



# Binäre Darstellung von Zahlen

Labor Digitales Design

## Inhalt

1 Ziel .....	1
2 Schaltung .....	2
3 Sinuswellengenerator .....	3
3.1 Sinustabelle .....	3
3.2 Simulation .....	4
4 Operationen .....	5
4.1 Inverter .....	5
4.2 Addierer .....	6
4.3 Multiplizierer .....	7
4.4 Anfügung .....	8
4.5 Verifikation .....	8
5 Checkout .....	10
Glossar .....	11

## 1 | Ziel

Dieses Labor dient dazu, die binäre Darstellung von Zahlen besser zu verstehen. Dabei werden verschiedene Operatoren analysiert und ihr Verhalten anhand von Simulationen getestet.

Die betrachteten Zahlendarstellungen sind: Binär (sowohl unvorzeichenbehaftet als auch im Zweierkomplement), Hexadezimal und Dezimal.

Folgende Operationen werden analysiert : Invertierung, Addition, Multiplikation, Verknüpfung von Zahlen.



## 2 | Schaltung

Die Schaltung (Abbildung 1), welche in diesem Labor angewandt wird, enthält ein Signalgenerator, welcher aus einem Zähler welcher das **5bit** Signal **Phase** erzeugt. Dieser zählt kontinuierlich und dient eine Sinuswelle über eine Look-up Tabelle zu erzeugen. Der Sinus hat eine Auflösung von **8bit** und heisst Sinus.

Diese Signal wird and 4 verschiedene Blöcke weitergeleitet welche danach die verschiedenen Operationen Invertierung (**inv[7:0]**), Addition (**sum[7:0]**), Multiplikation (**prod[15:0]**) sowie Konkatenerierung (**concat[15:0]**) durchführen.



Bis auf den Sinusgenerator sind alle Blöcke bereits implementiert und werden zur Verifizierung Ihrer Analyse verwendet.

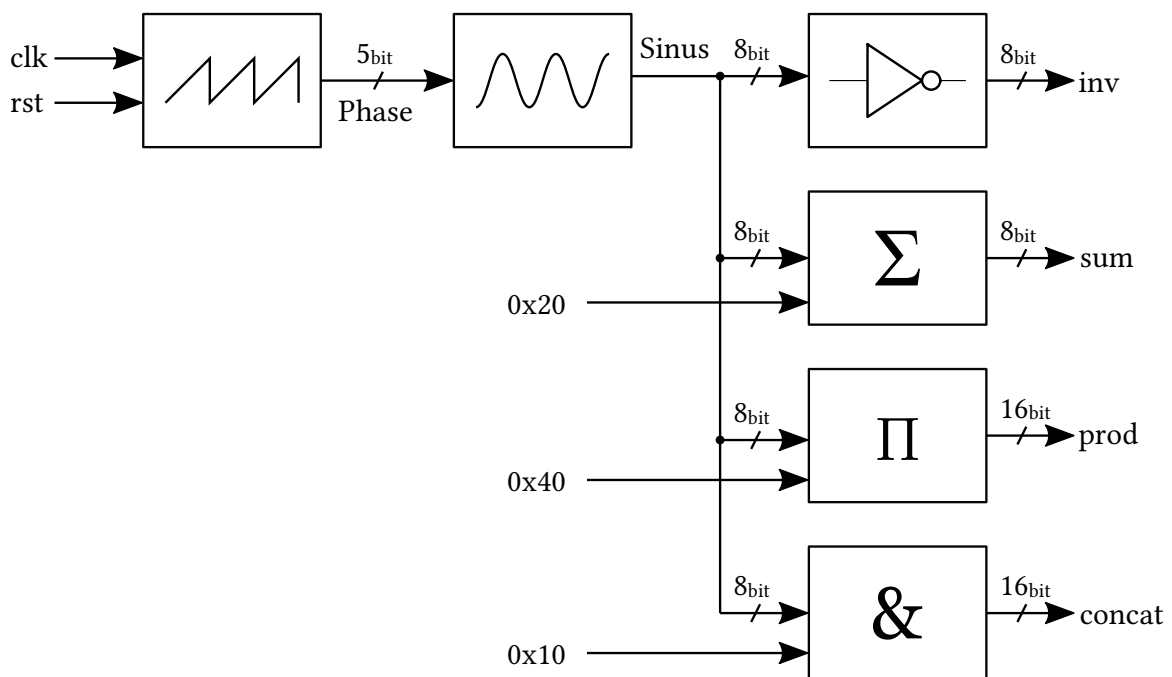


Abbildung 1 - Testschaltung zur Operatorenanalyse



### 3 Sinuswellengenerator

Damit die Schaltung funktioniert, muss ein Sinuswellengenerator implementiert werden. Dieser wird mithilfe eines bestehenden Zählers und einer Sinustabelle realisiert.

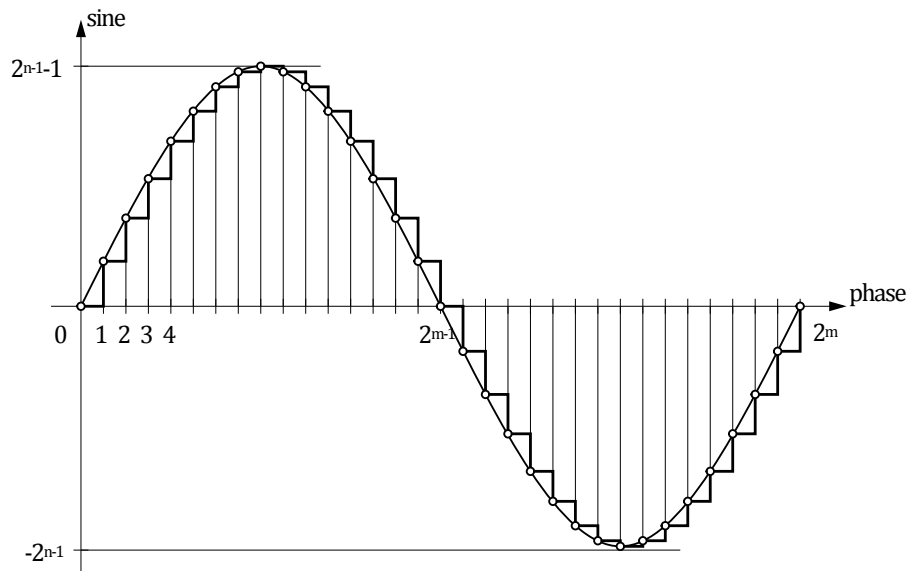


Abbildung 2 - Korrelation zwischen Phase und Sinus

#### 3.1 Sinustabelle

Die Sinustabelle ist eine Look-up-Tabelle, welche den entsprechenden Sinuswert (**sine[7:0]**) für jeden Wert der Phase (**phase[4:0]**) bereitstellt (Abbildung 2). Die Phase des Zählers wird als unvorzeichenbehaftete 5-Bit-Zahl betrachtet und variiert zwischen 0 und 31. Der Sinuswert ist eine vorzeichenbehaftete 8-Bit-Zahl und variiert zwischen  $-127$  und  $+127$ , und ist im 2er-Komplement codiert.

Im Block **sinewaveGenerator** ist ein [Very Highspeed Integrated Circuit Hardware Description Language \(VHDL\)](#)-Code vorgegeben. Für jeden Wert der Phase muss ein Sinuswert ergänzt werden.

```

1  ARCHITECTURE studentVersion OF sinewaveGenerator IS
2  BEGIN
3
4      table : process(phase)
5      begin
6          case to_integer(phase) is
7              when 0 => sine <= to_signed(16#00#,8);
8              when 1 => sine <= to_signed(16#00#,8);
9              ...

```

Listing 1 - Auszug aus dem VHDL-Code des Sinusgenerators

Die Code Linie 7 kann folgendermassen verstanden werden:

Falls **phase** den Wert 0 (when 0 =>) hat, wird der Sinuswert (**sine <=**) auf einen 8-Bit 2er-Komplement Wert gesetzt (**to\_signed(...,8)**), der Wert wird in Hexadezimaler form (**16#00#**) dargestellt. Dies bedeutet dass im Fall der Linie 7 der Sinuswert für die Phase 0 auf 0 gesetzt wird.



Erstellen Sie eine Excel Tabelle und berechnen Sie den Sinuswert für jeden Wert der Phase (**0..=31**).



Ergänzen Sie den **VHDL**-Code der Tabelle, welche den Sinus aus der Phase des Zählers erstellt.

### 3.2 Simulation

Führen Sie eine Simulation aus, um das richtige Verhalten des Funktionsgenerators zu überprüfen. Sie Testbank heisst **sinewave\_tb**. Die Simulation sollte für jeden Wert der Phase den korrekten Sinuswert ausgeben.



Simulieren Sie die Testbench **NUM\_test/sinewave\_tb** mit der Simulationsdatei **\$SIMULATION\_DIR/NUM1.do**.



## 4 Operationen

In diesem Abschnitt werden die verschiedenen Operationen, Invertierung, Addition, Multiplikation sowie Konkatenierung, analysiert (Abbildung 1).

In den folgenden Abschnitten müssen Sie die Ausgangskurve der verschiedenen Operationen skizzieren. Die untenstehenden Abbildungen können Sie benutzen um Ihre Kurven zu skizzieren (Abbildung 3, Abbildung 4, Abbildung 5 und Abbildung 7). Die Kurven müssen die charakteristischen Punkte enthalten, welche die Operationen auf das Signal anwenden.

### 4.1 Inverter

Der Inverter invertiert das Signal **sinus[7:0]** in **inv[7:0]**. Die Invertierung wird durch die Bitweise Negation des Signals durchgeführt.



Skizzieren Sie in der folgenden Abbildung 3 das zeitliche Verhalten des Ausgangssignals (**inv[7:0]**) des Blocks.

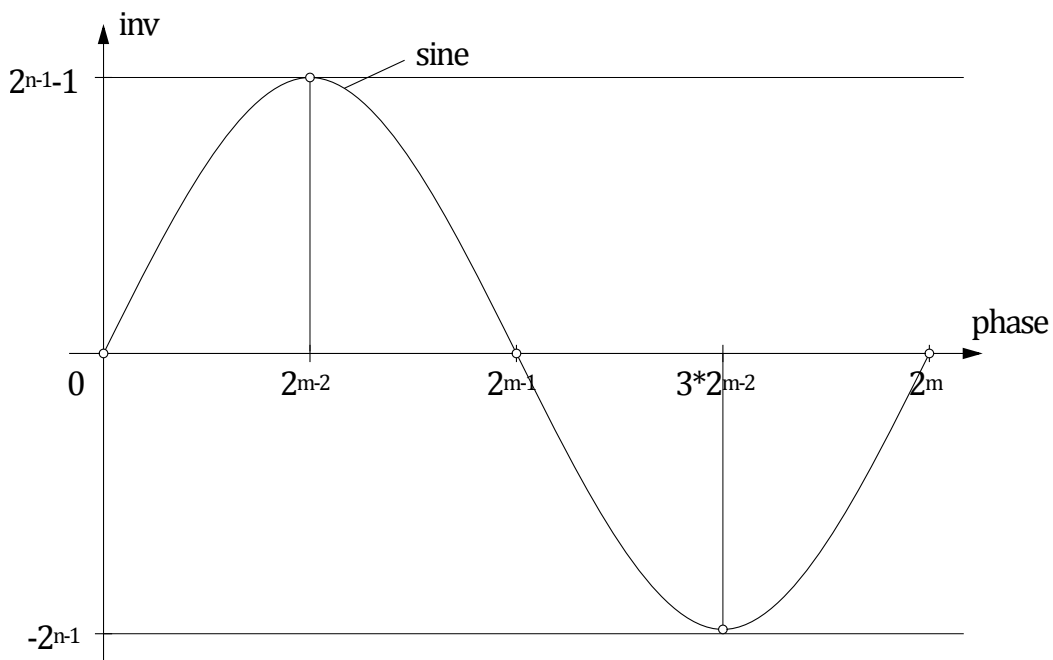


Abbildung 3 - Inverter



## 4.2 Addierer

Der Addierer addiert die Konstante  $20_h = 32_{10}$  zum Signal **sinus[7:0]** und gibt das Ergebnis im Signal **sum[7:0]** aus.



Skizzieren Sie in der folgenden Abbildung 4 das zeitliche Verhalten des Ausgangssignals (**sum[7:0]**) des Blocks.

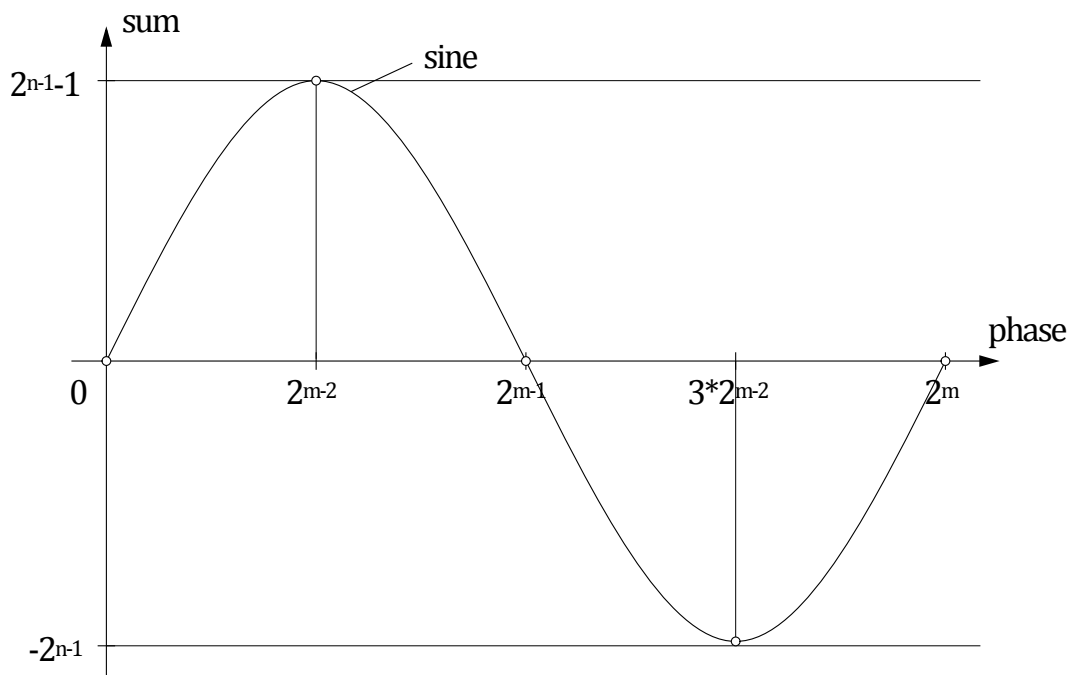


Abbildung 4 - Addierer



### 4.3 Multiplizierer

Der Multiplizierer multipliziert das Signal **sinus[7:0]** mit der Konstante  $40_h = 64_{10}$  und gibt das Ergebnis im Signal **prod[15:0]** aus. Beachten Sie dass das Ergebnis ein 16-Bit Signal ist.



Skizzieren Sie in der folgenden Abbildung 5 das zeitliche Verhalten des Ausgangssignals (**prod[15:0]**) des Blocks.

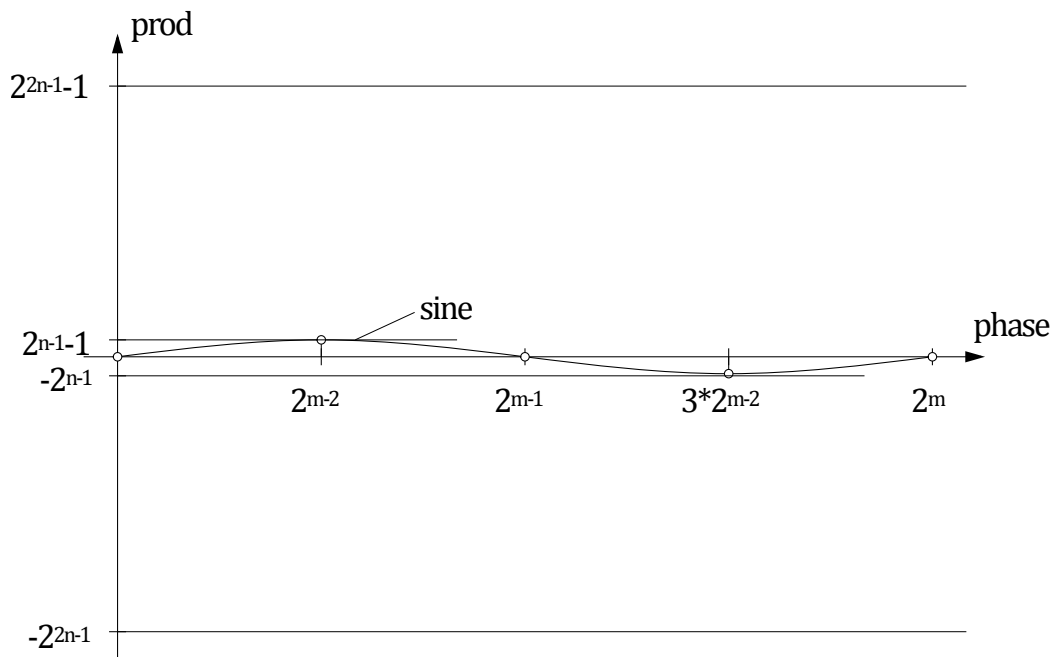


Abbildung 5 - Multiplizierer



## 4.4 Anfügung

Der Anfüger fügt am Ende des Signals die Anfügungskonstante ( $10_h$ ) an. Das Ergebnis wird im Signal **concat[15:0]** ausgegeben. Beachten Sie dass das Ergebnis ein 16-Bit Signal ist.

Eine Anfügung ist eine Operation, die zwei Signale aneinander verbindet. Im folgenden Fall werden 2 Signale von jeweils 8-Bit zu einem 16-Bit Signal zusammengefügt (Abbildung 6).

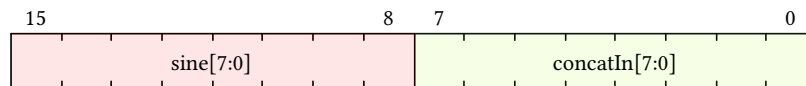


Abbildung 6 - Beispiel einer Anfügung



Skizzieren Sie in der folgenden Abbildung 7 das zeitliche Verhalten des Ausgangssignals (**concat[15:0]**) des Blocks.

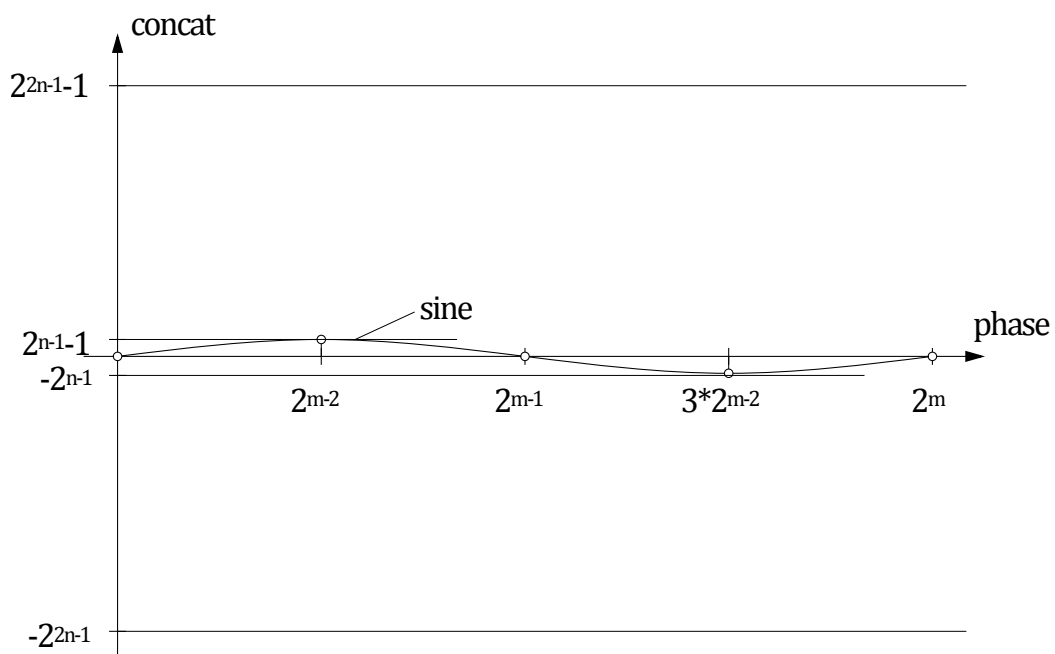


Abbildung 7 - Anfügung

## 4.5 Verifikation

Führen Sie eine Simulation durch, um die korrekte Funktion der analysierten Operatoren zu überprüfen.





Für jeden Operator finden Sie die entsprechende mathematische Gleichung. Verwenden Sie diese Informationen, um die korrekte Funktion der analysierten Operatoren zu validieren.

Simulieren Sie die Testbench **NUM\_test/sinewave\_tb** mit der Simulationsdatei **\$SIMULATION\_DIR/NUM2.do**.



## 5 | Checkout

Bevor Sie das Labor verlassen, stellen Sie sicher, dass Sie die folgenden Aufgaben erledigt haben:

- ☐ Schaltungsentwurf abgeschlossen
  - ☐ Überprüfen Sie, ob Ihr Sinusgenerator eine gültige Sinuswelle erzeugt
- ☐ Operationen
  - ☐ Für jede Operation eine Zeichnung erstellen, die die wichtigsten Punkte der Kurve enthält
  - ☐ Jede Operation mit einer mathematischen Formel darstellen
  - ☐ Überprüfen Sie Ihre Zeichnungen anhand der Simulationsergebnisse
- ☐ Dokumentation und Projektdateien
  - ☐ Stellen Sie sicher, dass alle Schritte (Entwurf, Konvertierungen, Simulationen) in Ihrem Laborbericht gut dokumentiert sind.
  - ☐ Speichern Sie das Projekt auf einem USB-Stick oder dem gemeinsamen Netzlaufwerk (`\\filer01.hevs.ch`).
  - ☐ Teilen Sie Dateien mit Ihrem Laborpartner, um die Arbeitskontinuität sicherzustellen.



# Glossar

**VHDL** – Very Highspeed Integrated Circuit Hardware Description Language [3](#), [4](#)