



Aufgabe A3 Entzerrung eines Audiosignals / Nichtlineare Transformationskennlinie

Hintergrund:

Als Vorbereitung nachfolgender Labore soll hier ein Prozessorelement entwickelt werden, das eine Verzerrung von Audiosignalen korrigiert. *Das in einer späteren Aufgabe zu verarbeitende Audiosignal ist digitalisiert mit einer Abtaste von 44100 Hz.*

Ein Störmodell eines möglichen Übertragungseffektes wird nachgebildet durch eine bewusste Verzerrung des Audiosignals mit einer quadratischen Funktion

$$y = x^2$$

Die entsprechende Entzerrung muss mit der Umkehrfunktion (also der Wurzelfunktion $w = y^{1/2}$) realisiert werden. Wenden Sie dazu das in der Vorlesung besprochene Verfahren an. Das Argument wird in Q11 übergeben und befindet sich konsequenter Weise im Intervall $[-1, \dots, +1]$. Das Ergebnis soll ebenfalls in Q11 abgeliefert werden. Für die Argumente: "-1" und "0" soll die berechnete Wurzel präzise sein. Für die anderen Argumente soll die berechnete Wurzel gemäß dem in der Vorlesung besprochenen Verfahren und dem Q11-Format max. präzise sein.

Folgen Sie den gelernten Entwurfs-Richtlinien (siehe "DT" und "CE"-bisher, also z.B. kein Gated-Clock, keine Latches, usw.) Eine Minimierung der Resource-Kosten hat Priorität und es gibt nur einen Datenstrom.

"Ziel-Idee" für Ihre Lösung ist:

- **30 FFs**
- **1 Subtrahierer** (12 Bit)
- **1 Addierer** (4 Bit)
- viele LUTs ☹ oder alternative bis zu 2 Shifter
- Das Ergebnis für das übergebene Argument soll nach max. **14 Takten** vorliegen (gemessen ab REQ). Falls Sie Ihren Entwurf sauber! in Control- und Data-Path strukturieren, dann sind auch 38 Takte sowie 33 FFs akzeptabel (sofern diese zusätzlichen Kosten in sinnvoller Weise anfallen).

"Das FPGA" wird erst später (in A6) in Betrieb genommen. Gegenwärtig (bei A3) macht es keinen Sinn die Schaltung ins FPGA zu laden. Belegen Sie daher insbesondere auch mit einer Timingsimulation, dass Ihre Lösung das Geforderte leitet. Ein(e) intelligente(r) Testbench / Response Checker erleichtert Ihnen mit Sicherheit die Auswertung und hilft Ihnen Ihre eigene Zeit zu sparen.

Nicht nur Ihr VHDL-Code muss sich in einem wartbaren Zustand befinden.

Fertigen Sie ein ausführliches Protokoll an, das den Entwurf mit all seinen Komponenten, seinem Aufbau, seiner Funktion, seine HW-Kosten, seine Laufzeiten (in Takten sowie kritischer Pfad) und den Test dokumentiert.

Insbesondere:

- Rechtfertigen Sie Ihre Entwurfs-Idee.
- Rechtfertigen Sie Ihre HW-Kosten.
- Rechtfertigen Sie Ihr Timing / Ihre Laufzeiten.
- Rechtfertigen Sie Ihren Test (Woher wissen Sie, dass Ihre Lösung fehlerfrei ist?)

Überzeugen Sie das Management, dass Sie gute Arbeit geleistet haben ☺

Das "A3-VHDL-DUT" soll der folgenden Schnittstelle funktional genügen:

```
entity SquareRootComputer is
  port (
    fx      : out std_logic_vector( 11 downto 0 ); -- f(x) / result/"square root" coded in Q0.11
    rdy     : out std_logic;                       -- ReaDY / result of computation valid
    --
    x       : in  std_logic_vector( 11 downto 0 ); -- argument coded in Q0.11
    req     : in  std_logic;                       -- REQuest for computation / argument valid
    clk     : in  std_logic;                       -- CLocK (50MHz for your FPGA)
    sres    : in  std_logic;                       -- Synchronous RESet
  );
end entity SquareRootComputer;
```

Später (A6) wäre zumindest ein Wrapper zum Erfüllen zusätzlicher Anforderungen nötig. Z.B. wird später "der Datenbus" bidirektional genutzt werden.