

AUFGABE 2: IMPLEMENTIERUNG FILTER

Ziel dieser Aufgabe ist es, ein besseres Verständnis für die Implementierung von Funktionalitäten für Signalverarbeitung wie Filterung und die Problematik von Fließkomma-Arithmetik zu entwickeln. Hierbei soll Wert auf einen modularen Softwareentwurf gelegt werden. Die Schnittstellen für Ein- und Ausgabe sowie deren Datentypen sollen eindeutig strukturiert werden.

Die Vorgabe dieser Aufgabe befindet sich im Ordner `02_filter` des Repositories <https://gitlab.cs.fau.de/ezs/vezs16-vorgabe.git>. Starten Sie die Anwendung mit `make sim` im Build-Verzeichnis.

Datentypen und Eingaben: 1.

Implementieren Sie eine Struktur vom Typen `struct input_t`, welche sowohl die Signalwerten als auch die Werte des Filters speichert. Die Eingabewerte des Signals und die Werte des Filters sollen anschließend mittels der globalen Arrays initialisiert werden. *Welche Entscheidungen bei der Definition der Eingabe kann die WCET-Analyse der entsprechenden Verarbeitungsfunktionen erleichtern bzw. erschweren?*

Vergleichen Sie die Signaturen der Funktionen `actuate()` mit einer Funktion der Signatur `process(struct input_t * input, struct output_t* output)`. Was fällt hierbei auf und was könnte verbessert werden?

Verarbeitung: 2.

Implementieren Sie nun eine Funktion `convolve()`, welche die Eingabedaten des Filters mit den Werten des Filters faltet. Diese Funktion soll mit den entsprechenden Parametern in `process()` aufgerufen werden. *Welche Überprüfungen der Eingabedaten müssen vorgenommen werden?*

mathworld.
wolfram.com/
Convolution.
html

Nach der Ausführung sollen nun die Daten verarbeitet werden. Hierfür soll eine Prüfsumme über die Ausgabedaten generiert werden. Implementieren Sie diese Funktionalität und behandeln sie den Fehlerfall. *Welches Verhalten eignet sich hier im Fehlerfall? Welche Prüfsumme ergibt sich?*

Q-Format: 3.

Erweitern Sie nun die Aufgabe so, dass für die Signalverarbeitung keine Fließkomma- sondern Festkomma-Datentypen verwendet werden. Die Eingabe muss dafür in eine neue Struktur konvertiert werden. Behalten Sie dabei die vorherige Implementierung. Benutzen Sie für das Auswählen der Implementierungsvariante Präprozessor Makros. Verwenden Sie für die arithmetischen Operationen die in der Angabe enthaltene Fixpunkt-Bibliothek. *Welche Prüfsumme ergibt sich nun für die Ausgabe? Unterscheidet sich der Wert von der Prüfsumme aus Teilaufgabe 2, wenn die Ausgabe-werte in das Fließkomma-Format zurück transformiert werden? Wenn ja, warum?*

Hinweise

- Bearbeitung: Gruppenarbeit
- Abgabezeit: 02.05. – 06.05.2016
- Fragen bitte an i4ezs@lists.cs.fau.de