

## Versuch 3: Simulation eines Volladdierers und eines Code-Umsetzers

In diesem Versuch wird mit Logisim-Evolution die Schaltung eines Volladdierers aufgebaut. Anschließend wird ein System erstellt, welches eine 4-Bit-Dualzahl als Hexadezimalzahl auf einer 7-Segmentanzeige darstellen kann.

### Vorbereitung (Vor dem Praktikum!)

- Wiederholen Sie den Abschnitt der Vorlesung über Addierer.
- Wiederholen Sie die Abschnitte der Vorlesung über Code-Umsetzer.
- In der Vorlesung wurden folgende Gleichungen für die Ausgänge  $s_0$  und  $s_1$  eines Volladdierers mit den Eingängen  $a$ ,  $b$  und  $c$  vorgestellt:

$$s_0 = (\bar{a} \wedge b \wedge \bar{c}) \vee (a \wedge \bar{b} \wedge \bar{c}) \vee (a \wedge b \wedge c) \vee (\bar{a} \wedge \bar{b} \wedge c) = (a \oplus b) \oplus c$$

$$s_1 = (a \wedge b) \vee (a \wedge c) \vee (b \wedge c)$$

Wandeln Sie diese Gleichungen so um, dass Sie mit Schaltnetzen realisiert werden können, welche (neben den Ein- und Ausgängen) ausschließlich aus NAND- und XOR-Gattern bestehen.

- Vervollständigen Sie die Tabelle im Anhang mit den Codierungen für die 7-Segmentanzeige in den Darstellungen zur Basis 2 und 16.

### Aufgabe 1: Simulation eines Volladdierers

- a) Erzeugen Sie eine neue Logisim-Schaltung mit dem Namen **VA1**. Bauen Sie darin die Schaltung des Volladdierers auf. **Verwenden Sie dabei nur NAND- und XOR-Gatter**. Die dafür benötigten Gleichungen haben Sie bereits in der Vorbereitung erstellt. Wenn Sie die Zahl der Eingänge eines Gatters ändern müssen, wählen Sie es in der Schaltung durch Anklicken aus und ändern den Wert „Anzahl der Eingänge“ im Properties-Fenster links unten. Vergleichen Sie für alle möglichen Eingangskombinationen der Eingänge  $a$ ,  $b$  und  $c$  die Werte der Ausgänge  $s_0$  und  $s_1$  mit den von Ihnen erwarteten Werten. Speichern Sie das Projekt unter dem Dateinamen **Aufgabe3.circ** ab.

**In der Ausarbeitung:** Fügen Sie die Herleitung der umgeformten Gleichungen sowie den Schaltplan (mit Ihren Namen und dem Datum der Versuchsdurchführung) in Ihre Ausarbeitung ein.

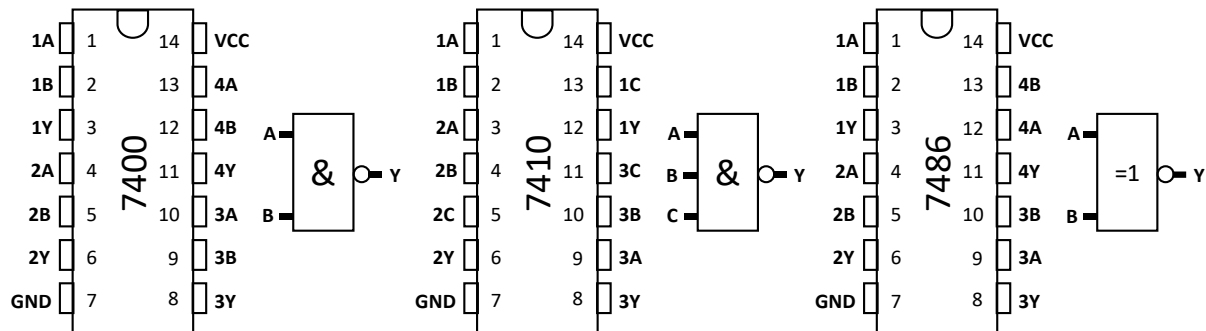
- b) In Versuch 4 soll der Volladdierer mit folgenden Bausteinen aus der TTL-Schaltkreisfamilie realisiert werden:
- **7400:** Jeder Baustein enthält 4 NAND-Gatter mit jeweils zwei Eingängen.
  - **7410:** Jeder Baustein enthält 3 NAND-Gatter mit jeweils drei Eingängen.
  - **7486:** Jeder Baustein enthält 4 XOR-Gatter mit jeweils zwei Eingängen.

Erzeugen Sie eine neue Logisim-Schaltung mit dem Namen **VA2**. Bauen Sie darin die Schaltung des Volladdierers erneut auf, nun unter Verwendung dieser Bausteine (Rubrik: TTL). Sie können über die Attributtabelle unten links einstellen, dass die interne Struktur der Bausteine sichtbar ist. Setzen Sie dazu den Wert für das Attribut „ShowInternalStructure“ auf „Ja“

Testen Sie die Funktion der Schaltung.

**In der Ausarbeitung:** Fügen den Schaltplan (mit Ihren Namen und dem Datum der Versuchsdurchführung) in Ihre Ausarbeitung ein.

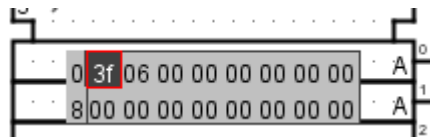
**Bitte bringen Sie die Logisim-Datei zu Versuch 4 in das Labor mit. Sie können dazu zum Beispiel die hochschuleigene Cloud „Netcase“ oder einen USB-Stick verwenden. Sie werden die Schaltung dann mit realen Logikbausteinen aufbauen.**



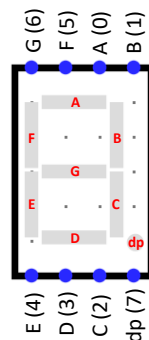
## Aufgabe 2: Code-Umsetzer

Mit Hilfe eines ROMs soll für 4-Bit-Dualzahlen der Code für die Darstellung auf einer 7-Segmentanzeige ermittelt werden.

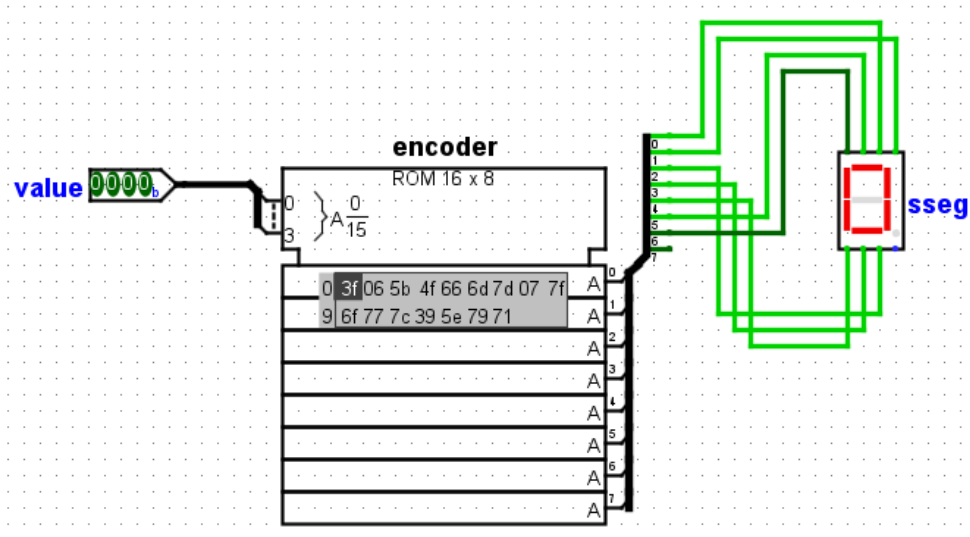
- a) Erzeugen Sie eine neue Logisim-Schaltung mit dem Namen **Anzeige**. Die von Ihnen als Vorbereitung in der Tabelle eingetragenen Codierungen sollen in einem ROM abgelegt werden. Instanzieren Sie also ein ROM (ROM, Rubrik: Speicher). Stellen Sie Namen für das ROM den Namen „encoder“ ein. Setzen Sie in der Attributtabelle die „Bitbreite der Adresse“ auf „4“. Für die „Bitbreite der Daten“ wählen Sie „7“. Zum Editieren der Werte verwenden Sie das Hand-Werkzeug:



- b) Zur Eingabe von Werten in das System werden Eingangs-Pins verwendet. Ändern Sie nach dem Einfügen eines Eingangs-Pins dessen Attribut „Datenbits“ auf „4“ und das Attribut „Basis“ auf „Hexadezimal“. Als Beschriftung geben Sie „value“ ein. Nun können die Eingangs-Pins mit dem Adresseingang des ROMs verbunden werden.
- c) Als Ausgabekomponente instanzieren Sie eine „7-Segmentanzeige“ (7seg, Rubrik: Eingabe/Ausgabe). Als Beschriftung geben Sie „sseg“ ein. Damit die Beschriftung sichtbar ist, muss das Attribut „Label Visible“ auf „Ja“ gesetzt werden.
- d) Der Datenausgang des Speichers ist ein Signalvektor. Dieser Signalvektor kann über einen „Verteiler“ (JF, Rubrik: Verdrahtung) in einzelne Signale aufgefächert werden. Stellen Sie die Attribute „Ausfächerung“ und „Bitbreite“ beide auf „7“. Das bedeutet, dass der Signalvektor der Bitbreite 7 in 7 einzelne Signale aufgefächert werden soll. Verbinden Sie nun die einzelnen Leitungen mit den Eingangs-Pins der 7-Segmentanzeige. Die Belegung entnehmen Sie der folgenden Abbildung:



e) Nun könnte die Schaltung etwa so aussehen:



















Prüfen Sie, ob für alle möglichen Eingaben das korrekte Muster auf der 7-Segmentanzeige erscheint. Erneut verwenden Sie hierzu das Hand-Werkzeug.

**In der Ausarbeitung:** Fügen Sie das Deckblatt mit der ausgefüllten Tabelle sowie den Schaltplan (wieder mit Ihren Namen und dem Datum der Versuchsdurchführung) in Ihre Ausarbeitung ein.

**Anhang: Deckblatt für die Ausarbeitung zu Versuch 3**

Teilnehmer	Gruppe Nr.:

**Codetabelle Dual-nach-7-Segment**

Eingabe (Basis 16)	Anzeige	Code (Basis 2) GFE DCBA	Code (Basis 16)
0		011 1111	3f
1		000 0110	06
2			
3			
4			
5			
6			
7			
8			
9			
A		111 0111	
B		111 1100	
C		011 1001	
D		101 1110	
E		111 1001	
F		111 0001	