abmodule

August 5, 2016

This document provides a mathematical description of the Faust program text stored in the `abmodule.dsp` file. See the notice in Section 3 (page 1) for details.

1 Mathematical definition of `process`

The *abmodule* program evaluates the signal transformer denoted by `process`, which is mathematically defined as follows:

1. Output signals y_i for $i \in [1, 4]$ such that

$$
\begin{aligned}
y_1(t) &= 0.5 \cdot (x_3(t) + s_1(t) + x_4(t)) \\
y_2(t) &= 0.5 \cdot ((x_4(t) + s_1(t)) - x_3(t)) \\
y_3(t) &= 0.5 \cdot (0 - ((x_2(t) + x_4(t)) - (x_1(t) + x_3(t)))) \\
y_4(t) &= 0.5 \cdot (0 - ((x_2(t) + x_3(t) + x_4(t)) - x_1(t)))
\end{aligned}
$$

2. Input signals x_i for $i \in [1, 4]$

3. Intermediate signal s_1 such that

$$s_1(t) = x_1(t) + x_2(t)$$

2 Block diagram of `process`

The block diagram of `process` is shown on Figure 1 (page 3).

3 Notice

- This document was generated using Faust version 2.0.a41 on August 05, 2016.
- The value of a Faust program is the result of applying the signal transformer denoted by the expression to which the `process` identifier is bound to input signals, running at the f_S sampling frequency.

- Faust (*Functional Audio Stream*) is a functional programming language designed for synchronous real-time signal processing and synthesis applications. A Faust program is a set of bindings of identifiers to expressions that denote signal transformers. A signal s in S is a function mapping[1] times $t \in \mathbb{Z}$ to values $s(t) \in \mathbb{R}$, while a signal transformer is a function from S^n to S^m, where $n, m \in \mathbb{N}$. See the Faust manual for additional information (http://faust.grame.fr).

- Every mathematical formula derived from a Faust expression is assumed, in this document, to having been normalized (in an implementation-dependent manner) by the Faust compiler.

- A block diagram is a graphical representation of the Faust binding of an identifier I to an expression E; each graph is put in a box labeled by I. Subexpressions of E are recursively displayed as long as the whole picture fits in one page.

- The `abmodule-mdoc/` directory may also include the following subdirectories:

  - `cpp/` for Faust compiled code;
  - `pdf/` which contains this document;
  - `src/` for all Faust sources used (even libraries);
  - `svg/` for block diagrams, encoded using the Scalable Vector Graphics format (http://www.w3.org/Graphics/SVG/);
  - `tex/` for the LaTeX source of this document.

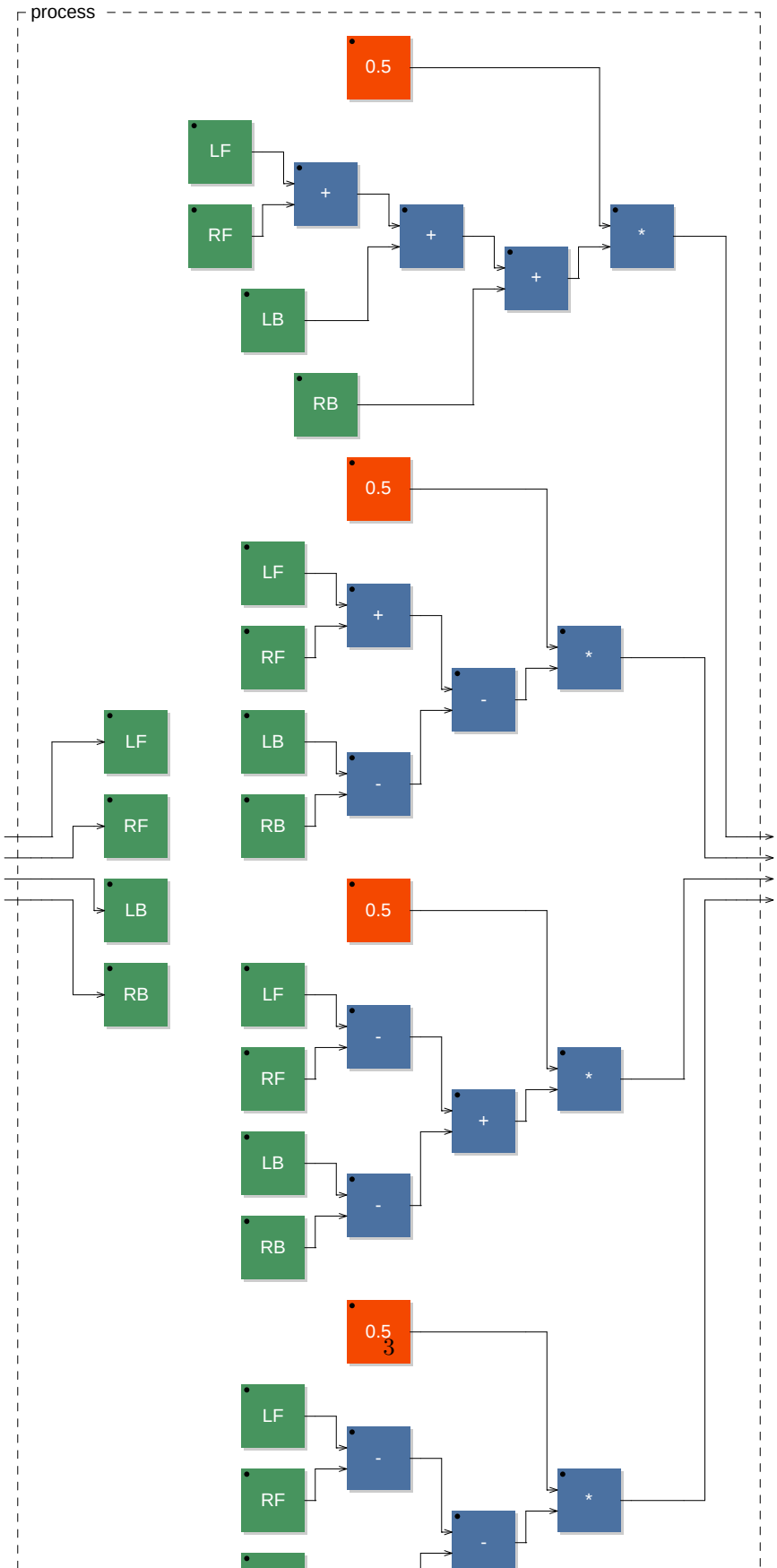
4 Faust code listing

This section provides the listing of the Faust code used to generate this document.

Listing 1: `abmodule.dsp`

```
1  //-----------------------------------------------
2  // AB MODULE
3  //-----------------------------------------------
4
5  // for LF RF LB RB produce a 1th order WXYZ B-FORMAT
6
7  process(LF, RF, LB, RB) = (0.5 * ( LF + RF + LB + RB )), // W
8                            (0.5 * ((LF + RF) - (LB - RB))), // X
9                            (0.5 * ((LF - RF) + (LB - RB))), // Y
10                           (0.5 * ((LF - RF) - (LB + RB))); // Z
```

[1] Faust assumes that $\forall\, s \in S, \forall\, t \in \mathbb{Z}, s(t) = 0$ when $t < 0$.


process
0.5
LF
RF
+
+
+
*
LB
RB
0.5
LF
RF
+
*
-
LF
LB
RF
-
RB
LB
0.5
RB
LF
-
RF
*
+
LB
-
RB
0.5
LF
-
RF
*
-