

Aufgabenstellung

29. 11. 2022

Digitaltechnik - Challenge 2 Abacus 2.0

Institut für Technik der Informationsverarbeitung

Prof. Dr.-Ing. Dr. h.c. Jürgen Becker

M.Sc. Julian Höfer

Wintersemester 22/23



Challenge 2

Abacus 2.0

Das Ziel der zweiten Challenge lässt sich sehr kurz zusammenfassen:

Es soll im Logiksimulator Circuitverse der Rechner *Abacus 2.0* entworfen werden, der die folgenden Eigenschaften erfüllt:

- Addition von zweistelligen Dezimalzahlen (mithilfe des Stibitz-Codes)
- Die Ein- und Ausgabe soll im besser lesbaren BCD-Code erfolgen
- Der Ergebniswertebereich soll in dieser Challenge [0, 198] betragen.
- Es sollen nur Und-, Oder-, Nicht- und Entweder-Oder-Gatter, sowie Ein- und Ausgabe verwendet werden. Eine genaue Auflistung und kurze Beschreibung aller benötigten "Elemente" ist in [Tabelle 2.3](#) zusammengefasst.

2.1 Abgabe an die Tutoren

Die Challenge wird im Logiksimulator Circuitverse bearbeitet, welchen wir selbst unter folgender URL-Adresse hosten:

dt-simulator.itiv.kit.edu

Wird die Challenge in Circuitverse in der jeweiligen Tutoriengruppe als Assignment bearbeitet, können die Tutoren den Fortschritt verfolgen. Eine Anleitung für die korrekte Anmeldung in Circuitverse ist separat in Ilias zu finden. Bewertet wird die in Circuitverse erstellte Schaltung nach Ende der Deadline (voraussichtlich nach Tutoriumswoche 5).

2.2 Hinweise und Tipps

2.2.1 Ein- und Ausgabe

Wenn die Digitalschaltung für den Abacus 2.0 in Circuitverse fertig ist, kann diese auch auf einem Evaluations-Board im ITIV-Poolraum getestet werden. Damit die Synthese der Schaltung funktioniert, müssen die Schnittstellen, also Ein- und Ausgabe, eindeutig definiert werden. Das geschieht durch die Benennung der Ein- und Ausgabeelemente mit Labels.

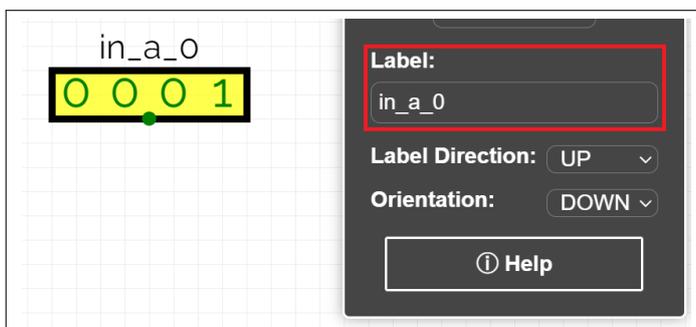


Abbildung 2.1 Ein- und Ausgabeelemente können mit einem Label benannt werden.

Die Ein- und Ausgabeelemente müssen folgendermaßen benannt sein:

Ein- / Ausgabeelement	Bitbreite	Label
1. Summand, Einerstelle	4	in_a_0
1. Summand, Zehnerstelle	4	in_a_1
2. Summand, Einerstelle	4	in_b_0
2. Summand, Zehnerstelle	4	in_b_1
Summe Einerstelle	4	out_0
Summe Zehnerstelle	4	out_1
Summe Hunderterstelle	4	out_2

Tabelle 2.1 Benennung der Ein- und Ausgabeelemente

2.2.2 Subcircuits

Um die stetig steigende Komplexität moderner Hardwarearchitekturen beherrschbar zu halten basiert die digitale Hardwareentwicklung auf drei mächtigen Designparadigmen:

- Modularität ("Baukasten-Prinzip")
- Hierarchie ("Teile-und-Herrsche")
- Abstraktion ("Verstecke-Details")

Diese Paradigmen können in CircuitVerse vereinfacht mit Subcircuits, umgesetzt werden, siehe [Abbildung 2.2](#). Beim Erstellen eines Subcircuits wird ein neues Blatt hinzugefügt. Die Definition der Schnittstellen erfolgt wieder über Input- und Outputelemente, welche ein Label tragen müssen.

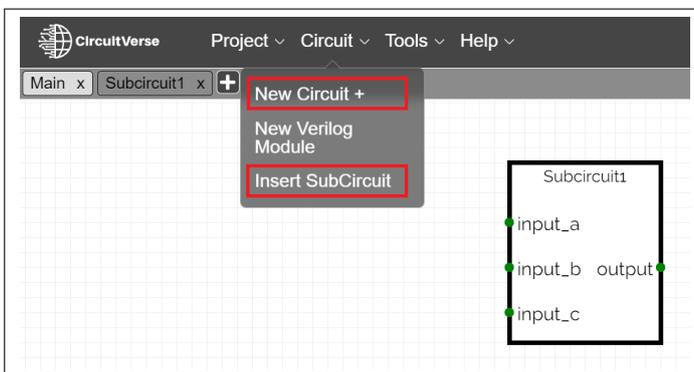


Abbildung 2.2 Erstellen und Einfügen von Subcircuits. Hier ein Beispiel mit drei Eingängen und einem Ausgang

2.2.3 Übersicht erforderlicher Elemente

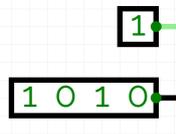
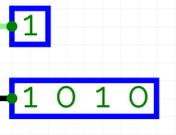
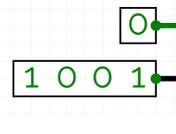
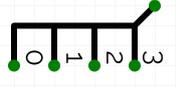
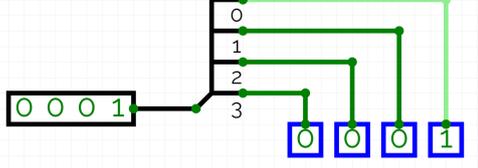
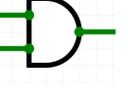
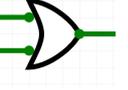
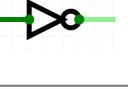
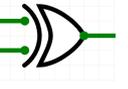
Name	Symbol	Beschreibung
Input (Input)		Eingabe von Binärwerten erfolgt durch Klicken auf das entsprechende Bit. Bitbreite (BitWidth) ist im rechten Menü einstellbar.
Output (Output)		Ausgabe von Binärwerten. Bitbreite (BitWidth) ist im rechten Menü einstellbar.
ConstantVal (Input)		Definition von konstanten Binärwerten. Bitbreite ist auch hier variabel. Die Eingabe des Werts erfolgt bei Doppelklick über ein Abfragemenü.
Splitter (Misc)		<p>Splitter teilen Leitungen (Bitvektoren) mit großer Bitbreite in Leitungen mit kleinerer Bitbreite auf und umgekehrt. Die Bitbreite kann in gleiche oder ungleiche Teile aufgeteilt werden, die Eingabe erfolgt über zwei aufeinander folgende Abfragemenüs.</p> <p>Beispiel mit gleichen Teilen: 4 -> 1 1 1 1: Abfrage 1: 4 Abfrage 2: 1 1 1 1 (Beachte Leerzeichen zwischen 1en!)</p>  <p>z. B. auch möglich: Ungleiche Teile: 4 -> 1 2 1</p>
AND-Gate (Gates)		Und-Gatter. Anzahl der Eingänge ist einstellbar.
OR-Gate (Gates)		Oder-Gatter. Anzahl der Eingänge ist einstellbar.
NOT-Gate (Gates)		Nicht-Gatter, Inverter.
XOR-Gate (Gates)		Entweder-Oder-Gatter (XOR), Anzahl der Eingänge ist einstellbar.

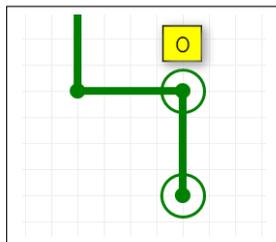
Tabelle 2.2 Alle benötigten Elemente für die Challenge

2.2.4 Keyboard Shortcuts

Keys	Actions
Copy	Ctrl + c
Cut	Ctrl + x
Paste	Ctrl + v
Undo	Ctrl + z
Redo	Ctrl + y
Delete Object(s)	Delete
Select All	Ctrl + a
De-select	ESC
Multi-object Select	Shift + Maus klicken und ziehen
Save Project Online	Ctrl + s
Open Project	Ctrl + o
Download Image	Ctrl + d
Make object looks Right	Arrow Right
Make object looks Left	Arrow Left
Make object looks Up	Arrow Up
Make object looks Down	Arrow Down
Open Documentation Page	F1

Tabelle 2.3 Hilfreiche Keyboard Shortcuts

Leitungen bewegen: Alle gewünschten Knoten müssen ausgewählt werden, z. B. mit Multi-object Select (Shift + Maus klicken und ziehen):



Fehlermeldungen verstehen: <https://docs.circuitverse.org/#/chapter8/3cverrormessages>

Weitere Dokumentation: <https://docs.circuitverse.org>