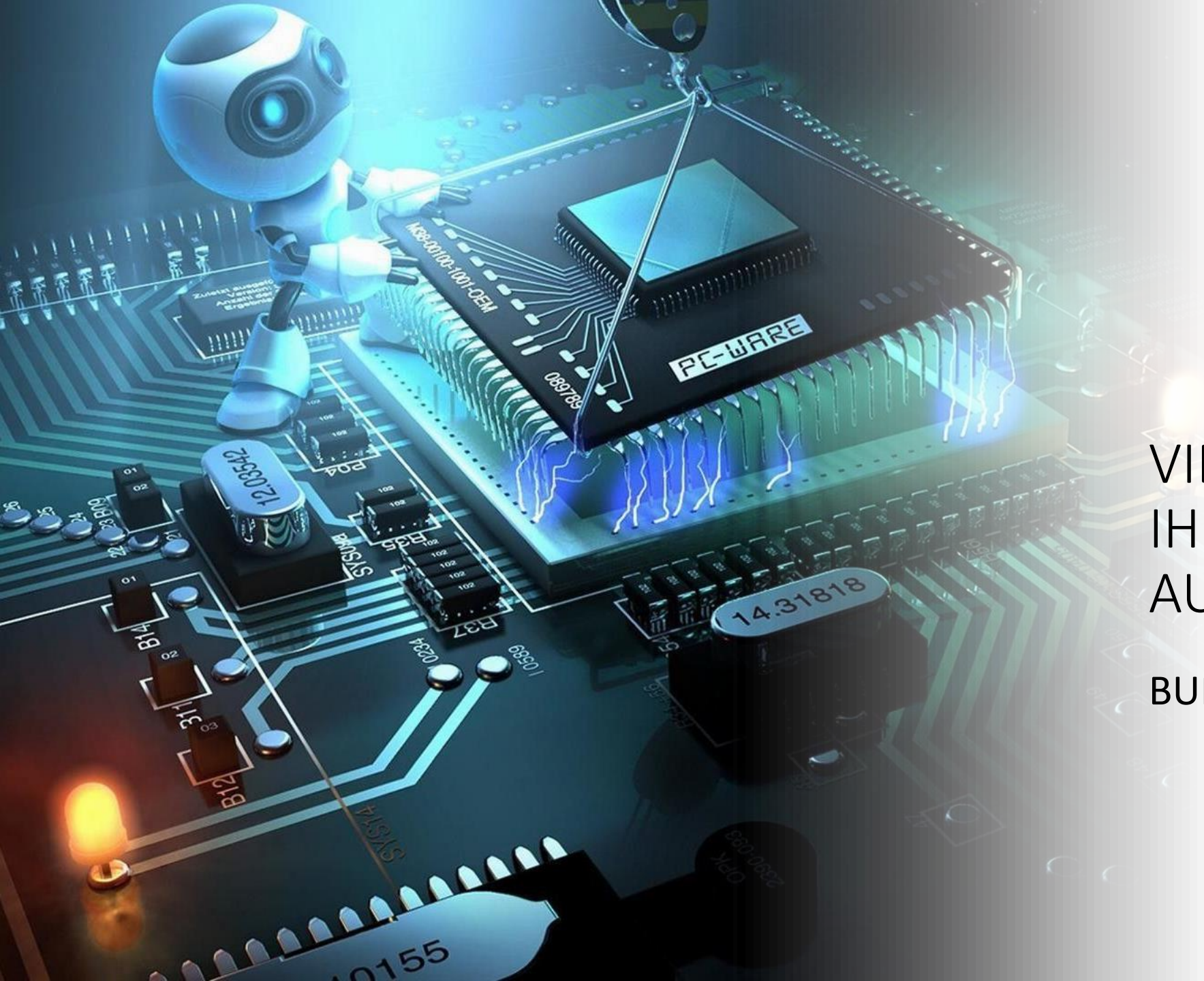
FAZIT

- Projektphasen waren erfolgreich
- Lauffähiges Programm
- Vorgaben erfüllt
- Programm genutzt werden

ZUKUNFT

- Erfahrungen
- Lernfähigkeit





VIELEN DANK FÜR
IHRE
AUFMERKSAMKEIT

BURCU ARIK