

Digitale Schaltungssimulation im Browser

Benutzerhandbuch - Deutsch

Version 1.1

02.07.2017

Historie der Dokumentversionen

Version	Datum	Autor	Änderungsgrund / Bemerkungen
1.0	25.04.2017	jknu	Ersterstellung
1.1	02.07.2017	jknu	Vorläufige Abgabeverision

Inhaltsverzeichnis

Historie der Dokumentversionen	2
Inhaltsverzeichnis	3
1 BEAST	4
1.1 Allgemeines	4
1.2 Aufruf der Anwendung	4
1.3 Projektstruktur	4
1.4 Bibliotheken	4
1.4.1 Basisbibliotheken	4
1.4.2 Spezifische Bibliotheken	4
1.5 Komponenten	5
1.5.1 Grundkomponenten	5
1.5.2 Spezifische Komponenten	5
1.6 Menschenlesbare Exporte	5
2 Benutzeroberfläche	6
2.1 Menüleiste	6
2.1.1 Projektorganisation	7
2.1.2 Bibliothekorganisation	7
2.2 Navigationsbaum	7
2.3 Arbeitsbereich	8
2.3.1 Tableiste	8
2.3.2 Editor Toolbar	8
2.3.3 Editor Arbeitsfläche	8
3 Blockschaltbildbearbeitung	9
3.1 Schaltungsbearbeitung	10
3.2 Anzeigoptionen	11
4 Schaltungssimulation	12
4.1 Komponentensimulation	12
4.2 Komponenteninteraktion	12
5 Glossar	13
6 Abbildungsverzeichnis	14
7 Anhang	15
7.1 Grundkomponenten	15
7.2 Zusammengesetzte Grundkomponenten	20

1 BEAST

1.1 Allgemeines

BEAST (Blockdiagram Editing And Simulating Tool) ist ein Webmodul, welches es ermöglicht digitale Schaltungen visuell darzustellen und zu simulieren. Dabei können Komponenten zu Blockschaltbildern zusammengefügt werden.

Zusätzlich zur reinen Schaltungsbearbeitung und -simulation bietet BEAST die Organisation von Komponenten in Bibliotheken und Projekten. So können verschiedene Szenarien in Projekt- oder Bibliothekform importiert und exportiert werden.

1.2 Aufruf der Anwendung

Da BEAST ein Webmodul darstellt, wird es über einen Browser aufgerufen. Hierfür kann BEAST entweder direkt über eine URL erreicht werden, oder alternativ als Modul in einer umfassenden Webseite genutzt werden.

Die Möglichkeit einer eingebetteten Version von BEAST bietet den Vorteil, dass Ein- und Ausgänge auch von der umfassenden Webseite gesetzt oder gelesen werden können. Eine genaue Anleitung wie dies geschehen kann, entnehmen Sie bitte der jeweiligen Spezifikation.

1.3 Projektstruktur

BEAST nutzt eine Projektstruktur. Hierbei setzt sich ein Projekt aus einem Projekt-Blockschaltbild zusammen, welches allgemein als übergeordnete Schaltung bezeichnet werden kann. Des Weiteren umfasst ein Projekt auch Bibliotheken.

1.4 Bibliotheken

Die Bibliotheken werden in Basisbibliotheken und spezifische Bibliotheken unterteilt.

1.4.1 Basisbibliotheken

BEAST beinhaltet drei verschiedene Basisbibliotheken. Hierzu gehören die Basiskomponenten-Bibliothek, die Bibliothek der zusammengesetzten Komponenten und eine Ablage-Bibliothek.

Die Komponenten, welche in den Basisbibliotheken angeordnet sind, können nicht bearbeitet werden. Sie können lediglich zur Schaltungserstellung genutzt werden.

Basisbibliotheken können somit nicht editiert oder gelöscht werden.

1.4.2 Spezifische Bibliotheken

Spezifische Bibliotheken können vom Benutzer angelegt werden. Sobald eine Bibliothek vom Benutzer erstellt wurde, kann sie mit Komponenten gefüllt werden. Hierzu verschiebt der Benutzer die Komponenten via Drag&Drop über den Navigationsbaum in die Ziel-Bibliothek. Spezifische Bibliotheken können über die Menüleiste importiert und exportiert werden.

1.5 Komponenten

Die Komponenten werden ebenfalls in Grundkomponenten und spezifische Komponenten unterteilt. Grundkomponenten sind in den Basisbibliotheken des Projekts enthalten.

1.5.1 Grundkomponenten

Die Grundkomponenten, sind in jeder BEAST-Instanz vorhanden und können immer verwendet werden. Zusammengesetzte Basiskomponenten bilden eine Struktur aus Basiskomponenten und können nicht bearbeitet werden

1.5.2 Spezifische Komponenten

Benutzer können in BEAST auch spezifische Komponenten erstellen. Hierzu können Basiskomponenten zu eigenen Strukturen zusammengesetzt und anschließend gespeichert werden. Einzelne Komponenten können nicht exportiert werden. Allerdings können sie innerhalb spezifischer Bibliothek gespeichert werden, welche exportierbar sind.

1.6 Menschenlesbare Exporte

Sowohl Projekt- als auch Bibliothek-Exporte können ohne die Verwendung von BEAST vom Benutzer mit einem Texteditor eingesehen werden. Die Struktur der Dateien ist menschenlesbar. Ihre Struktur ist wie folgt aufgebaut.

Im Circuit stehen zuerst die ID des Projekts und der Name. Anschließend folgt eine Aufzählung „devices“, in welchem alle im Arbeitsbereich befindlichen Komponenten mit ihren Daten stehen. Jede hat einen Typ welcher sich aus der Bibliothek und dem Namen des Bauteils zusammensetzt. Darunter befindet sich noch die eindeutige ID der Komponente selbst und Koordinaten, wo sie im Arbeitsbereich platziert ist. Nach den „devices“ folgt eine Aufzählung namens „connectors“. Dort befinden sich alle Verbindungen. Anschließend folgt die Version von BEAST und die von den Basis- und zusammengesetzten Komponenten verschiedenen Bibliotheken mit ihrer ID, ihrem Namen, Versionsnummer und den in ihr befindlichen Komponenten.

2 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.

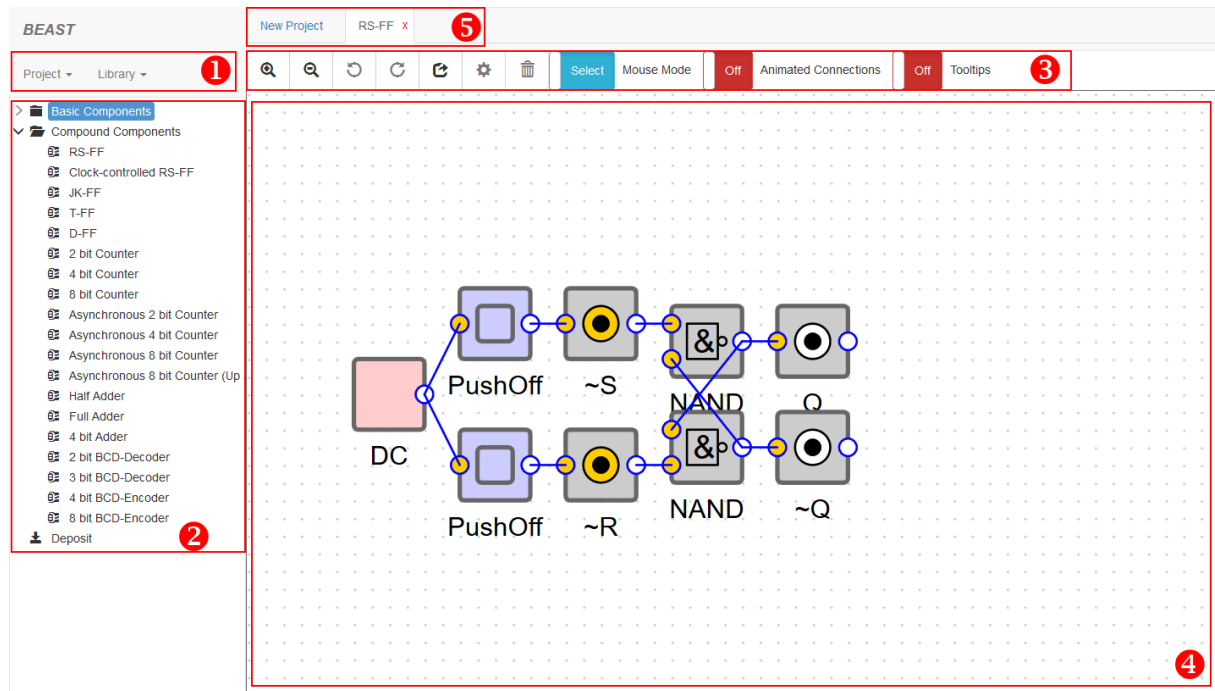


Abbildung 1: Grafische Benutzeroberfläche BEAST

2.1 Menüleiste

Die Menüleiste dient der BEAST-Projektorganisation (siehe 1 in **Fehler! Verweisquelle konnte nicht gefunden werden.**). Sie ist unterteilt in „Project“ und „Library“.

2.1.1 Projektorganisation

Unter dem Reiter „Project“ befinden sich die Optionen für BEAST-Projekte.

New	Über den Menüeintrag „New“ kann ein neues Projekt geöffnet werden. Ist das aktuelle Projekt nicht gespeichert, hat der Benutzer die Möglichkeit das aktuelle Projekt zu speichern, als .beast Datei zu exportieren oder den Vorgang abubrechen.
Open	Wählt der Benutzer die Option „Open“, kann er ein Projekt aus seinem lokalen Speicher auswählen und laden. Projekte werden kontinuierlich im Zwischenspeicher abgelegt.
Delete	Gespeicherte Projekte können über die Option „Delete“ gelöscht werden. Hierzu wählt der Benutzer das zu löschende Projekt aus der Liste der gespeicherten Projekte aus.
Export	Projekte können als .beast Dateien auf dem lokalen System gespeichert werden. Hierzu wählt der Benutzer den Menüpunkt „Export“. Anschließend gibt er dem Projekt einen Namen und startet den Download der .beast Datei.
Import	Exportierte Projekte können jederzeit wieder in BEAST importiert werden. Hierzu wählt der Benutzer die Option „Import“. Anschließend kann eine lokal gespeicherte .beast Datei ausgewählt und importiert werden.

2.1.2 Bibliothekorganisation

Unter dem Reiter „Project“ befinden sich die Optionen für BEAST-Bibliotheken.

New	Über den Menüeintrag „New“ kann eine neue Bibliothek angelegt werden. Der Benutzer wählt einen Namen und die neue Bibliothek wird dem Projekt hinzugefügt.
Export	Bibliotheken können als .bdcl Dateien auf dem lokalen System gespeichert werden. Hierzu wählt der Benutzer den Menüpunkt „Export“. Anschließend wählt er die zu exportierenden Bibliotheken, welche als eine .bdcl Datei mit allen in den ausgewählten Bibliotheken vorhandenen Komponenten exportiert werden.
Import	Exportierte Bibliotheken können jederzeit wieder in BEAST importiert werden. Hierzu wählt der Benutzer die Option „Import“. Anschließend kann eine lokal gespeicherte .bdcl Datei ausgewählt und importiert werden.

2.2 Navigationsbaum

Der Navigationsbaum auf der linken Seite der Anwendung (siehe 2 in **Fehler! Verweisquelle konnte nicht gefunden werden.**) zeigt die Struktur des Projektes. Es werden alle Bibliotheken (sowohl Basisbibliotheken, als auch spezifische) als Ordner dargestellt. Die Ordner enthalten in einer niedrigeren Strukturebene die Komponenten.

Innerhalb des Baumes kann der Benutzer mittels eines kontextsensitiven Rechtsklick-Menüs Änderungen an der Projektstruktur durchführen. So können spezifische Bibliotheken oder Komponenten gelöscht oder editiert werden.

Der Baum dient ebenfalls als Toolbox. Alle Komponenten können per Drag&Drop zur Schaltungserstellung auf die Arbeitsfläche gezogen werden. Eine Ausnahme stellt der Ablage-Ordner „Deposit“ dar. Komponenten, welche im Deposit abgelegt wurden, können nicht der Arbeitsfläche hinzugefügt werden.

Zusammengesetzte Komponenten können per Doppelklick als neuer Tab in der Tableiste dargestellt werden.

2.3 Arbeitsbereich

Der Arbeitsbereich setzt sich aus Tab-Struktur, Editor-Toolbar und Editor-Arbeitsbereich zusammen.

2.3.1 Tableiste

Die Tableiste (siehe 5 in **Fehler! Verweisquelle konnte nicht gefunden werden.**) dient zur parallelen Darstellung von Schaltungen. So können zusammengesetzte Strukturen in einem neuen Tab geöffnet werden. So können einzelne zusammengesetzte Komponenten bearbeitet werden, während die Hauptschaltung in einem gesonderten Tab geöffnet bleibt.

Der Tab der Hauptschaltung des Projekts kann nicht geschlossen werden.

Werden an einer bereits verwendeten Struktur Änderungen vollzogen, muss die Struktur als neue zusammengesetzte Komponente gespeichert werden. Anschließend muss der Benutzer die Komponente aus seiner Schaltung durch die neu angelegte Komponente, welche die Änderungen beinhaltet, ersetzen.

2.3.2 Editor Toolbar

Die Editor-Toolbar (siehe 3 in **Fehler! Verweisquelle konnte nicht gefunden werden.**) bietet sowohl Möglichkeiten zur Schaltungsbearbeitung als auch Optionen für die Anzeige und die Simulation.

2.3.3 Editor Arbeitsfläche

Die Arbeitsfläche (siehe 4 in **Fehler! Verweisquelle konnte nicht gefunden werden.**) stellt die Schaltung als Struktur von Komponenten dar. Innerhalb der Schaltung können zusammengesetzte Komponenten via Doppelklick in einem neuen Tab geöffnet werden.

Mit der Arbeitsfläche kann der Benutzer zusammengesetzte Strukturen durch Verbinden von Komponenten erzeugen. Des Weiteren bietet die Arbeitsfläche eine Live-Simulation der logischen Belegungen aller Ein- und Ausgänge.

3 Blockschaltbildbearbeitung

Um Blockschaltbilder zu erstellen, befinden sich im Navigationsbaum Grundkomponenten (basic components), zusammengesetzte Komponenten (compound components) und eine Ablage für selbst-erstellte Komponenten (Deposit). Es ist möglich mit Hilfe von Drag&Drop einzelne Komponenten aus dem Baum auf die Arbeitsfläche zu verschieben.

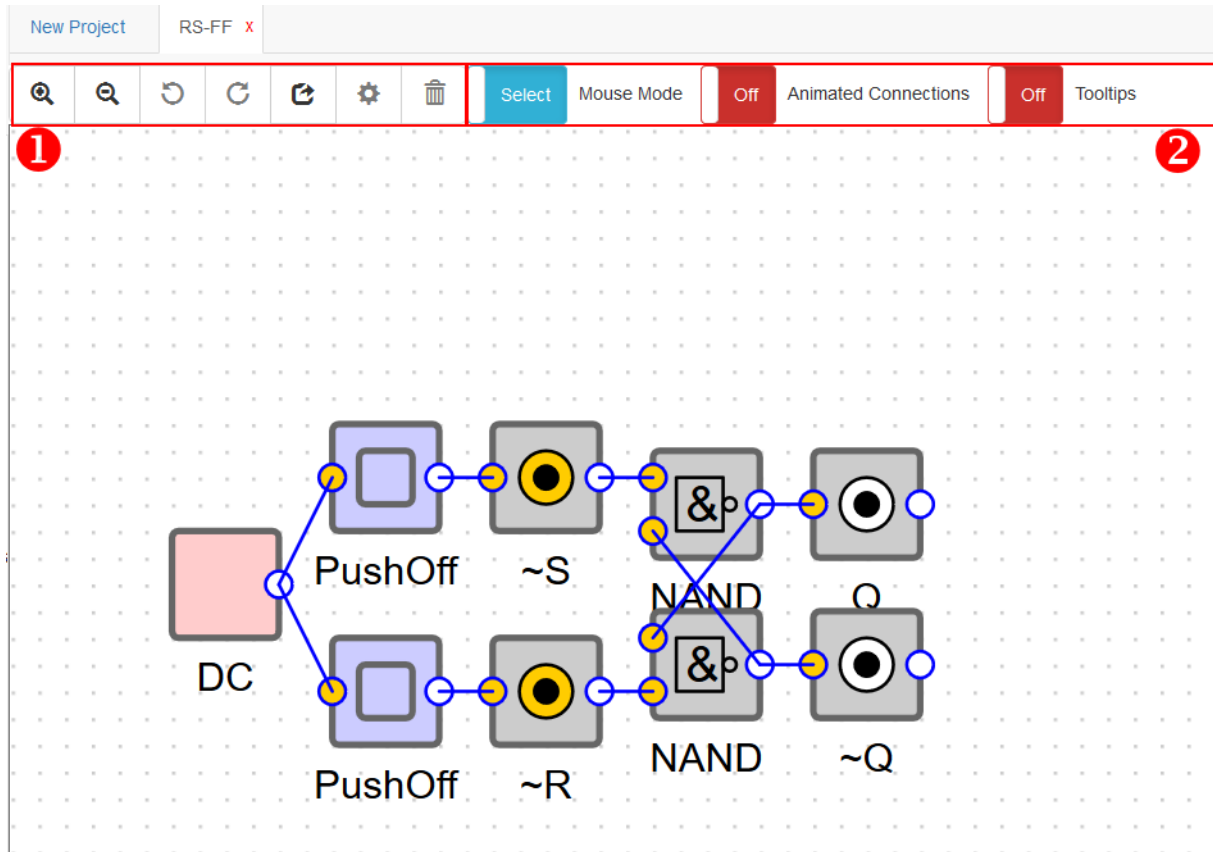



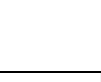


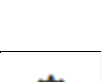


Abbildung 2: Arbeitsbereich

In der Arbeitsfläche können die einzelnen Komponenten per Mausklick verschoben und angeordnet werden. Außerdem kann mit Hilfe der Maus ein Bereich ausgewählt und alle in ihm liegenden Komponenten verschoben werden. Zieht man eine Komponente in einen Bereich außerhalb der Arbeitsfläche, wird sie gelöscht. Des Weiteren ist es möglich Komponenten miteinander zu verknüpfen. Dies kann mit einem Links-Klick auf einen Ein- oder Ausgang einer Komponente und dem Loslassen der Maustaste auf dem zu verbindenden Ein- oder Ausgang bewerkstelligt werden. Dabei ist zu beachten, dass an jedem Eingang nur eine Leitung angelegt werden kann. Verbindet man eine Leitung zu einem bereits angeschlossenen Eingang, wird diese Verbindung erstellt und die vorherige gelöscht. Um alles übersichtlicher gestalten zu können, ist es möglich Knotenpunkte während des Verbindens durch Betätigung der Leertaste zu erstellen. Ist eine Verbindung noch nicht angeschlossen, kann auch der jeweils zuletzt erstellte Knoten durch die Rücktaste gelöscht werden.

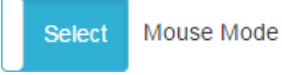
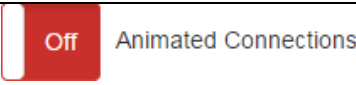
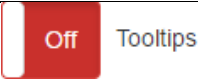
3.1 Schaltungsbearbeitung

Um besser und übersichtlicher zu arbeiten, befinden sich in der Toolbar über dem Arbeitsbereich verschiedene Schaltflächen (siehe 1 in Abbildung 2).

	Vergrößern: Bei einem Klick auf den Zoom-In-Button wird der Arbeitsbereich verkleinert und alle in ihm enthaltenen Elemente vergrößert. Dies funktioniert auch durch das Rollen des Mausekzes nach unten.
	Verkleinern: Bei einem Klick auf den Zoom-Out-Button wird der Arbeitsbereich vergrößert und alle in ihm enthaltenen Elemente verkleinert. Dies funktioniert auch durch das Rollen des Mausekzes nach oben.
	Links-Rotieren: Sind ein oder mehrere Elemente in der Arbeitsfläche ausgewählt, ist es möglich, diese durch einen Klick auf die Rotate-Left-Schaltfläche um 90° nach links zu drehen. Hierbei drehen sich alle Komponenten jeweils um den Mittelpunkt der ausgewählten Elemente. Es kann ebenfalls das Tastenkürzel „Strg“ + „Pfeiltaste Links“ verwendet werden.
	Rechts-Rotieren: Sind ein oder mehrere Elemente in der Arbeitsfläche ausgewählt, ist es möglich, diese durch einen Klick auf die Rotate-Right-Schaltfläche um 90° nach rechts zu drehen. Hierbei drehen sich alle Komponenten jeweils um den Mittelpunkt der ausgewählten Elemente. Es kann ebenfalls das Tastenkürzel „Strg“ + „Pfeiltaste Rechts“ verwendet werden.
	Komponente extrahieren: Durch einen Klick auf den Extract-Component-Button wird der Arbeitsbereich als neue Komponente extrahiert. Diese wird im Deposit abgelegt und kann anschließend als eigene zusammengesetzte Komponente bearbeitet und verwendet werden.
	Parameter setzen: Ist eine einzelne Komponente markiert, ist es möglich den Set-Parameter-Button anzuklicken. Daraufhin öffnet sich ein Fenster, in dem alle Parameter der Komponente geändert werden können. Durch einen Klick auf die Apply Schaltfläche im Fenster werden die Änderungen übernommen. Die Parameter-Bearbeitung kann auch durch einen Doppelklick auf eine Komponente ausgelöst werden.
	Löschen: Sind ein oder mehrere Elemente in der Arbeitsfläche ausgewählt, ist es möglich diese durch einen Klick auf den Delete-Button oder Drücken „Entf“-Taste aus der Arbeitsfläche zu entfernen.

3.2 Anzeigoptionen

Um die Schaltung zu analysieren, können in der Toolbar verschiedene Anzeigoptionen durch Schalter gewählt werden (siehe 2 in Abbildung 2).

	<p>Mit einem Klick auf den Mouse-Mode-Switch kann man zwischen Select und Move umschalten. Im Select-Modus ist es möglich, mit Hilfe einer Bereichsauswahl mehrere Komponenten auszuwählen. Außerdem können bereits platzierte Elemente neu angeordnet werden. Im Move-Modus kann man durch Anklicken, Gedrückt-halten und Bewegen der Maus den sichtbaren Arbeitsbereich in alle Richtungen verschieben.</p>
	<p>Durch einen Klick auf den Animated-Connections-Switch wird die Leitungssimulation in der Arbeitsfläche an- bzw. abgeschaltet. Die Leitungen zeigen dabei den logischen Zustand ihres Quell- und Zielports.</p>
	<p>Durch einen Klick auf den Tooltips-Switch kann man die Tooltipfunktion an- bzw. abschalten. Ist diese angeschalten, wird unter allen markierten Komponenten eine Anzeige mit ihren Parametern angezeigt. Der Name eines Bauteils ist zwar ein Parameter, wird aber nicht extra ausgeführt.</p>

4 Schaltungssimulation

Eine in BEAST angelegte Schaltung wird während der Bearbeitung simuliert. Hierzu werden an den Ein- und Ausgängen aller Komponenten die jeweilige Belegung mit eins oder null unterschiedlich farblich dargestellt. Voraussetzung hierfür ist die ausgewählte Option der Schaltungssimulation in der Editor-Toolbar.

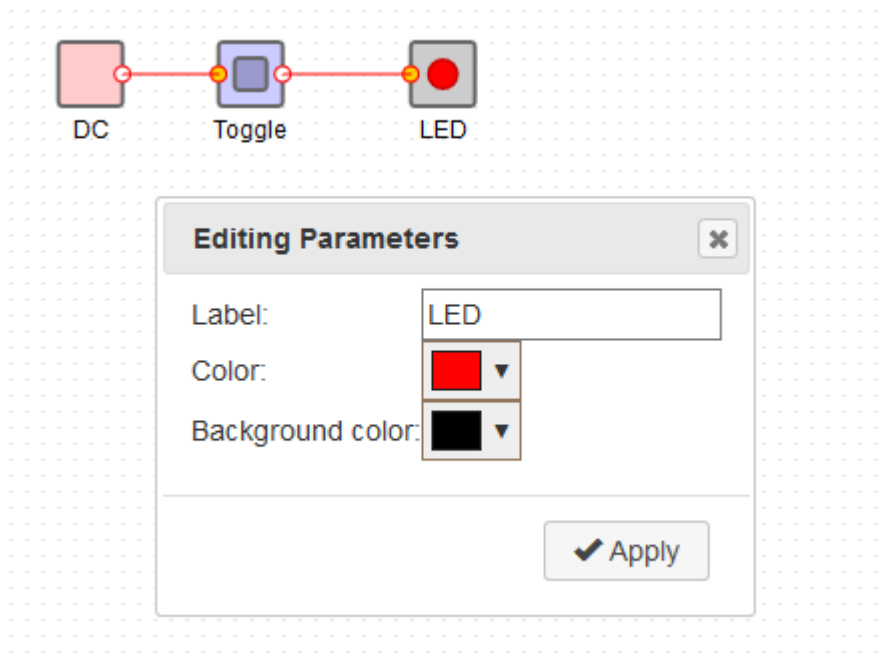


Abbildung 3: Simulation

4.1 Komponentensimulation

Verschiedene Komponenten besitzen Möglichkeiten zur zusätzlichen Darstellung der Simulation. Ein Beispiel stellt die LED dar (siehe Abbildung 3). Je nach Belegung ist sie ein- oder ausgeschaltet.

Die Spezifikation, wie die unterschiedlichen Komponenten zur Simulation beitragen entnehmen Sie bitte der Komponententabelle im Anhang.

4.2 Komponenteninteraktion

Einige Komponenten bieten die Möglichkeit die Schaltung zu beeinflussen und logische Belegungen zu verändern. Ein Beispiel stellt der Toggle-Button dar. Klickt der Benutzer auf den Toggle-Button, wechselt dessen Ausgangsbelegung zwischen logisch eins und logisch null. Die Spezifikation, mit welchen Komponenten interagiert werden kann, entnehmen Sie bitte der Komponententabelle im Anhang.

5 Glossar

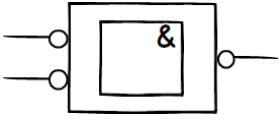
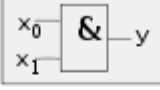
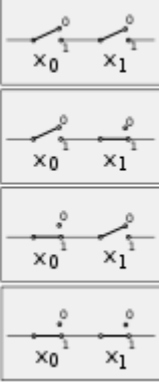
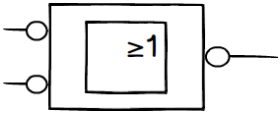
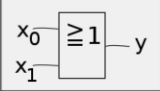
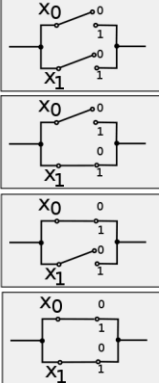
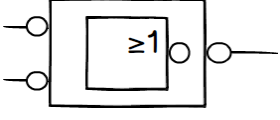
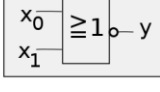
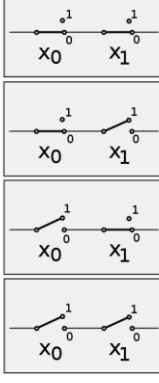
Begriff	Erläuterung/Erklärung
GOLDi	Grid of Online Labs Ilmenau
Webmodul	Web-Anwendung, welche als Modul in bestehende Webseiten aufgenommen werden kann
BEAST	Block Diagram Editing And Simulating Tool
Komponente	Ein Baustein zur Erstellung von Blockschaltbildern, z.B. AND, OR, etc.
Leitung	Verbindung 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.
Shortcut	Tastenkürzel oder Mausaktion zur schnellen Betätigung von Funktionen

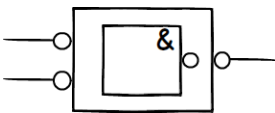

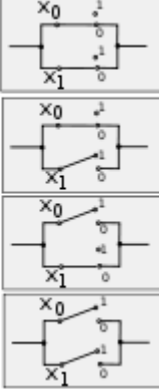
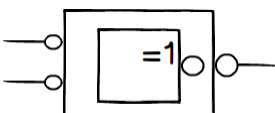
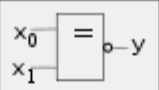
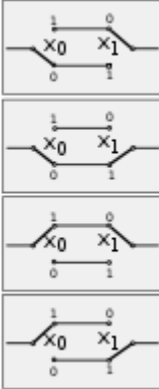
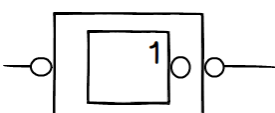
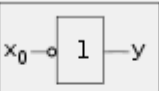
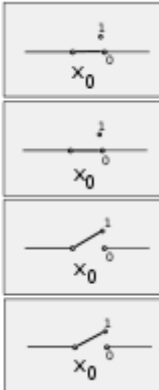
6 Abbildungsverzeichnis

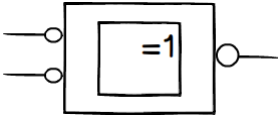

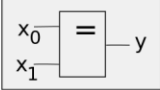
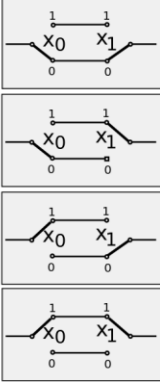
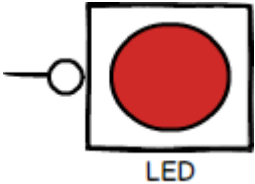
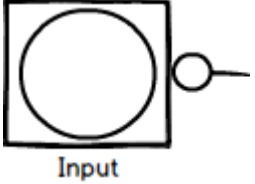
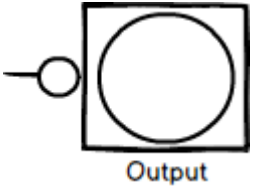
Abbildung 1: Grafische Benutzeroberfläche BEAST	6
Abbildung 2: Arbeitsbereich	9
Abbildung 3: Simulation	12


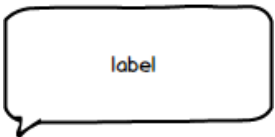
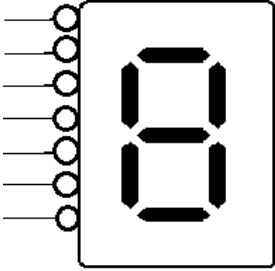
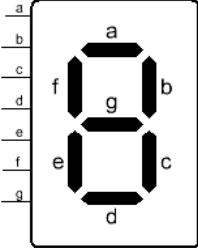
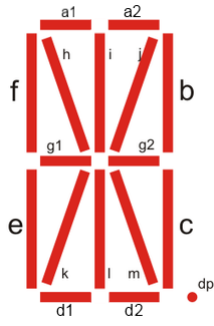
7 Anhang

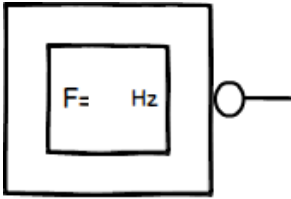
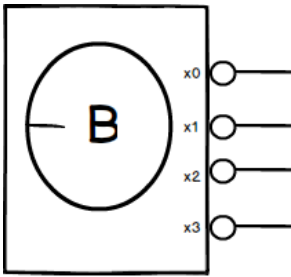
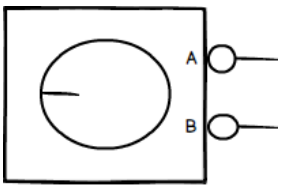
Grundkomponenten

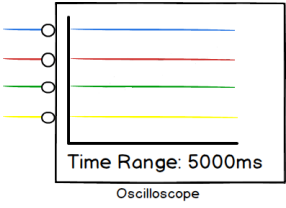
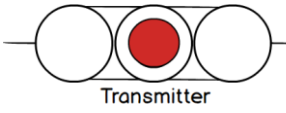
<p>AND</p> 	<p>AND (Konjunktion)</p> <p>DNF: $x_0 x_1$ KNF: $\overline{x_0} \overline{x_1}$ weitere NF: $\overline{x_0 \downarrow x_1}$</p> <table border="1" data-bbox="480 607 624 752"> <tr> <td>$x_1 \backslash x_0$</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> </table> <table border="1" data-bbox="703 506 839 719"> <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: $x_0 \downarrow x_1$</p> <table border="1" data-bbox="480 1077 624 1223"> <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="703 976 839 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: $\overline{x_0 \downarrow x_1}$</p> <table border="1" data-bbox="480 1559 624 1704"> <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="703 1458 839 1671"> <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="475 421 624 562"> <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="703 309 842 533"> <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="475 913 624 1055"> <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="703 801 842 1025"> <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="475 1366 624 1507"> <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="703 1254 842 1478"> <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																								

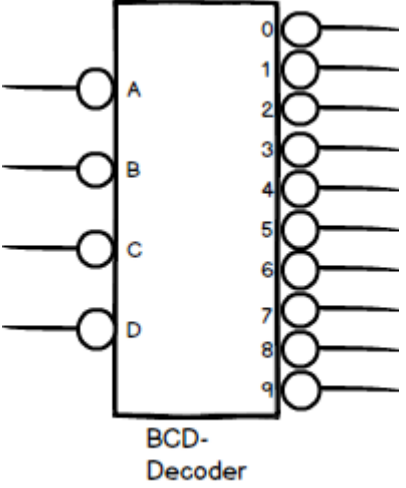
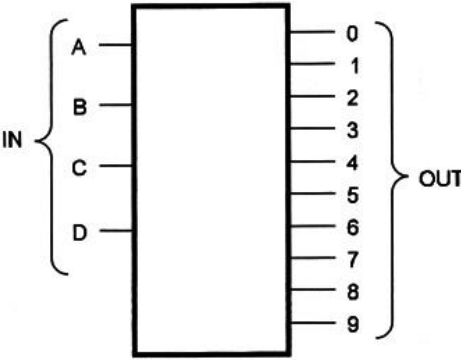
<p>XNOR</p> 	<p>Äquivalenz (XNOR; NXOR)</p> <p>DNF: $(x_0 \wedge x_1) \vee (\overline{x_0} \wedge \overline{x_1})$ KNF: $(x_0 \vee \overline{x_1}) \wedge (\overline{x_0} \vee x_1)$ weitere NF: $x_0 \leftrightarrow x_1$ $x_0 \sim x_1$</p> <table border="1" data-bbox="459 432 592 573"> <tr> <td>$x_1 \backslash x_0$</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>1</td></tr> </table>   	$x_1 \backslash x_0$	0	1	0	1	0	1	0	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>NXOR-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	1									
<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>Input/Output</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>									

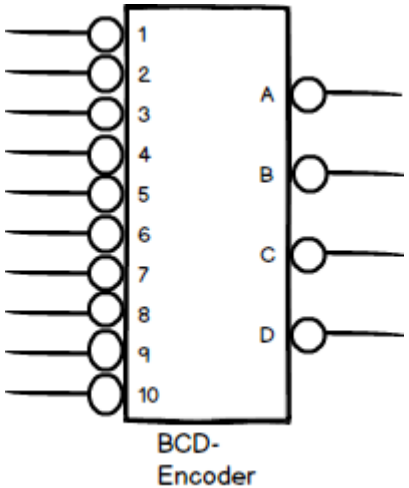
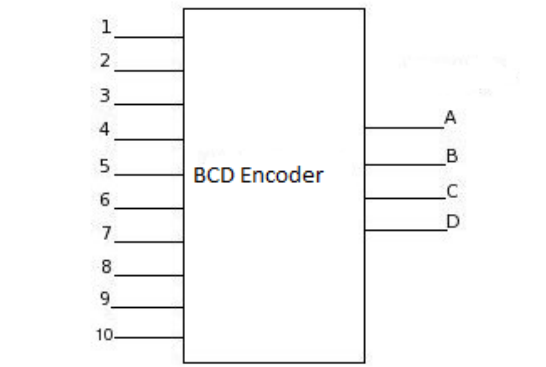
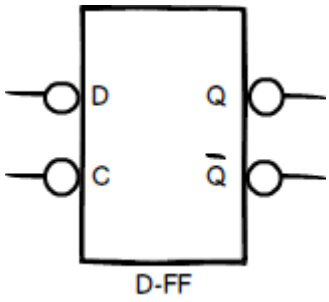
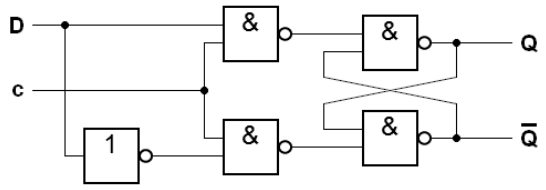
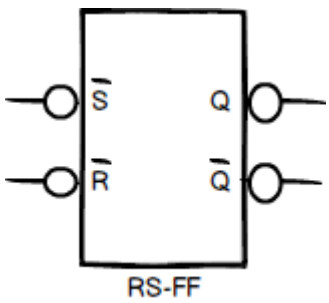
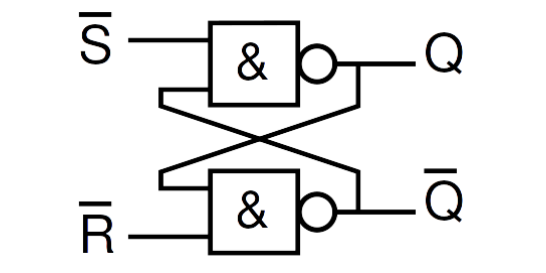
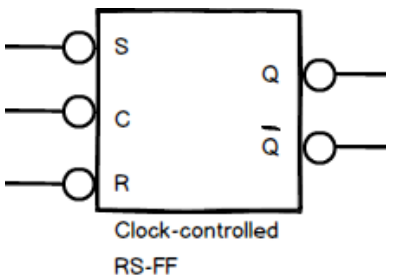
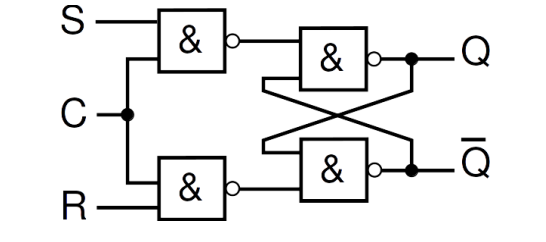
<p>Node</p> 		<p>Sollte dazu dienen, Logische Zustände zu duplizieren.</p> <p>Die Zustände der Knotenpunkte hängt von der Verknüpfung ab. Sie sollten sich den Veränderungen der Schaltung anpassen.</p> <p>Knotenpunkte können auch per Linksklick während dem Verbinden von Komponenten angelegt werden.</p>
<p>Label</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 einzufügen zu können, um sich Texte zu vermerken</p>
<p>7/16 Segment Display</p>  <p>7 Segment Display</p>	<p>7 Segment Display</p>  <p>16 Segment Display</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>

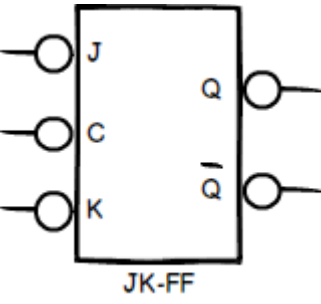
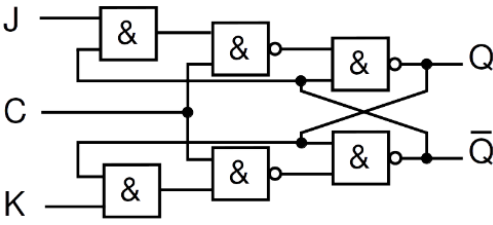
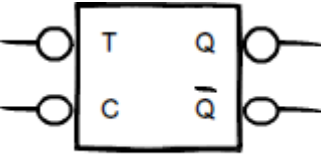
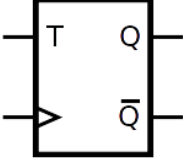
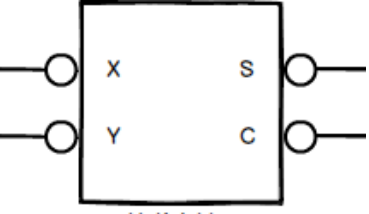
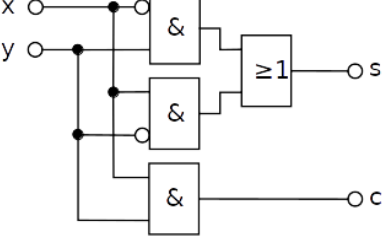
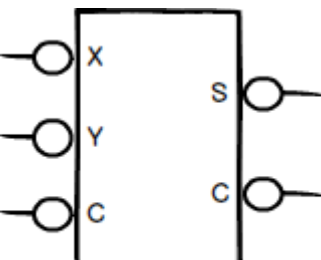
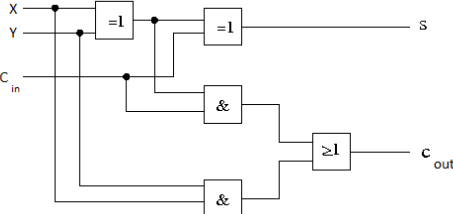
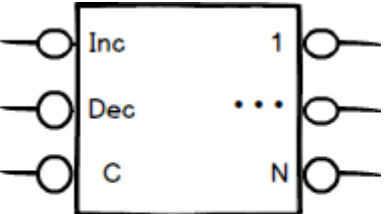
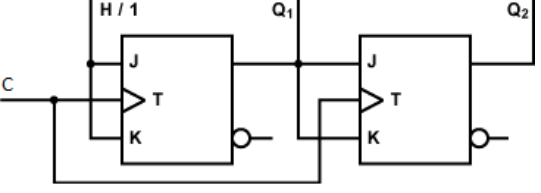
<p>Oscillator</p>  <p>Oscillator</p>		<p>Mögliche Parameter</p> <ul style="list-style-type: none"> • Frequenz <p>Der Generator sollte mit einer LED ausgestattet sein, um zu verdeutlichen wann der Generator aktiviert ist.</p> <p>In der Komponente sollte die Frequenz variabel per Parameter wählbar sein.</p>
<p>Rotary Encoder (4bit)</p>  <p>Rotary Encoder</p>		<p>Mögliche Parameter:</p> <ul style="list-style-type: none"> • Encoder für Endsignal <p>Der Rotary Encoder sollte über 4 Ausgänge verfügen.</p> <p>Jeder Ausgang sollte mit einer LED ausgestattet sein, welche den Logischen Zustand des jeweiligen Ausganges verdeutlicht.</p> <p>Bei der Komponente sollte ein Drehknopf(scrollen), Trigger, oder Zahleneingang vorhanden sein, um das Ausgangssignal genau festzulegen</p>
<p>Incremental Encoder</p>  <p>Incremental Encoder</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>Oscilloscope(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>

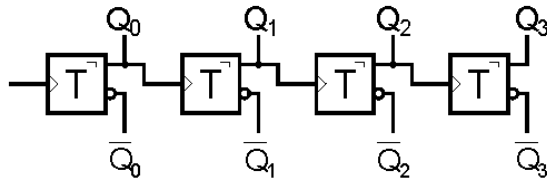
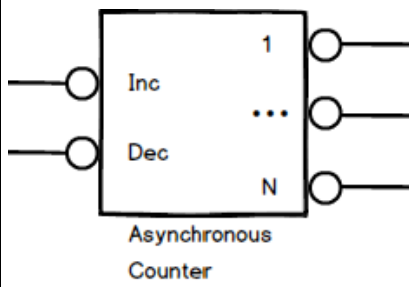
Zusammengesetzte Grundkomponenten

<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>BCD-Encoder</p>		<p>Codiert die Eingänge x0 bis x9 auf die Ausgänge y0 bis y3.</p>
<p>D-FF</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>RS-FF</p>		<p>Flipflop mit einem Set und einem Reset Eingang sowie Q und -Q Ausgängen.</p>
<p>Clock-controlled RS-FF</p>  <p>Clock-controlled RS-FF</p>		<p>Flipflop mit einem Set, Reset und CLK Eingang sowie Q und -Q Ausgängen.</p>

<p>JK-FF</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>T-FF</p>		<p>Flipflop mit einem Toggle Eingang und einem CLK Eingang, sowie einem Q und -Q Ausgang.</p>
<p>Half Adder</p>  <p>Half Adder</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>Full Adder</p>  <p>Full Adder</p>		<p>Addierer mit drei Eingängen X, Y, C_{in} und zwei Ausgängen C_{out} und S.</p>
<p>Parallel Counter</p>  <p>Parallel Counter</p>		<p>Bauteil mit einem Zähleingang einem Takteingang und einem Zählerausgang. Durch Verschaltung n dieser Elemente entsteht ein n-Bit Zähler.</p>

Asynchronous Counter



Bauteil mit einem Zähleingang und einem Zählerausgang. Durch Verschaltung n dieser Elemente entsteht ein n-Bit Zähler.