

Digitale Schaltungssimulation im Browser

Pflichtenheft

Niels Andrä
Maximilian Engelhardt
Dario Götze
Jonas Knüpper
Paul Lindner
Dorian Müller
Markus Seeber
Alexander Zenkner

Version 2.0

Autor des Dokuments	Jonas Knüpper	Erstellt am	11.04.2017
Projektname	Digitale Schaltungssimulation im Browser		
Seitenanzahl	30	© 2017	Vertraulich!

Historie der Dokumentversionen

Version	Datum	Autor	Änderungsgrund / Bemerkungen
1.0	13.04.2017	jknu	Ersterstellung
1.5	20.04.2017	jknu	Vorlage einer unfertigen Version zur Besprechung
2.0	21.04.2017	jknu	Übergabe des fertigen Pflichtenheftes an Projektbetreuer

Inhaltsverzeichnis

Historie der Dokumentversionen	2
Inhaltsverzeichnis	3
1 Einleitung	5
1.1 Zweck und Ziel des Dokuments	5
1.2 Projektmitarbeiter	5
1.3 Meilensteine	6
2 Konzept und Rahmenbedingungen	7
2.1 Einsatzbereich	7
2.2 Zielgruppe	7
2.3 Allgemeine Ziele der Projektgruppe	7
2.4 Ressourcen	7
2.5 Zeitlicher Rahmen	7
2.6 Produktumgebung	8
2.7 Zielbestimmungen	8
2.8 Muss-Kriterien	8
2.9 Kann-Kriterien	9
2.10 Abgrenzungskriterien	9
2.11 Aufwand-Nutzen-Analyse	9
2.12 Risikoanalyse	9
3 Benutzeroberfläche	10
3.1 Navigationsbaum	10
3.2 Editor	10
3.3 Menüleiste	10
4 Funktionale Anforderungen	11
4.1 Benutzerfunktionen	11
4.1.1 Projekte	11
4.1.2 Bibliotheken	11
4.1.3 Komponenten	11
4.1.4 Baumfunktionen	12
4.1.5 Editorfunktionen	12
4.1.6 Toolboxfunktionen ohne ausgewählte Komponente	13
4.1.7 Toolboxfunktionen mit einer ausgewählten Komponente	13
4.1.8 Toolboxfunktionen mit einer oder mehreren ausgewählten Komponenten	14
4.1.9 Hilfefunktionen	14
4.2 Simulationsfunktionen	14
4.2.1 Schaltung	14
4.2.2 Komponenten	14
5 Nichtfunktionale Anforderungen	14
5.1 Im-und Export von Daten im menschenlesbaren Format	14
5.2 Benutzbarkeit der Grafischen Benutzerschnittstelle	15
5.3 Bedienungsanleitung	15
5.4 Robustheit	15
5.5 Integrationsschnittstelle	15

6	Anwendungsfälle.....	16
7	Glossar.....	20
8	Freigabe / Genehmigung.....	21
9	Anhang / Ressourcen	22
9.1	Grundkomponenten	22
9.2	Zusammengesetzte Grundkomponenten	27
10	Abbildungsverzeichnis	30

1 Einleitung

1.1 Zweck und Ziel des Dokuments

Dieses Pflichtenheft beschreibt den Ablauf, die Ziele, die Anforderungen und den Aufwand des Projektes „Digitale Schaltungssimulation im Browser“. Ziel des Projektes ist die Erweiterung des GOLDi-Labs (Grid of Online Labs Ilmenau) um einen grafischen Editor zur Darstellung und Simulation von Blockschaltbildern. Dieser soll es ermöglichen auf einer Arbeitsfläche Schaltungsbausteine zu einem Blockschaltbild zusammenzufügen und die daraus resultierende Schaltung zu simulieren. Darüber hinaus, soll es die Möglichkeit geben eigene Schaltungsbausteine zu erstellen und zu editieren. Zusätzlich soll es eine Schnittstelle geben, damit das Projekt in die bestehende Web-Umgebung integriert werden kann. Das Softwareprojekt wird im Folgenden als BEAST („Block Diagram Editing And Simulating Tool“) bezeichnet.

1.2 Projektmitarbeiter

Rolle / Rollen	Name	Bemerkungen
Projektleiter	Jonas Knüpper	Teamleitung, Kommunikation im Team, Planung, Überwachung, Programmierung, Test, Dokumentation
Projektleiter	Markus Seeber	theoretische Grundlagensammlung, technische Leitung Aufwands- und Zeitplanung, Programmierung, Test, Dokumentation
Entwickler	Maximilian Engelhardt	Programmierung, Test, Dokumentation
Entwickler	Dario Götze	Programmierung, Test, Dokumentation
Entwickler	Paul Lindner	Programmierung, Test, Dokumentation
Entwickler	Niels Andrä	Programmierung, Test, Dokumentation
Entwickler	Alexander Zenkner	Programmierung, Test, Dokumentation
Entwickler	Dorian Müller	Programmierung, Test, Dokumentation

1.3 Meilensteine

Vorbereitungsphase	
Überprüfung Lastenheft	06.04.2017
Freigabe Lastenheft	13.04.2017
Grobentwurf	
Freigabe/Präsentation Pflichtenheft	21.04.2017
Feinentwurf	
Freigabe/Präsentation Feinentwurf	03.05.2017
Implementierung	
Fertigstellung GUI	18.05.2017
Implementierungsabschluss	07.06.2017
Test	
Testabschluss	30.06.2017
Abschluss	
Abschlusspräsentation	06.07.2017
Übergabe/Bewertung	07.07.2017

2 Konzept und Rahmenbedingungen

2.1 Einsatzbereich

Das Webmodul BEAST soll für die Vorbereitung der Praktika Technische Informatik an der TU Ilmenau genutzt werden. Im Praktikum realisieren Studierende einen Schaltungsentwurf mittels von ihnen in der Praktikumsvorbereitung erstellter Blockschaltbilder.

2.2 Zielgruppe

Da BEAST von Studierenden zur Praktikumsvorbereitung für das Praktikum Technische Informatik genutzt werden soll, bilden Studierende, welche die genannten Praktika absolvieren, als Zielgruppe. Studierende, welche die Praktika nicht absolvieren, können BEAST auch zur Veranschaulichung und Simulation von logischen Schaltungen verwenden.

2.3 Allgemeine Ziele der Projektgruppe

- Anwendungspaket Digitale Schaltungssimulation im Browser, bestehend aus:
 - Schnittstellen zur Einbettung in bestehendes GOLDi-System
 - Werkzeug zur Erstellung und Simulation von Blockschaltbildern
 - Schnittstellen zur Speicherung einzelner und zusammengesetzter Bibliotheken
- Alle dazugehörigen Quellcode-Dateien
- Pflichtenheft
- Projektstrukturplan
- Lückenlose Dokumentation
- Bedienungsanleitung auf Deutsch/Englisch

2.4 Ressourcen

- Web-Server für Tests
- SVN-Server
 - Repository mit festgelegtem Namen
- Privater Entwicklungs-PC
 - Entwicklungsumgebung: Webstorm
 - Programmiersprachen: HTML5, Typescript/Javascript
 - Tortoise SVN
 - Kommunikationssoftware: Slack, TeamViewer
 - Dokumentationssoftware: Microsoft Word
 - Visualisierungssoftware: Microsoft Visio
 - Präsentationssoftware: Microsoft Powerpoint
 - Testmanagement: TestOffice
- Projektmanagementsoftware Projektron BCS
- Verwendete Bibliotheken
 - Bootstrap
 - SimcirJS

2.5 Zeitlicher Rahmen

- Beginn: 03.04.2017
- Fertigstellung: 06.07.2017

2.6 Produktumgebung

BEAST fügt sich in das bestehende System des GOLDi Labs des Fachgebietes "Integrierte Kommunikationssysteme" der Fakultät "Informatik und Automatisierung" ein. Das GOLDi Lab ist eine Plattform, bestehend aus einer Sammlung von Webapplikationen, welche die Lehre des Fachgebietes unterstützen sollen. Über dieses System haben Studierende Zugriff auf virtuelle und reale Versuchsaufbauten. Es stehen bereits Werkzeuge zur Erzeugung von Automatengraphen, sowie zur manuellen Steuerung der Modelle zur Verfügung. BEAST erweitert das System um eine Komponente zum Entwurf logischer Schaltungen im Browser. Die so entworfenen logischen Schaltungen können von Studierenden zur Vorbereitung auf die vom Fachgebiet Technische Informatik angebotene Praktika genutzt werden. Außerdem bietet BEAST die Grundlage für eine stärkere Integration von GOLDi Labs in den Praktikumsbetrieb. Folgeprojekte sollen es möglich machen, die entworfenen Schaltungen direkt zur Steuerung der Modelle innerhalb des GOLDi-Labs zu nutzen.

2.7 Zielbestimmungen

BEAST beschreibt ein Webmodul, mit welchem es möglich sein soll, Blockschaltbilder aus Komponenten zusammenzusetzen und den entstehenden Blockschaltplan live („on the fly“) zu simulieren. Komponenten sollen zusammengefasst als Bibliotheken importiert und exportiert werden können. Des Weiteren soll der Projektfortschritt lokal speicherbar sein.

2.8 Muss-Kriterien

- Benutzer
 - Anlegen eines neuen Projektes/neuer Bibliothek/neuer Komponente
 - Anlegen eines Blockschaltbildes
 - Importieren eines/r Projektes/Bibliothek
 - Auswahl von Komponenten aus Bibliotheken
 - Verknüpfung von Komponenten zu zusammengesetzten Komponenten
 - Bearbeitung von zusammengesetzten Komponenten
 - Speichern von zusammengesetzten Komponenten in einer Komponentenbibliothek
 - Exportieren von mehreren Bibliotheken und Komponenten in eine neue Bibliothek
 - Lokale Speicherung des Projekts
 - Leitungssimulation An-/Abschalten
 - Bereichsauswahl von mehreren Komponenten und Extraktion dieser in eine neue zusammengesetzte Komponente
 - Rotieren von Komponenten
 - Benennung/Umbenennung von Komponenten
 - Bedienung der Oberfläche per Shortcuts
- Simulation
 - Live-Anzeige von LEDs
 - farbliche Markierung von Ein- und Ausgängen, je nach Signal
 - Integration in Webseiten mit definierter Schnittstelle
 - Benutzerinteraktion mit dafür vorgesehenen Komponenten (z.B. Schalter, Taster, ...)
- Unterstützung
 - Browser:
 - Mozilla Firefox Version 52.0.1
 - Google Chrome Version 57.0.2987.133

2.9 Kann-Kriterien

- Benutzer
 - Undo-Funktion
 - Anlegen von mehreren Blockschaltbildern in einem Projekt
 - Komponenten ersetzen (lokal/global)
 - Kontextbasiertes Rechtsklick-Menü im Baum und im Editor
 - Frequenzgenerator Umschaltung auf manuelle Taktgebung
 - Umschaltung generell high oder low aktiv

2.10 Abgrenzungskriterien

- Keine Unterstützung mobiler Geräte und mobiler Browser
- Keine automatische Positionierung oder Re-Positionierung von Komponenten auf der Benutzeroberfläche

2.11 Aufwand-Nutzen-Analyse

Aufgrund der Voraussetzungen, dass die Auftragnehmer Studenten der TU-Ilmenau sind und das Projekt in Folge ihrer Lehrausbildung absolvieren, um Erfahrungen mit Projekten der freien Wirtschaft zu sammeln wird auf eine tabellarische und monetäre Aufwand-Nutzen-Analyse verzichtet.

Aufwände der Projektgruppe sind durch die Studien- und Prüfungsordnung der Fakultät für Informatik und Automatisierung der TU-Ilmenau festgelegt. Mittels der zu erlangenden Leistungspunktzahl von acht Credits und der zeitlichen Umrechnung eines Credits in 30 Zeitstunden, kann für die acht Projektmitarbeiter ein Gesamtaufwand von 1.920 Stunden berechnet werden. Da für das Projekt bestehende Web-Server, sowie Software-Lizenzen der TU-Ilmenau genutzt werden, fallen keine zusätzlichen Kosten an.

Gegenüber den Aufwänden steht der Nutzen, dass Studierende, welche die Praktika der Technischen Informatik an der TU-Ilmenau absolvieren, diese besser vorbereiten können. Durch die Simulation ist es den Benutzern möglich, ausgearbeitete Schaltpläne zu überprüfen, was zu einer erleichterten Durchführung der Praktika führen kann.

2.12 Risikoanalyse

Dem Projekt liegen Risiken unterschiedlicher Art zu Grunde. Auf eine Aufschlüsselung von verschiedenen Eintrittswahrscheinlichkeiten wird in diesem Dokument verzichtet. Im Folgenden werden kurz Risiken, die das Projekt betreffen können, aufgelistet.

- Unvorhersehbare Ressourceneinschränkung (z.B. Ausfall von Projektmitarbeitern)
- Erhöhter Anpassungsaufwand bei verwendeten Bibliotheken
- Erhöhter Zeitaufwand durch das Auftreten und Beheben von Fehlern

Des Weiteren ist ein Planungsrisiko in dem Projekt enthalten, da weder die Projektleiter, noch die Projektmitarbeiter Erfahrungen in der Planung und Bearbeitung von Projekten haben.

3 Benutzeroberfläche

Die Benutzeroberfläche von BEAST ist in drei Bereiche unterteilt. Hierzu gehören ein Navigationsbaum auf der linken Seite, ein Editor-Fenster inklusive Toolbox auf der rechten Seite und eine Menüleiste im oberen Bereich der Anwendung.

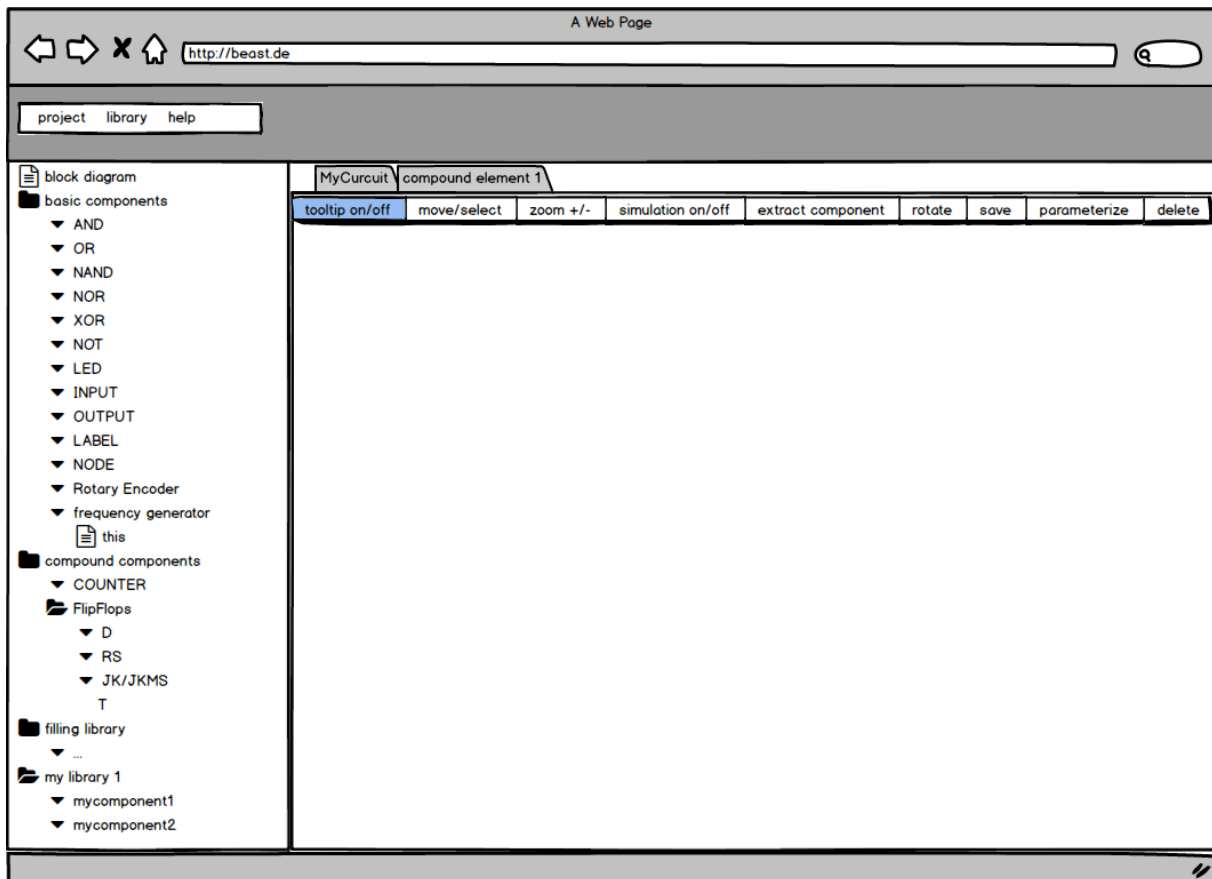


Abbildung 1: Grafische Benutzeroberfläche BEAST

3.1 Navigationsbaum

Der Navigationsbaum enthält außer dem Blockschaltbild der Schaltung auch die Bibliotheken. Zusätzlich zu den Basisbibliotheken „Grundkomponenten“, „zusammengesetzte Grundkomponenten“ und „Komponentenablage“ können weitere Bibliotheken mit spezifischen Komponenten enthalten sein.

3.2 Editor

Im Editor werden die Komponenten frei per Drag&Drop angeordnet. Dazu können Komponenten aus dem Projekt zugewiesenen und somit im Navigationsbaum dargestellten Bibliotheken auf die Arbeitsfläche gezogen werden. Der Editor unterliegt einer Tab-Struktur, welche es ermöglicht Teilkomponenten einzeln zu bearbeiten.

3.3 Menüleiste

Für Grundfunktionen wie das Importieren und Exportieren von Bibliotheken oder einem Projekt steht eine Menüleiste zur Verfügung. Diese enthält auch eine Hilfe-Funktion, zum Aufruf der Bedienungsanleitung.

4 Funktionale Anforderungen

4.1 Benutzerfunktionen

4.1.1 Projekte

DSB_BF_F111_Anlegen

Der Benutzer kann ein Blockschaltbild-Projekt anlegen. Hierzu wählt er die Option Projekt → Neu aus der Menüleiste. Ein neues Projekt beinhaltet neben dem Blockschaltplan die Basisbibliotheken, bestehend aus Grundkomponenten (siehe Anhang) und zusammengesetzten Grundkomponenten und eine leere Bibliothek für die Ablage spezifische Komponenten.

DSB_BF_F112_Öffnen

Neben dem Anlegen von Projekten, ist es dem Benutzer möglich bestehende Projekte zu öffnen. Hierzu nutzt er die Option Projekt → Öffnen und wählt eine Projektdatei von seinem lokalen System aus. Durch das Öffnen wird sowohl das in der Projektdatei enthaltene Blockschaltbild, als auch die verknüpften Bibliotheken mit den zusätzlichen Komponenten geladen.

DSB_BF_F113_Exportieren

Durch das Exportieren von Projekten können erstellte Blockschaltbilder gemeinsam mit den verknüpften Komponentenbibliotheken lokal auf dem System gespeichert werden. Hierzu wählt der Benutzer die Option Projekt → Export. Anschließend gibt er einen spezifischen Pfad auf seinem lokalen System an, in welchen das Projekt exportiert wird.

4.1.2 Bibliotheken

DSB_BF_F121_Anlegen

Neue Bibliotheken können über die Menü-Leiste erzeugt werden. Anschließend tauchen sie in der Baumansicht auf. Komponenten können per Drag&Drop zugewiesen werden.

DSB_BF_F122_Importieren

Bibliotheken können über die Menü-Leiste importiert werden. Hierzu wählt der Benutzer eine Bibliothek von seinem lokalen System aus. Nach dem Import stehen dem Benutzer die Komponenten aus der importierten Bibliothek zur Verfügung.

DSB_BF_F123_Exportieren

Einzelne oder auch mehrere Bibliotheken können zusammengefasst und exportiert werden. Hierzu wählt der Benutzer Export aus der Menüleiste. Anschließend wählt er die zu exportierenden Bibliotheken und Komponenten. Diese können anschließend in der Form einer neuen Bibliothek vom Server heruntergeladen werden. Die Basisbibliotheken können nicht exportiert werden.

DSB_BF_F124_Basisbibliotheken

Die Basisbibliotheken werden mit der Erstellung eines Projektes dem Projekt hinzugefügt. Sie können nicht gelöscht oder exportiert werden. Des Weiteren heben sie sich visuell für den Benutzer sichtbar von den Bibliotheken mit den spezifischen Komponenten ab.

4.1.3 Komponenten

DSB_BF_F131_Anlegen

Neue Komponenten können durch Bearbeiten von bereits existierenden zusammengesetzten Komponenten oder das Extrahieren von ausgewählten Komponenten erstellt werden. Eine bearbeitete Komponente kann durch Speichern in der Button-Toolbar als neue Komponente einer bestehenden oder neuen Bibliothek hinzugefügt werden. Des Weiteren ist es möglich mehrere auf

der Arbeitsfläche ausgewählte Komponenten als eine neue, zusammengesetzte Komponente zu extrahieren.

DSB_BF_F132_Bearbeiten

Bestehende Komponenten können durch Doppelklick in der Baumansicht ausgewählt und anschließend bearbeitet werden. Grundkomponenten und zusammengesetzte Grundkomponenten können nicht bearbeitet werden. Die Bearbeitungsmöglichkeiten sind genauer in den Funktionen 4.1.5 beschrieben.

DSB_BF_F133_Löschen

Komponenten können aus Bibliotheken entfernt werden. Falls die Komponente im aktuellen Blockschaltbild des Projekts verwendet wurde, muss der Benutzer die Löschung und das damit verbundene Entfernen aus dem Blockschaltbild bestätigen.

DSB_BF_F134_Grundkomponenten

Sowohl die Grundkomponenten, als auch die zusammengesetzten Grundkomponenten sind n beim Anlegen eines neuen Projektes bereits vorhanden. Eine Liste der Grundkomponenten befindet sich im Anhang.

4.1.4 Baumfunktionen

DSB_BF_F141_Baumauswahl

Dem Benutzer steht am linken Rand der Anwendung eine Baumauswahl zur Verfügung. Diese zeigt die Struktur des Projektes, bestehend aus einem Blockschaltbild und den verknüpften Bibliotheken auf. Innerhalb des Baums können Komponenten via Doppelklick im Editor geöffnet werden.

DSB_BF_F142_Verschieben

Komponenten können innerhalb des Projekts in eine andere Bibliothek verschoben werden. Hierzu wählt der Benutzer die entsprechende Bibliothek aus und verschiebt sie in die Ziel-Bibliothek per Drag&Drop.

DSB_BF_F143_Toolbox

Neben den Auswahlfunktionen ermöglicht der Baum das Hinzufügen von Komponenten zum aktuellen Blockschaltbild. Hierzu werden die Komponenten via Drag&Drop zur Arbeitsfläche hinzugefügt. Der Baum fungiert hierbei als Toolbox.

4.1.5 Editorfunktionen

DSB_BF_F151_Komponentenanordnung

Der Editor dient zur Ansicht und Bearbeitung von zusammengesetzten Komponenten. Im Editor hat der Benutzer die Möglichkeit Komponenten beliebig per Drag&Drop auf der Arbeitsfläche zu platzieren.

DSB_BF_F152_Komponentenlöschung

Komponenten können von der Arbeitsfläche entfernt werden. Hierzu gibt es in der Toolbar einen Entfernen-Button. Des Weiteren kann der Benutzer Komponenten durch Drag&Drop auf den Navigationsbaum die Komponente von der Arbeitsfläche entfernen.

DSB_BF_F153_Komponentenverbindung

Komponenten können in der Arbeitsfläche durch Leitungen verbunden werden. Hierzu zieht der Benutzer eine Verbindung vom Ausgang einer Komponente zum Eingang einer anderen Komponente. Verbindungen können ähnlich wie Komponenten auch entfernt werden.

DSB_BF_F154_SimulationInteraktion

Der Benutzer kann mit Hilfe spezieller Komponenten die Simulation beeinflussen. Die Interaktionsmöglichkeiten finden sich bei der Auflistung der Grundkomponenten.

DSB_BF_F155_Bereichsauswahl

Es ist möglich mehrere Komponenten auf der Arbeitsfläche mittels einer Bereichsauswahl zu selektieren und anschließend gemeinsam zu verschieben oder zu kopieren.

DSB_BF_F156_Tooltip

Zu selektierten Komponenten wird ein Tooltip angezeigt. Der Tooltip enthält sowohl den Namen der Komponente, als auch die gesetzten Parameter.

DSB_BF_F157_Tabstruktur

Der Editor unterliegt einer Tab-Struktur wodurch es möglich ist, mehrere Komponenten zu bearbeiten. Ein neuer Tab kann durch Extrahieren von Komponenten, oder einem Doppelklick auf eine zusammengesetzte Komponente eines bereits geöffneten Tabs erzeugt werden.

4.1.6 Toolboxfunktionen ohne ausgewählte Komponente

DSB_BF_F161_BewegenAuswählen

In der Toolbox kann zwischen Auswahl von Komponenten und Bewegen der Arbeitsfläche umgeschaltet werden.

DSB_BF_F162_Leitungssimulation

Der Benutzer kann durch An- und Abwählen der Option Leitungssimulation auswählen, ob die Leitungen (Verbindungen von Komponenten) farblich bei der Simulation markiert werden. Logisch eins belegte Leitungen werden farblich von logisch null belegten Leitungen differenziert.

DSB_BF_F163_Zoom

Durch Betätigen der Zoom-In und Zoom-Out Buttons in der Button-Toolbox kann der Benutzer die Rasterung der Arbeitsflächenansicht vergrößern und verringern. Das Zoom kann auch durch die Betätigung des Mausrades mit auf die Arbeitsfläche gesetztem Fokus, durchgeführt werden. Es erfolgt im Intervall 25% bis 200% in 25% Schritten.

DSB_BF_F164_TooltipAnzeige

Über den Button Tooltip, kann die Anzeige von Komponenten-Tooltips an- und abgeschaltet werden. Tooltips werden angezeigt, wenn Komponenten selektiert sind.

4.1.7 Toolboxfunktionen mit einer ausgewählten Komponente

DSB_BF_F171_Rotieren

Ausgewählte Komponenten können durch Betätigen des Rotieren-Buttons um 90° gedreht werden.

DSB_BF_F172_Parameter

Für eine ausgewählte Komponente können unterschiedliche Parameter gesetzt werden. Die zu setzenden Parameter werden in der Darstellung der Komponenten aufgezeigt. Zusätzlich zu den spezifischen Parametern einer Komponente, verfügen alle über einen Namens-Parameter. Der Namensparameter ist bei Grundkomponenten nicht editierbar.

DSB_BF_F173_Speichern

Eine zusammengesetzte Komponente kann durch Betätigen des Buttons Speichern in eine auszuwählende Bibliothek gesichert werden.

4.1.8 Toolboxfunktionen mit einer oder mehreren ausgewählten Komponenten

DSB_BF_F181_Extrahieren

Eine oder mehrere ausgewählte Komponenten können extrahiert und als neue Komponente abgespeichert werden. Hierzu drückt der Benutzer den entsprechenden Button und wählt im Anschluss die Zielbibliothek aus. Die extrahierten Komponenten öffnen sich in einem neuen Tab des Editors als neue Komponente.

4.1.9 Hilfefunktionen

DSB_BF_F191_Hilfe

In der Menüleiste existiert die Option „Hilfe“. Diese verlinkt die Bedienungsanleitung für BEAST.

DSB_BF_F192_Shortcuts

Die Oberfläche kann neben der Bedienung über Buttons in der Toolbar auch über Shortcuts bedient werden. Die Shortcuts sind intuitiv benutzbar und orientieren sich an Standard-Shortcuts (zum Beispiel STRG+S für Speichern).

4.2 Simulationsfunktionen

4.2.1 Schaltung

DSB_SF_F211_EinAusgänge

Während der Bearbeitung von Blockschaltbildern werden diese simuliert. Um die Simulation erkenntlich zu machen werden Ein- und Ausgänge der Komponenten unterschiedlich farblich dargestellt. Sollte der Benutzer die Funktion „Leitungssimulation“ in der Editor-Toolbar angeschaltet haben, werden außerdem die Leitungen zwischen Ein- und Ausgängen farblich identisch zu den Ein- und Ausgängen dargestellt.

4.2.2 Komponenten

DSB_SF_F221_Komponentenanzeige

Einige Komponenten verfügen über integrierte Anzeigemöglichkeiten um logisch eins und logisch null an Ein- und Ausgängen darzustellen. Hierzu leuchten Indikatoren für logisch eins und für logisch null unterschiedlich farblich auf.

DSB_SF_F222_Interaktion

Der Benutzer kann bei einigen Bauteilen die Simulation durch Benutzerinteraktionen beeinflussen. Hierbei kann er neben der Eingabe von Parametern auch direkt eine Signaländerung zum Beispiel durch Anwählen eines Schalters erzeugen.

5 Nichtfunktionale Anforderungen

5.1 Im-und Export von Daten im menschenlesbaren Format

DSB_NFA_011_JSON

Die Datenstruktur der Komponentenbibliotheken inklusive ihrer Komponenten ist menschenlesbar. Daten, welche die Komponenten beschreiben, werden im menschenlesbaren JSON-Format abgespeichert. So kann der Benutzer die Definition ohne den Gebrauch von BEAST lesen. Alle Eigenschaften der verwendeten Komponenten werden gespeichert.

DSB_NFA_012_Dateitypen

Projekte und Bibliotheken besitzen eigene Dateitypen, wodurch das Risiko umgangen wird, dass

nicht-kompatible Datenformate importiert werden können. Die Dateien mit den entsprechenden Datei-Endungen werden beim Exportieren von Projekten und Bibliotheken erzeugt.

5.2 Benutzbarkeit der Grafischen Benutzerschnittstelle

DSB_NFA_021_GUI

Die Nutzerschnittstelle ist simpel aufgebaut und intuitiv zu bedienen. Dazu ist sie performant implementiert, soweit dies technisch möglich ist.

DSB_NFA_022_Reaktionsfähigkeit

Die Oberfläche verfügt über ein sogenanntes *responsive design*. Dieses ermöglicht die dynamische Anpassung der Arbeitsfläche an die Fenstergröße.

5.3 Bedienungsanleitung

DSB_NFA_031_Bedienungsanleitung

Für BEAST wird eine Benutzer-Bedienungsanleitung sowohl in englischer, als auch in deutscher Sprache angefertigt.

5.4 Robustheit

DSB_NFA_041_KonstanteAnordnung

Das Tool speichert die Anordnung von Komponenten so, dass bei einem Neu-Laden der Seite oder bei einem Speichern/Laden-Vorgang die Komponenten ihre ursprüngliche Position wieder einnehmen.

DSB_NFA_042_Reload

Nicht exportierte Änderungen am Projekt werden nicht durch ein Neu-Laden der Webseite verworfen, sondern im Session-Speicher des Browsers zwischengespeichert. Wird der Browser jedoch geschlossen, wird der Benutzer darauf hingewiesen, dass eventuelle offene Änderungen durch das Schließen verworfen werden.

DSB_NFA_043_Funktionstests

Die in BEAST enthaltenen Funktionen werden umfassend getestet. Ein Aufschluss gebendes Testprotokoll wird geführt und dem Endprodukt beigelegt.

5.5 Integrationsschnittstelle

DSB_NFA_051_Integration

BEAST lässt sich durch eine definierte Schnittstelle in sämtliche Webseiten einbauen und bietet eine definierte Schnittstelle zum Setzen und Lesen von Ein- bzw. Ausgängen. Über die Schnittstelle kann mit BEAST interagiert werden.

6 Anwendungsfälle

Die Anwendungsfälle werden im Folgenden durch UseCase-Diagramme dargestellt.

Der Benutzer verwendet BEAST, weil er ein BEAST Projekt organisieren, eine Schaltung und deren Blockschaltbild bearbeiten, und/oder eine bestehende Schaltung simulieren möchte. Dies sind die drei Grundfunktionen von BEAST (Abbildung 1Abbildung 2).

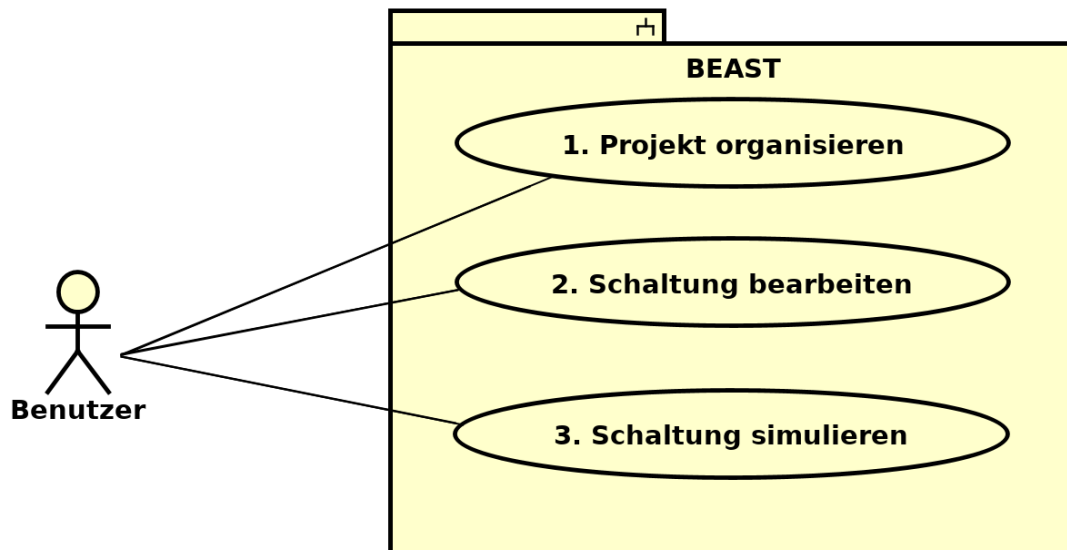
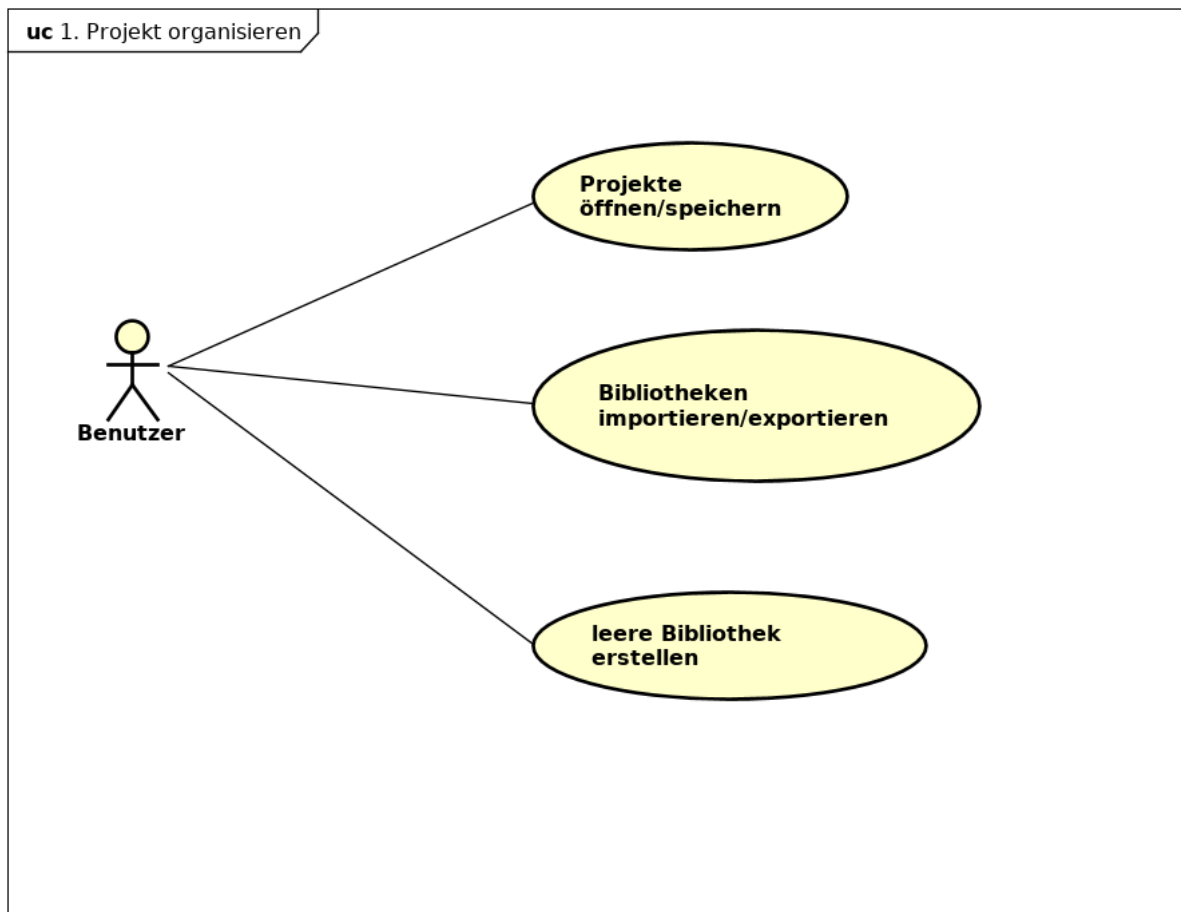


Abbildung 2: UseCases BEAST

Diese Grundfunktionen setzen sich aus UseCases der zweiten Strukturebene zusammen. Daher werden in diesen Diagrammen die den Grundfunktionen untergeordneten UseCases als Subsysteme dargestellt.

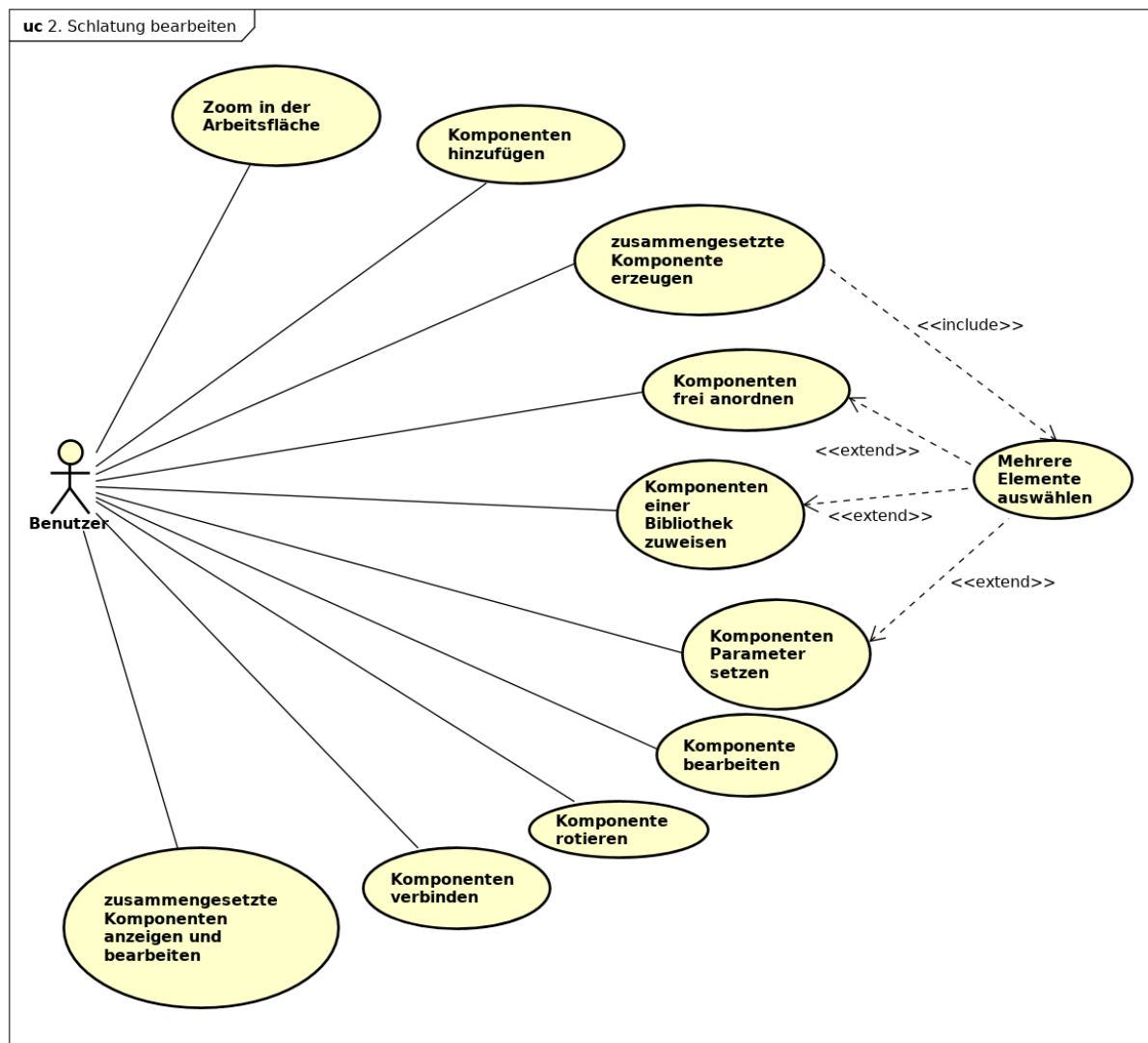
Möchte ein Benutzer ein Projekt organisieren, beinhaltet dies das Öffnen und Speichern von Projekten. Darüber hinaus ist das Anlegen, Importieren und Exportieren von Komponentenbibliotheken ein Anwendungsfall dieser Grundfunktion. Um auch Bibliotheken organisieren zu können, kann der Benutzer auch mehrere Bibliotheken gemeinsam mit einzelnen Komponenten zusammenfassen und als eine neue Bibliothek exportieren. Die UseCases für die Organisation von Projekten werden in Abbildung 3 dargestellt.



powered by Astah

Abbildung 3: UseCase Projekt organisieren

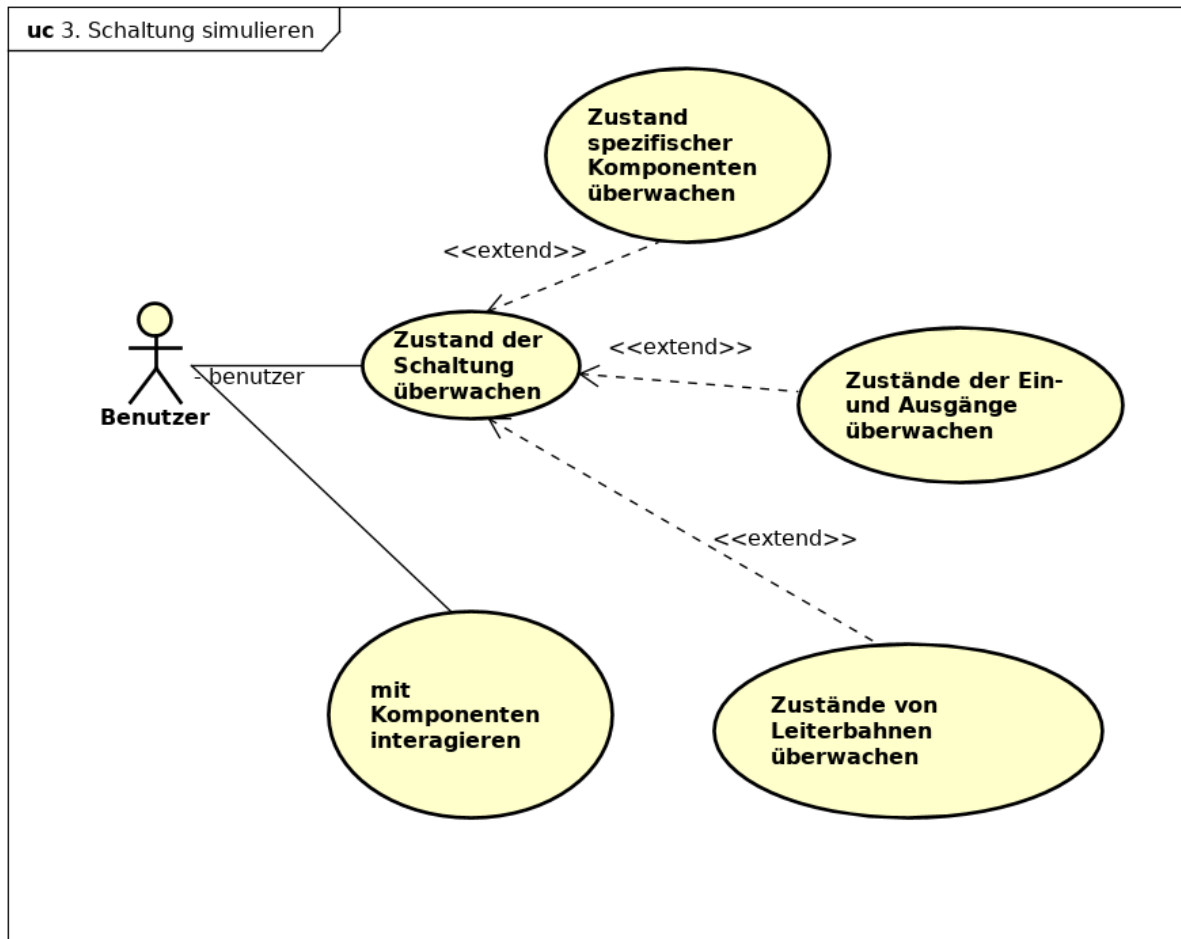
Bei der Bearbeitung von Schaltungen (Abbildung 4), erstellt der Benutzer eine Oberfläche. Hierzu muss er die Komponenten auf der Arbeitsfläche frei zu einer Schaltung anordnen. Hierzu gehören hinzufügen, verbinden, bearbeiten und rotieren von Komponenten. Um die Übersichtlichkeit zu bewahren, soll der Benutzer auch mehrere Komponenten auswählen und zu einer zusammengesetzten Komponente extrahieren können. Für die Simulation kann der Benutzer Parameter wie zum Beispiel LED-Farbe, Anzahl der Eingänge und Ähnliches festlegen. Um auf der Oberfläche arbeiten zu können, kann der Benutzer zwischen Auswahlmodus und Bewegungsmodus (Verschieben der Arbeitsfläche) umschalten können.



powered by Astah

Abbildung 4: UseCase Schaltung bearbeiten

Der Benutzer möchte das von ihm erstellte Blockschaltbild auch simulieren (Abbildung 5), hierzu wird ihm während der Bearbeitung farblich angezeigt, welche Ein- und Ausgänge der Komponenten mit logisch eins und welche mit logisch null belegt sind. Die Simulation kann auch durch Interaktion mit Komponenten wie zum Beispiel Tastern und Schaltern durch den Benutzer beeinflusst werden. Um den Signalweg nachverfolgen zu können, kann dem Benutzer zusätzlich durch farbliche Markierung der Leitung signalisiert werden, welches Signal der dazugehörige Ein- und Ausgang trägt. Der Benutzer soll jedoch die farbliche Markierung der Leitungen an- und abschalten können.



powered by Astah

Abbildung 5: UseCase Schaltung simulieren

7 Glossar

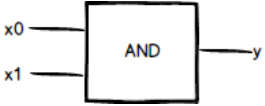
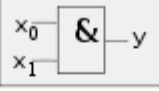
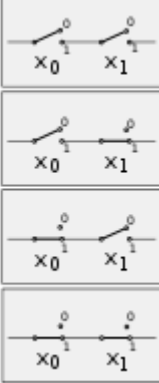
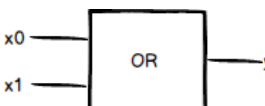
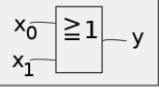
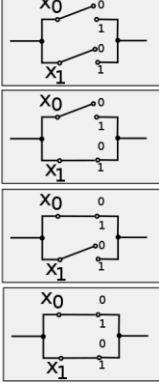
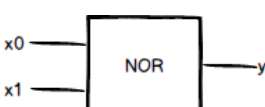
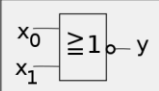
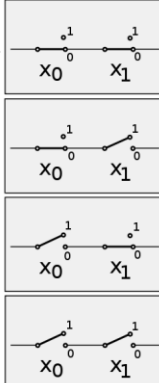
Begriff	Erläuterung/Erklärung
GOLDi	Grid of Online Labs Ilmenau
BEAST	Block Diagram Editing And Simulating Tool
Komponente	Ein Baustein zur Erstellung von Blockschaltbildern, z.B. AND, OR, etc.
Leitung	Verbindung zwischen zwei Komponenten
Bibliothek	Übergeordnete Struktur, welche Komponenten beinhaltet
Projekt	Struktur aus einem Blockschaltbild, Basisbibliotheken und spezifischen Bibliotheken
Basisbibliotheken	Bibliothek mit Grundkomponenten, Bibliothek mit zusammengesetzten Grundkomponenten und eine Bibliothek zur Ablage von neu-erstellten Komponenten, welcher keiner spezifischen Bibliothek zugeordnet sind.
responsive design	Die Oberflächenanordnung und Größenverhältnisse verhalten sich relativ zur Größe des übergeordneten Elementes
Shortcut	Tastenkürzel oder Mausaktion zur schnellen Betätigung von Funktionen


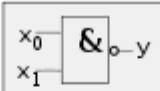
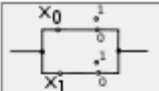
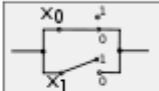
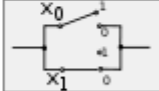
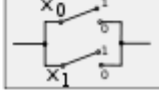

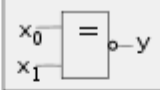
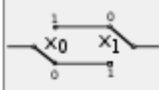
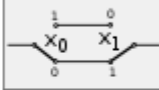
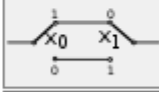
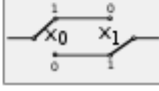
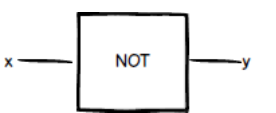
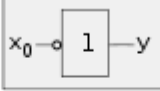
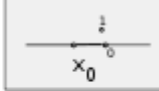
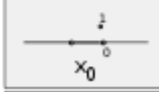
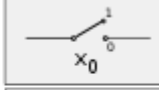
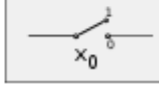

8 Freigabe / Genehmigung

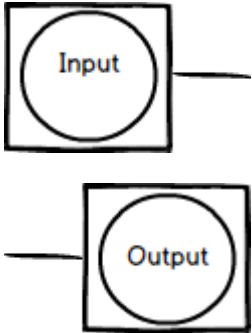

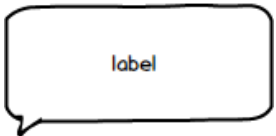
Datum:	
Unterschrift Auftraggeber:	
Unterschrift Projektleiter:	
<Weitere Unterschriften>:	

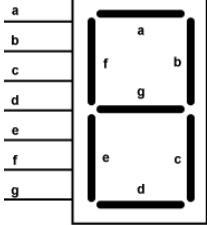
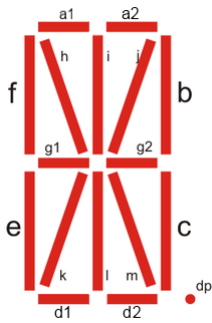
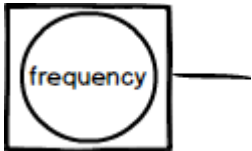
9 Anhang / Ressourcen

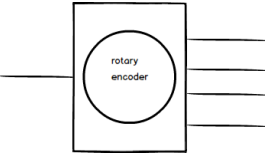
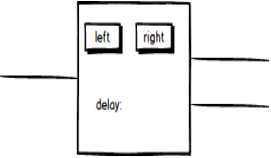
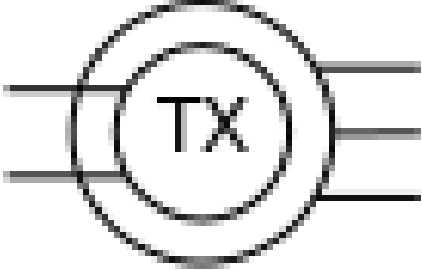
9.1 Grundkomponenten

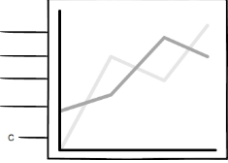

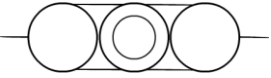
<p>AND</p> 	<p>AND (Konjunktion)</p> <p>DNF: $x_0 x_1$ KNF: $\overline{x_0} \overline{x_1}$ weitere NF:</p> <table border="1" data-bbox="464 607 603 757"> <tr> <td>$x_1 \backslash x_0$</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> </table> <table border="1" data-bbox="687 510 826 719"> <tr> <td>x_0</td> <td>x_1</td> <td>y</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </table>  	$x_1 \backslash x_0$	0	1	0	0	0	1	0	1	x_0	x_1	y	0	0	0	0	1	0	1	0	0	1	1	1	<p>Mögliche Parameter:</p> <ul style="list-style-type: none"> Anzahl der Eingänge <p>Anzeige von Ein- und Ausgängen sollte mithilfe von LEDs dargestellt werden</p> <p>AND-Gatter hat eine klare Aufgabe, der Nutzer sollte nichts bei der Simulation ändern können</p>
$x_1 \backslash x_0$	0	1																								
0	0	0																								
1	0	1																								
x_0	x_1	y																								
0	0	0																								
0	1	0																								
1	0	0																								
1	1	1																								
<p>OR</p> 	<p>OR (Disjunktion)</p> <p>DNF: $x_0 \vee x_1$ KNF: $\overline{x_0} \overline{x_1}$ weitere NF: $\overline{x_0} \vee \overline{x_1}$</p> <table border="1" data-bbox="464 1077 603 1227"> <tr> <td>$x_1 \backslash x_0$</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </table> <table border="1" data-bbox="687 981 826 1189"> <tr> <td>x_0</td> <td>x_1</td> <td>y</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </table>  	$x_1 \backslash x_0$	0	1	0	0	1	1	1	1	x_0	x_1	y	0	0	0	0	1	1	1	0	1	1	1	1	<p>Mögliche Parameter:</p> <ul style="list-style-type: none"> Anzahl der Eingänge <p>Anzeige von Ein- und Ausgängen sollte mithilfe von LEDs dargestellt werden</p> <p>OR-Gatter hat eine klare Aufgabe, der Nutzer sollte nichts bei der Simulation ändern können</p>
$x_1 \backslash x_0$	0	1																								
0	0	1																								
1	1	1																								
x_0	x_1	y																								
0	0	0																								
0	1	1																								
1	0	1																								
1	1	1																								
<p>NOR</p> 	<p>NOR (Pierce-Funktion)</p> <p>DNF: $\overline{x_0} \overline{x_1}$ KNF: $\overline{x_0} \overline{x_1}$ weitere NF: $\overline{x_0} \vee \overline{x_1}$</p> <table border="1" data-bbox="464 1547 603 1697"> <tr> <td>$x_1 \backslash x_0$</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> </table> <table border="1" data-bbox="687 1451 826 1659"> <tr> <td>x_0</td> <td>x_1</td> <td>y</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </table>   <p>$x_0 \downarrow x_1 = \overline{x_0 \vee x_1}$</p>	$x_1 \backslash x_0$	0	1	0	1	0	1	0	0	x_0	x_1	y	0	0	1	0	1	0	1	0	0	1	1	0	<p>Mögliche Parameter:</p> <ul style="list-style-type: none"> Anzahl der Eingänge <p>Anzeige von Ein- und Ausgängen sollte mithilfe von LEDs dargestellt werden</p> <p>NOR-Gatter hat eine klare Aufgabe, der Nutzer sollte nichts bei der Simulation ändern können</p>
$x_1 \backslash x_0$	0	1																								
0	1	0																								
1	0	0																								
x_0	x_1	y																								
0	0	1																								
0	1	0																								
1	0	0																								
1	1	0																								

<p>NAND</p> 	<p>NAND (Sheffer-Funktion. NOT-AND)</p> <p>DNF: $\overline{x_0} \vee \overline{x_1}$ KNF: $\overline{x_0} \vee \overline{x_1}$ weitere NF: $\overline{x_0 x_1}$</p> <table border="1" data-bbox="464 416 608 551"> <tr> <td>$x_1 \backslash x_0$</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </table> <table border="1" data-bbox="687 304 831 528"> <tr> <td>x_0</td> <td>x_1</td> <td>y</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </table> <p>$x_0 x_1 = \overline{x_0 \wedge x_1}$</p>     	$x_1 \backslash x_0$	0	1	0	1	1	1	1	0	x_0	x_1	y	0	0	1	0	1	1	1	0	1	1	1	0	<p>Mögliche Parameter:</p> <ul style="list-style-type: none"> Anzahl der Eingänge <p>Anzeige von Ein- und Ausgängen sollte mithilfe von LEDs dargestellt werden</p> <p>NAND-Gatter hat eine klare Aufgabe, der Nutzer sollte nichts bei der Simulation ändern können</p>
$x_1 \backslash x_0$	0	1																								
0	1	1																								
1	1	0																								
x_0	x_1	y																								
0	0	1																								
0	1	1																								
1	0	1																								
1	1	0																								
<p>XOR</p> 	<p>XOR (Antivalenz, Exklusives-ODER)</p> <p>DNF: $(\overline{x_0} \wedge x_1) \vee (x_0 \wedge \overline{x_1})$ KNF: $(\overline{x_0} \vee \overline{x_1}) \wedge (x_0 \vee x_1)$ weitere NF: $x_0 \neq x_1$</p> <table border="1" data-bbox="464 909 608 1043"> <tr> <td>$x_1 \backslash x_0$</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </table> <table border="1" data-bbox="687 797 831 1021"> <tr> <td>x_0</td> <td>x_1</td> <td>y</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </table>     	$x_1 \backslash x_0$	0	1	0	0	1	1	1	0	x_0	x_1	y	0	0	0	0	1	1	1	0	1	1	1	0	<p>Mögliche Parameter:</p> <ul style="list-style-type: none"> Anzahl der Eingänge <p>Anzeige von Ein- und Ausgängen sollte mithilfe von LEDs dargestellt werden</p> <p>XOR-Gatter hat eine klare Aufgabe, der Nutzer sollte nichts bei der Simulation ändern können</p>
$x_1 \backslash x_0$	0	1																								
0	0	1																								
1	1	0																								
x_0	x_1	y																								
0	0	0																								
0	1	1																								
1	0	1																								
1	1	0																								
<p>NOT</p> 	<p>NOT A (Negation von A, Negation von x_0)</p> <p>DNF: $\overline{x_0}$ KNF: $\overline{x_0}$ weitere NF: $\overline{x_0}$</p> <table border="1" data-bbox="464 1357 608 1491"> <tr> <td>$x_1 \backslash x_0$</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </table> <table border="1" data-bbox="687 1267 831 1491"> <tr> <td>x_0</td> <td>x_1</td> <td>y</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </table>     	$x_1 \backslash x_0$	0	1	0	1	0	1	1	0	x_0	x_1	y	0	0	1	0	1	1	1	0	0	1	1	0	<p>Keine Parameter erforderlich</p> <p>Anzeige von Ein- und Ausgängen sollte mithilfe von LEDs dargestellt werden.</p> <p>NOT-Gatter hat eine klare Aufgabe, der Nutzer sollte nichts bei der Simulation ändern können</p>
$x_1 \backslash x_0$	0	1																								
0	1	0																								
1	1	0																								
x_0	x_1	y																								
0	0	1																								
0	1	1																								
1	0	0																								
1	1	0																								
<p>LED</p> 		<p>Mögliche Parameter:</p> <ul style="list-style-type: none"> Farbe <p>Die LEDs sollten die logischen Zustände der Eingänge bzw. Ausgänge verdeutlichen. Dazu sollte die Farbe je nach Zustand bzw. Benutzung veränderbar sein.</p>																								

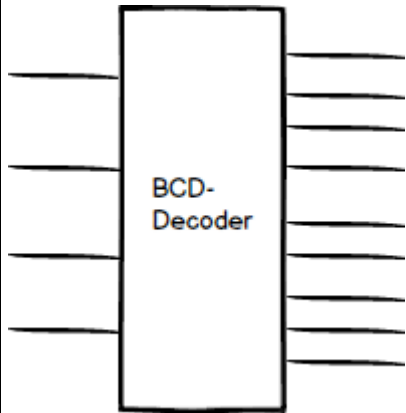
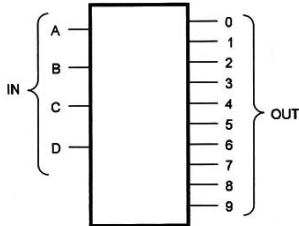
<p>Inputs/Outputs</p> 		<p>Mögliche Parameter</p> <ul style="list-style-type: none"> • Logisch 0/1 <p>Sollte zur Ansteuerung der Gatter bzw. Netze dienen</p> <p>Die Zustände der Inputs/Outputs hängen von der Verknüpfung ab. Sie sollten sich den Veränderungen der Schaltungen anpassen</p>
<p>Knotenpunkte</p> 		<p>Sollte dazu dienen, Logische Zustände zu duplizieren</p> <p>Die Zustände der Knotenpunkte hängt von der Verknüpfung ab. Sie sollten sich den Veränderungen der Schaltung anpassen.</p> <p>Knotenpunkte können auch per Linksklick während dem Verbinden von Komponenten angelegt werden.</p>
<p>Label/Freitext</p> 		<p>Das Label/Freitext sollte dem Nutzer die Möglichkeit geben gewünschten Text einzufügen.</p> <p>Der Nutzer sollte zu jeder Zeit in der Lage sein ein von der Schaltung Unabhängiges Label einfügen zu können, um sich Texte zu vermerken</p>

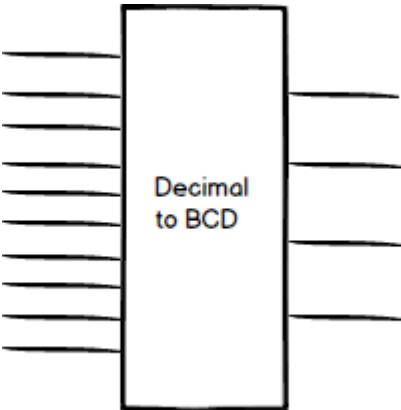
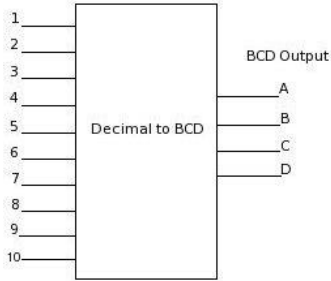
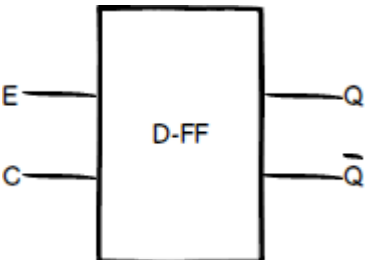
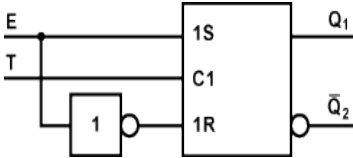
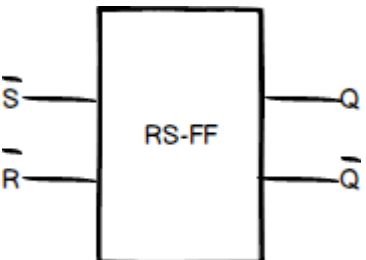
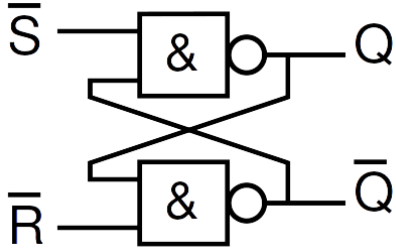

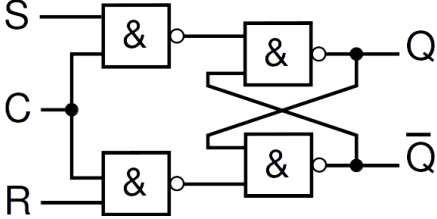
<p>7/16 Segment Anzeige</p> <p>7-Segment-Anzeige</p> 	<p>7 Segment Anzeige</p> <p>16 Segment Anzeige</p> 	<p>Mögliche Parameter</p> <ul style="list-style-type: none"> Farbe <p>Die Segment Anzeige sollte dazu dienen, bestimmte Signaturen/Zeichen/Zahlen darzustellen</p> <p>Die einzelnen Segmente der Anzeige, sollte die gleichen Eigenschaften besitzen wie eine LED (s. oben). Dabei sollte jedes Segment durch einen einzelnen Eingang gesteuert werden.</p>
<p>Frequenzgenerator</p> 		<p>Mögliche Parameter</p> <ul style="list-style-type: none"> Frequenz <p>Der Generator sollte mit einer LED ausgestattet sein, um zu verdeutlichen wann der Generator aktiviert ist</p> <p>In der Komponente sollte die Frequenz variabel per Parameter wählbar sein.</p>

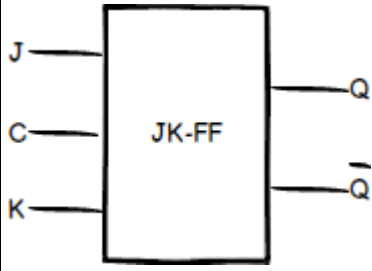
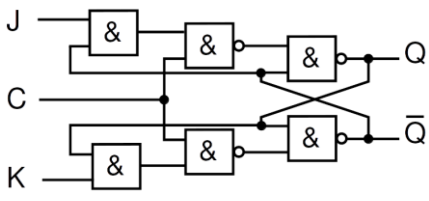
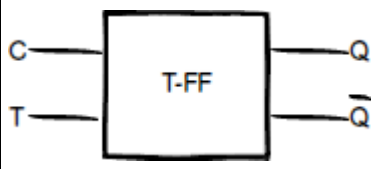
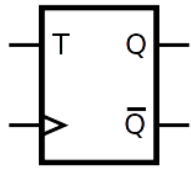

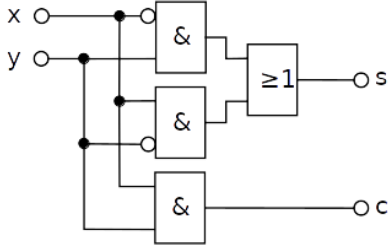
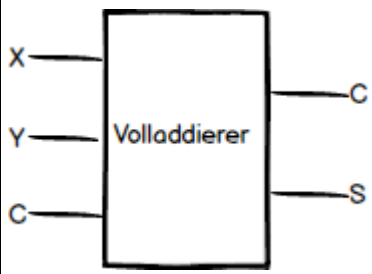
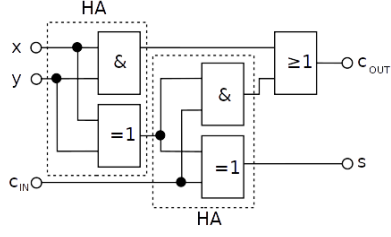

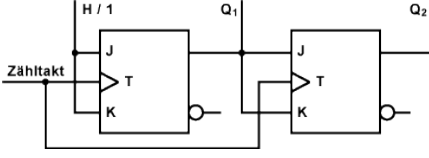
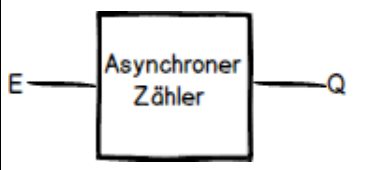
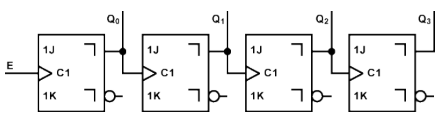
<p>Rotary Encoder (4bit)</p> 		<p>Mögliche Parameter:</p> <ul style="list-style-type: none"> • Belegung an den Eingängen • Encoder für Endsignal <p>Der Rotary Encoder sollte über 4 Eingänge und Ausgänge verfügen</p> <p>Jeder Ein-/Ausgang sollte mit einer LED ausgestattet sein, welche den Logischen Zustand des jeweiligen Ein-/Ausgang verdeutlicht.</p> <p>Bei der Komponente sollte ein Drehknopf(scrollen), Trigger, oder Zahleneingang vorhanden sein, um das Ausgangssignal genau festzulegen</p>
<p>Inkrementalgeber</p> 		<p>Mögliche Parameter</p> <ul style="list-style-type: none"> • Verzögerungszeit <p>Der Inkrementalgeber sollte über einen Knopf verfügen, bei dem man über eine Links-/Rechtsseitige Verschiebung wählen kann. Außerdem sollte er über einen Regler oder Textstelle verfügen, bei der man den Verschiebungswinkel skalieren bzw. frei eingeben kann</p>

<p>Digitales Oszilloskop(4bit)</p> 		<p>Mögliche Parameter</p> <ul style="list-style-type: none"> • Zeit • Belegung an den Eingängen <p>Der Nutzer sollte in der Lage sein, während der Simulation den anzuzeigenden Zeitbereich zu wählen. Dies sollte durch einen Parameter wählbar sein.</p>
<p>Transmitter</p> 		<p>Das zu emittierende Signal ist abhängig von der Eingangsbelegung</p> <p>Der Transmitter sollte über eine LED verfügen, welche die Tätigkeit des Transmitters verdeutlicht (on/off).</p>

9.2 Zusammengesetzte Grundkomponenten

<p>BCD Decoder</p> 		<p>Decodiert BCD Code an den Eingängen x0 bis x3 auf die Ausgänge y0 bis y9.</p>
--	---	--

<p>BCD Encoder</p> 		<p>Codiert die Eingänge x0 bis x9 auf die Ausgänge y0 bis y3.</p>
<p>D-FF</p> 		<p>Taktgesteuertes FlipFlop mit einem Takt-Eingang, einem D Eingang (E in der Abbildung), sowie den Ausgängen Q und -Q.</p>
<p>RS-FF</p> 		<p>Flipflop mit einem Set und einem Reset Eingang sowie Q und -Q Ausgängen.</p>
<p>RS-FF Taktgesteuert</p> 		<p>Flipflop mit einem Set, Reset und CLK Eingang sowie Q und -Q Ausgängen.</p>

<p>JK-FF</p> 		<p>Flipflop mit einem Jump, Kill und CLK Eingang sowie Q und -Q Ausgängen.</p>
<p>T-FF</p> 		<p>Flipflop mit einem Toggle Eingang und einem CLK Eingang, sowie einem Q und -Q Ausgang.</p>
<p>Halbaddierer</p> 		<p>Komponente mit zwei Eingängen und zwei Ausgängen. Die Modulo2-Summe der Eingänge liegt am S-Ausgang an, der Übertrag am C-Ausgang.</p>
<p>Volladdierer</p> 		<p>Addierer mit drei Eingängen X, Y, Cin und zwei Ausgängen Cout und S.</p>
<p>Synchroner Zähler</p> 		<p>Bauteil mit einem Zählengang einem Takteingang und einem Zählerausgang. Durch Verschaltung n dieser Elemente entsteht ein n-Bit Zähler.</p>
<p>Asynchroner Zähler</p> 		<p>Bauteil mit einem Zählengang und einem Zählerausgang. Durch Verschaltung n dieser Elemente entsteht ein n-Bit Zähler.</p>

10 Abbildungsverzeichnis

Abbildung 1: Grafische Benutzeroberfläche BEAST	10
Abbildung 2: UseCases BEAST	16
Abbildung 3: UseCase Projekt organisieren.....	17
Abbildung 4: UseCase Schaltung bearbeiten	18
Abbildung 5: UseCase Schaltung simulieren.....	19