

齋藤 貴志 (筑波大学大学院システム情報工学研究科)<sup>1</sup>平野 砂峰旅 (京都精華大学芸術学部デザイン学科)<sup>2</sup>丸山 勉、星野 力 (筑波大学機能工学系)<sup>3</sup>

## 1 はじめに

従来のハードウェアシンセサイザには多くの発音方式があり、それぞれ特徴を有している。しかし、それらすべての方式を同時に実現するには、特殊なソフトウェアシンセサイザを用いる以外に方法はなかった。

それは、ハードウェアシンセサイザでは内部構成が固定されていて、その構成を再編することができないためである。

一部のソフトウェアシンセサイザを使用すれば、内部構成を再編することは可能である。しかし、同時に多くの音色を出力させると時間遅れが大きく、音楽を実時間で演奏することは難しい。それらのシンセサイザをコンサートなど実時間の演奏に使用するには、内部で行なわれる計算を高速化することが必要である。

そこでFPGAを用いて、内部構成を自由に変えられるシンセサイザを、ハードウェア上で構築した。

## 2 シンセサイザ

### 2.1 シンセサイザの構成

多くのシンセサイザは、基本となる周波数の波を作る発振器、その波に対して様々な変化を加えるフィルタ、最終的な波の振幅を変化させるアンプなどからなる[3]。これは、アナログ、FM方式のシンセサイザに共通で、ウェーブテーブル方式の場合、発振器の代わりにメモリテーブルからの読み出しを行なう。

図1のように発振器で作られた波に対し、フィルタで加工する発振器とフィルタから成るユニットをいくつか使用し、最後に足し合わせ1つの音を出力する。その際、1つの発振器に対し、2つ程度のフィルタが用いられる。以下、発振器とフィルタからなるユニットを Generator Unit(GU) と略す。

このGUは、アナログ方式の場合2、3個、FM方式の場合6個程度で1つの音を出力できる。

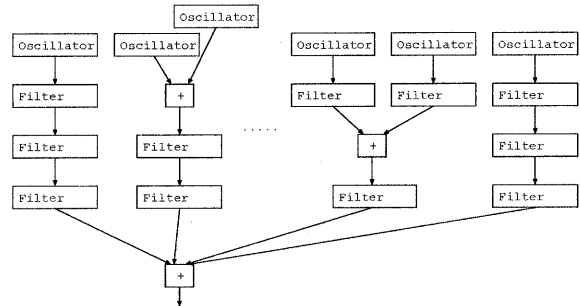


図1: シンセサイザの構成

### 2.2 フィルタ

図2に示したのはIIRフィルタである。

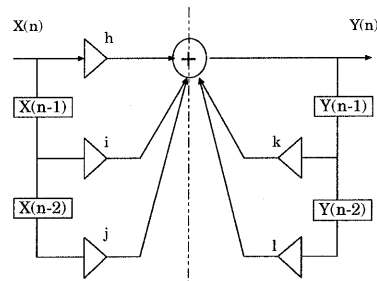


図2: 2次IIRフィルタ

入力は  $X(n)$  のみで、1clock 前の入力である  $X(n-1)$ 、2clock 前の入力である  $X(n-2)$ 、1clock 前の出力  $Y(n-1)$ 、2clock 前の出力  $Y(n-2)$  とを足し合わせることで、出力  $Y(n)$  が得られる。

このフィルタは左右対称になっているため、図2の点線の左側だけを作り、右側はその入力および係数を変えることでフィルタ1つを実現することができる。以下、このフィルタの半分のことをフィルタユニットと呼ぶ。

<sup>0</sup>A Music Synthesizer on FPGA

<sup>1</sup>Takashi Saitou Graduate School, Doctoral Program in System & Information Engineering, University of Tsukuba

<sup>2</sup>Saburo Hirano School of Design, Faculty of Art, University of Kyoto Seika

<sup>3</sup>Tsutomu Maruyama, Tsutomu Hoshino Institute of Engineering Mechanics and Systems, University of Tsukuba

### 3 ハードウェアシンセサイザの設計

#### 3.1 時分割

現在のオーディオ用 CD のサンプリング周波数は 44.1kHz である。この周波数で動作すれば十分良い音質が得られる。そこで本研究で作成するシンセサイザでも、CD と同じサンプリング周波数で発音させることとした。

乗算器の回路がサンプリング周波数の 1000 倍で動作しているならば、1つの乗算器でサンプリングの 1 周期の間に 1000 回の乗算ができることになる。その結果、1つの乗算器で 1000 個分の乗算器の役割を果たすことになる。そのうちの 3 個の乗算器でフィルタユニット 1つを実現できるので、実際には 1つの乗算器があれば、333 個のフィルタユニットを実現できることになる。

#### 3.2 フィルタユニット

シンセサイザを設計するうえで、一番重要な部分はフィルタユニットである。最大同時発音数及び動作速度を極力大きくするために、回路規模を小さくおさえる必要がある。

そこで、1つのフィルタユニットで 3つの発振器 ⇒ フィルタ ⇒ フィルタ という GU を計算する。

図 3のように、1つの GU だけを計算するとき空いてしまった A0 の計算後に、他の GU の発振器 B0、C0 の計算を入れる。このようにすると、計算結果を待つ必要がなくなり、C0 の計算終了直後に A1 の計算ができる。フィルタユニットが空いている時間も無くなるため、動作効率も良くなる。

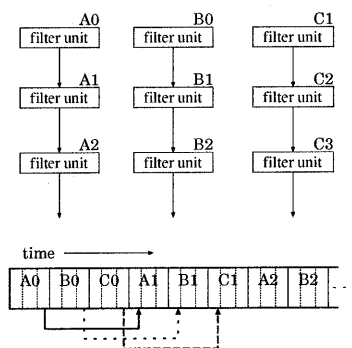


図 3: GU が 3つの場合

### 4 PureData

FPGA を制御するユーザインターフェイスとして、PureData を用いる。PureData では構築したシンセサイザをテキストファイル形式で出力できる。そのファイルを今回作成したハードウェア用に変換するプログラムを作成し、直接 FPGA にダウンロードすれば、PureData からハードウェアシンセサイザを作ることができるようになる (図 4)。

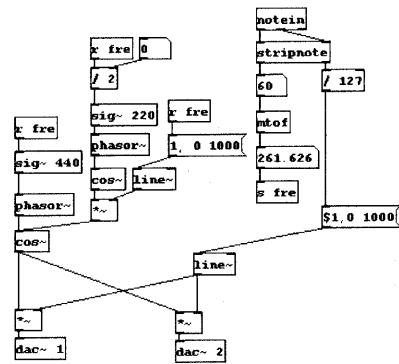


図 4: PureData

### 5 おわりに

本研究では FPGA を用いてハードウェアシンセサイザを構築した。さまざまなシンセサイザ方式を再現でき、発振器、フィルタなどの数も自由に設定できる柔軟性を持ったハードウェアシンセサイザが設計できた。

今回、シンセサイザの中心部であるフィルタユニットシステムを構築し、それを 1つの FPGA 上に 32 個搭載した。これは 36.4MHz で動作し、従来のハードウェアシンセサイザよりも同時発音数を多くできた。

今後、外部 RAM との通信、PC からのデータ転送などを実現し、実際に音を出力できるようにさらに研究を進めていきたい。

### 参考文献

- [1] Virtex 2.5V Field Programmable Gate Arrays, Preliminary Product Specification ver.1.3
- [2] インターフェース増刊 TECH I vol.2 「デジタル信号処理と DSP」
- [3] Perfect MIDI Book :リットーミュージック出版編集部編