Systolic array implementation of IIR decimators and interpolators

by H K Kwan, BSc, MPhil, PhD, DIC, MIHKIE, CEng, MIEE, MIERE, Sen MIEEE, Lecturer, Department of Electrical and Electronic Engineering, University of Hong Kong, and
T S Okullo-Oballa, BSc, Former Research Student, Department of Electrical and Electronic Engineering, University of Hong Kong.

Abstract

Two systolic arrays for the VLSI implementation of the decimation and interpolation structures advanced by Valenzuela and Constantinides are presented. Each of the resultant arrays consists of one type of basic cell together with an input-output cell. The arrays are characterized by nearest neighbour interconnections and high throughput rate.

Introduction

Recently, digital signal processing schemes for efficient decimation and interpolation have been advanced in [1]. However, the hardware structures given in [1] are not systolic and so not optimal for VLSI implementation. In this paper, the decimating and interpolating structures are restructured based on the systolic array for recursive digital filtering of [2] and the formulation of two types of input-output (I/O) cells. Each of these structures can be configured in the form of a systolic array consisting of one type of basic cell and the appropriate I/O cell. Finally, the two arrays for the VLSI implementation of the decimation and interpolation structures advanced by Valenzuela and Constantinides are presented. Each of the resultant arrays consists of one type of basic cell together with an input-output cell. The arrays are characterized by nearest neighbour interconnections and high throughput rate.

The decimation structure

The transfer function of the general decimating structure [1] shown in Fig 1a is

$$H(z^{-1}) = \sum_{i=0}^{N-1} H_i(z^{-N})$$

where the transfer function of the $i^{th}$ branch is a cascade of all-pass digital filters as given by

$$H_i(z^{-N}) = \prod_{k=1}^{2N-1} \frac{a_{ik} + z^{-1}}{1 + a_{ik} z^{-1}}$$

earlier version of this paper [3], has been modified so that it can be extended to any decimation or interpolation factor without any increase in the number of interconnections between the computing arrays and the I/O cells.

The decimation structure

The transfer function of the general decimating structure [1] shown in Fig 1a is

$$H(z^{-1}) = \sum_{i=0}^{N-1} H_i(z^{-N})$$

where the transfer function of the $i^{th}$ branch is a cascade of all-pass digital filters as given by

$$H_i(z^{-N}) = \prod_{k=1}^{2N-1} \frac{a_{ik} + z^{-1}}{1 + a_{ik} z^{-1}}$$

earlier version of this paper [3], has been modified so that it can be extended to any decimation or interpolation factor without any increase in the number of interconnections between the computing arrays and the I/O cells.
The problem here is two-fold: to obtain a systolic implementation of the all-pass cascade sections given in (2) as well as the overall transfer function of (1). In general, Fig. 1a can be modified as shown in Fig. 1b. For any N and Ki, the all-pass cascade sections are in powers of \( z^N \) and so can be implemented by a first-order all-pass section operating at the lower sampling rate. The basic cell presented in [2] is thus suitable for this purpose.

As an illustrative example, consider the case of N = 2 and Ki = 2, we have,

\[
H(z^2) = H_0(z^2) + z^{-2}H_2(z^2)
\]

and

\[
H_i(z^2) = \frac{a_i + z^{-2}}{1 + a_i z^{-2}} + \frac{a_2 + z^{-2}}{1 + a_2 z^{-2}}
\]

for \( i = 0 \) and 1. Figs 2a and 2b give respectively the basic cell definition and the operation of the resultant array of any one of the first-order basic sections of (4). The array for a single digital all-pass section together with its magnitude truncation operation, denoted by MT, needed for quantization is considered to be a basic independent block. The output additions have been partitioned to form a number of identical I/O cell type I as shown in Fig. 3. The switch of Fig 1b has been placed in the I/O cell where the input, p(n), controls the connection of x(n) to each of the I/O cells such that the connection changes from one I/O cell to another in a round-robin fashion. These modular features enable the array to be easily extended to the general systolic array of a decimator of N where Ki can be any value as shown in Fig 4.

### The Interpolating Structure

The transpose of the decimating structure in Fig 1a results in an interpolating structure and is shown in Fig 5a. Similarly, a modified version of the interpolating structure is shown in Fig 5b. In a manner similar to that of the decimating structure, an I/O cell type II can be constructed and hence a complete systolic array can be formed as shown in Fig 6 for N = 2 and Ki = 2. Fig 7 shows the general systolic array of an interpolator of N where Ki can be any value.

### Conclusions

Systolic implementations for efficient interpolation and decimation are presented in this paper. The basic cell developed can be easily extended to form a general systolic array of a decimator of N where Ki can be any value.
mation have been presented. The basic arrays are modular with nearest neighbour communications and so meet the basic requirements for VLSI implementation. The present systolic implementation provides an added advantage over the previous one in [1] in that it is completely systolic and so a higher processing rate will be achieved. In addition, it has also been shown that the arrays can be easily extended to form decimators and interpolators of any value of N or Ki.

Acknowledgment

T S Okullo-Oballa would like to acknowledge the financial support from the Hong Kong Government Education Department in the form of a Commonwealth Scholarship.

References

1. R A Valenzuela and A G Constantinides, "Digital signal processing schemes for efficient interpolation and decimation,"
