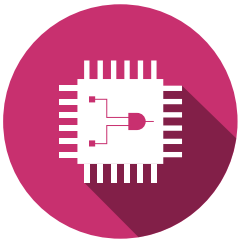


Digital-Analog-Wandler

Inhaltsverzeichnis

1 Einreichen

In diesem Labor werden wir das Schreiben des VHDL-Codes von kombinatorischen Blöcken und synchronen sequentiellen Blöcken durch die Realisierung eines digitalen Funktionsgenerators veranschaulichen. Alle diese Signale werden als 16-Bit-Zahlen codiert. Der Generator liefert :

- ein sägezahnartiges Signal (**sawtooth**)
- ein Rechtecksignal (**square**)
- ein dreieckiges Signal (**triangle**)
- ein sinusförmiges Signal (**sine**)

2 Start

Das Schema befindet sich in der Bibliothek **digitalToAnalogConverter**. Der Teststand ist unter **digitalToAnalogConverter_test**.

*Zur Erinnerung: Das Modellierungsprogramm muss über die Datei **digitalToAnalogConverter.bat** gestartet werden.*

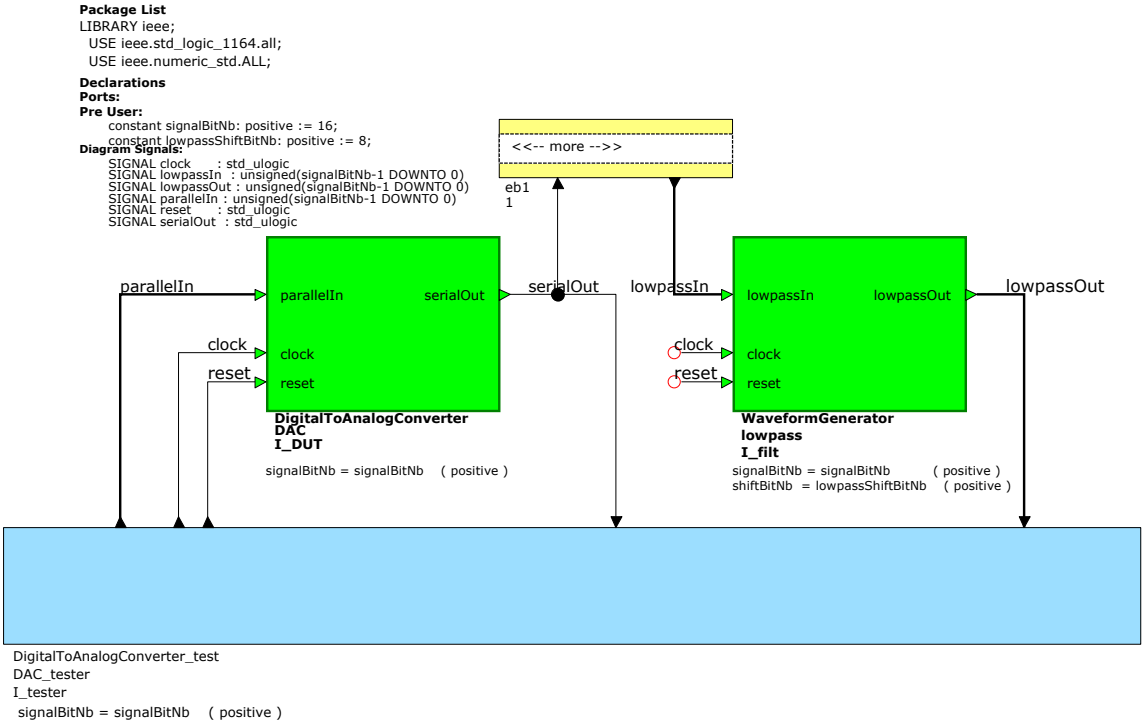


Abbildung 1: Digital-Analog-Wandler

3 Ziele

3.1 Modulator erster Ordnung

Der SD-Modulator empfängt ein digitales Signal und wandelt es in ein 1-Bit-codiertes Signal mit der Form einer Impulsfolge um.

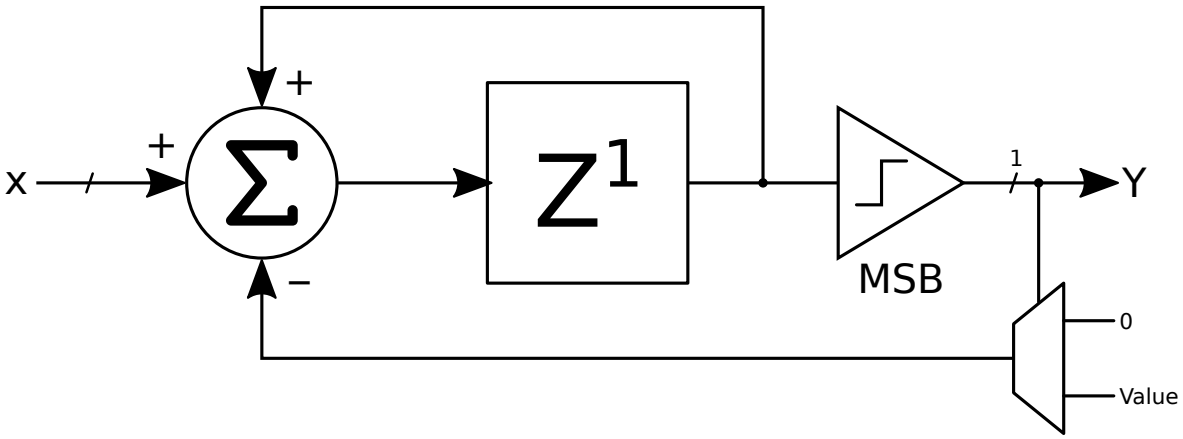
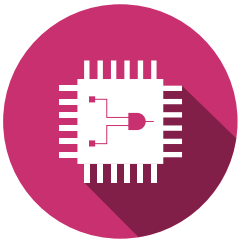


Abbildung 2: Modulator erster Ordnung

Der SD-Modulator erster Ordnung kann in der Analogie der Messung des Durchflusses eines



Wasserlaufs verstanden werden:

- Der Wasserlauf wird in ein Becken geleitet, das er füllen wird.
- Sobald der Wasserstand im Becken eine Referenzlinie überschreitet, nimmt ein Bediener einen Eimer mit Wasser aus dem Becken.
- Die durchschnittliche Fließgeschwindigkeit des Flusses ist gleich dem Inhalt des Eimers mal der Anzahl der pro Zeiteinheit abgeschöpften Eimer.

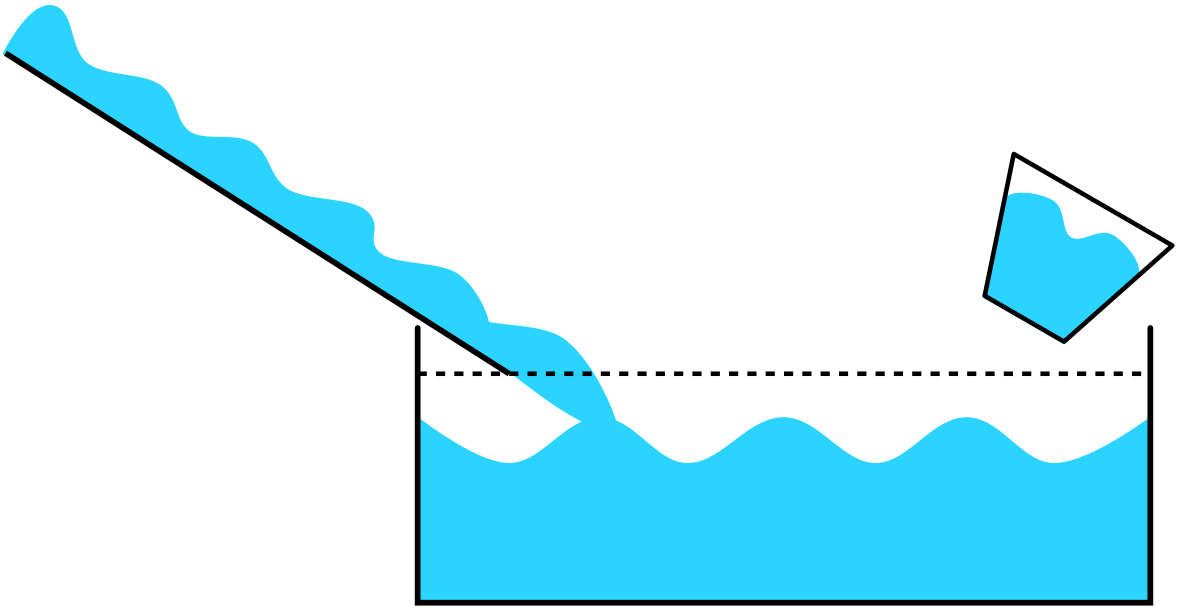


Abbildung 3: Analogie durch Wasser

Der Modulator wird wie folgt realisiert:

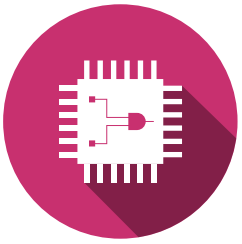
- In jeder Taktperiode wird der neue Wert des Signals zu einem Akkumulator addiert.
- Das höchstwertige Bit des Ergebnisses ist das Ausgangssignal des Modulators (Impuls oder nicht).
- Wenn dieses höchstwertige Bit auf '1' gesetzt ist, wird ein Wert von 2n vom Akkumulator abgezogen, wobei n die Anzahl der Bits des umzuwandelnden Signals ist.

Der interne Akkumulator benötigt mehr Bits als das Eingangssignal. Planen Sie 4 zusätzliche Bits ein und die Simulation wird zeigen, wie viele tatsächlich benötigt werden.

3.1.1 Todo

Dieser Block enthält einen generischen Parameter **bitNb**. Dieser Parameter wird sowohl für die Entität als auch für die zu schreibende Architektur festgelegt.

1. Schreibe die VHDL-Architektur des SD-Modulators.
2. Kompilieren und simulieren Sie den Block **DAC**$_t$*b.FindenSieeineMglichkeit, diesenMittelwertzusc*
3. Finden Sie eine Möglichkeit, diesen Mittelwert zu schätzen, um das parallele Eingangssignal und die Impulsfolge am Ausgang zu vergleichen. *Die analoge Wiedergabe des Signals, das lange auf '1' oder '0' bleibt, ist heikel.*



4. Fügen Sie dem Eingangssignal eine Verstärkung von 1/2 und einen Offset hinzu, um dieses Problem durch eine Verringerung des Signalbereichs zu vermeiden. Wiederholen Sie die Simulation mit dieser Änderung.
5. Betrachten Sie den zeitlichen Ablauf des Inhalts des Akkumulators und bestimmen Sie die erforderliche Anzahl von Bits.
6. Reduzieren Sie die Anzahl der Bits des Akkumulators und starten Sie die Simulation erneut.

3.2 Modulator zweiter Ordnung

Der Modulator zweiter Ordnung besteht aus zwei Akkumulatoren.

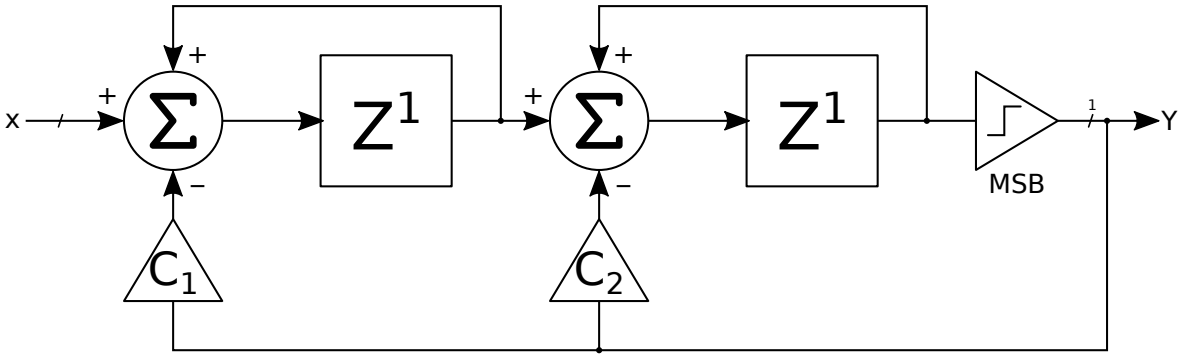


Abbildung 4: Modulateur de deuxième ordre

Für diese Schaltung werden die Zahlen als vorzeichenbehaftet betrachtet. Wenn der Wert des zweiten Akkumulators positiv ist:

- Die Ausgabe hat den Wert '1'.
- Ein Wert von **c1 = 2nBits - 1** wird vom ersten Akkumulator abgezogen.
- Ein Wert von **c2 = 2nBits + 3** wird vom zweiten Akkumulator abgezogen.

Wenn der Wert des zweiten Akkumulators negativ ist:

- Die Ausgabe hat den Wert '0'.
- Ein Wert von **c1 = 2nBits - 1** wird dem ersten Akkumulator hinzugefügt.
- Ein Wert von **c2 = 2nBits + 3** wird dem zweiten Akkumulator hinzugefügt.

Der Wert **nBits** ist die Anzahl der Bits des Eingangssignals des Modulators, **x**.

3.2.1 Todo

Planen Sie für jeden der Akkumulatoren 8 Bits mehr als das Eingangssignal. Invertieren Sie das höchstwertige Bit des Eingangssignals und wenden Sie eine Verstärkung von 7/8 an.

- Schreibe eine zweite VHDL-Architektur für den SD-Modulator.

- Kompilieren und simulieren Sie den Block **DAC**$_tb.berprfenSiedieFormdesAusgangssignals.BestimmenS$