PIPELINING OF CORDIC BASED IIR DIGITAL FILTERS *

Jun P. Ma[1] *Keshab K. Parhi*[1] *Ed F. Deprettere*[2]

[1] Department of Electrical Engineering, University of Minnesota, Minneapolis, MN 55455, U.S.A.
[2] Department of Electrical Engineering, Delft University of Technology, 2628 CD Delft, The Netherlands
E-mail: {junma, parhi}@ee.umn.edu, ed@cas.et.tudelft.nl

ABSTRACT

Cordic based IIR digital filters possess desirable properties for VLSI implementation such as local connection, regularity, and good finite word-length behavior, but can't be pipelined to finer levels (such as bit or multi-bit levels) due to the presence of feedback loops. In this paper, a pipelining method for the cordic based IIR digital filters is proposed using the constrained filter design methods and the polyphase decomposition technique. Using this method, the filter sample rate can be increased to any desired level.

1. INTRODUCTION

In recent years, rapid advances in VLSI technology have had much impact on modern signal processing. Some of the desirable properties for VLSI realization are regularity, local connection and pipelinability. Orthogonal filters are digital IIR filters whose internal computation scheme consists of an orthogonal matrix transformation.

The normalized lattice filter [1] and the scaled normalized lattice filter [2] are orthogonal filters in the sense that the denominator of the transfer function is implemented using orthogonal building blocks based on the *Schur algorithm.* But since the implementation of the numerator of the transfer function in these filters requires the use of *readout* taps, this could result in numerical difficulties [3]. To overcome this problem, the orthogonal double rotation (ODR) lattice filters [3] are developed such that both the numerator and the denominator are implemented in terms of orthogonal sections. These filters possess good numerical properties, lead to low sensitivity in the filter passband, and eliminate limit cycle oscillations. The normalized and ODR lattice filters have recently been pipelined in [2].

The cordic based IIR digital filters [4] discussed here are true cascaded orthogonal filters. These filters consist of a cascade of 4–terminal orthogonal sections, with each section implemented by an efficient CORDIC algorithm [5]. The ODR lattice filters are not true cascaded orthogonal filters by this criteria, since they have 6–terminals for each cascaded section. Because of this, it is expected that the ODR lattice filters have much higher sensitivity in filter stop band compared to the cordic IIR filters [4]. Similar to the ODR lattice filters, both the numerator and the denominator in cordic digital filters are also implemented using orthogonal sections, and these filters have low sensitivity in both the pass band and the stop band, leading to good numerical properties over the entire frequency band. But these filters can't be pipelined at finer levels (such as bit or multi-bit levels) due to the presence of feedback loops. This greatly restricts its applications for high speed and low power. In this paper, a pipelining method for the cordic based IIR digital filters is proposed using the constrained filter design methods and the polyphase decomposition technique [2]. Using this method, the maximum filter sample rate can be increased to any desired level.

*THIS RESEARCH WAS SUPPORTED IN PARTS BY NSF UNDER GRANT NUMBER MIP-9258670.

2. SYNTHESIS SKETCH OF CORDIC BASED IIR DIGITAL FILTERS

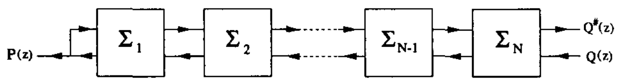

Figure 1. Cascade orthogonal filter structure

The cordic based IIR digital filters are developed for the realization of any stable, passive digital rational real transfer function in a cascaded interconnection of orthogonal sections. Fig. 1 represents a typical filter structure. $F(z) = P(z)/Q(z)$ is the desired Nth order transfer function. $Q^{\#}(z) = z^{-N}Q(z^{-1})$, is the reverse polynomial of $Q(z)$. Each box is an orthogonal section, which could be either a *degree–1 section* or a *degree–2 section.* Fig. 2 shows the structure of the degree–1 and the degree–2 sections, where the boxes in the figure are planar rotation building blocks shown in detail in Fig. 3. The dots go with the first row of the matrix. Here, $\{\phi_i\}$ are filter parameters, and can be determined through the synthesis routine. Notice that the whole filter consists of only the planar rotation blocks and storage elements. These rotation blocks can be implemented using an efficient CORDIC processor [5]. A degree–1 section implements one zero of the transfer function, and a degree–2 section implements a pair of complex conjugate zeros, since for real coefficient transfer functions, if a zero is complex, its complex conjugate must also be a zero.

The filter synthesis algorithm is based on a degree reduction procedure of the denominator polynomial of the transfer function. If the zero is real, we perform a degree–1 reduction, which generates a degree–1 section implementing the real zero, and results in a denominator polynomial with

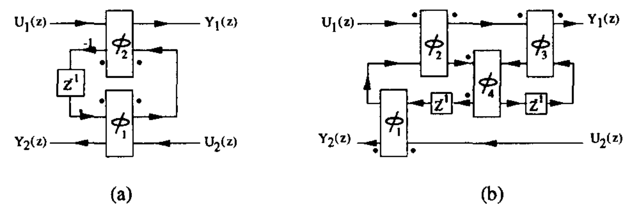

Figure 2. (a) Degree–1 section (b) Degree–2 section

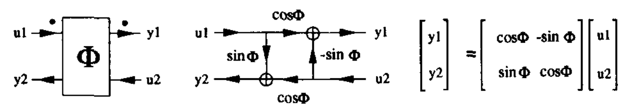

Figure 3. Planar rotation building block structure

degree reduced by one; if the zero is complex, we perform a degree–2 reduction, which generates a degree–2 section implementing the pair of complex conjugate zeros, and results in a denominator polynomial with degree reduced by two. This degree reduction procedure is followed until we implement all the zeros of the transfer function. This leads to the structure in Fig. 1. Notice the zero implementing order can be chosen at will. The filter synthesis procedure can be summarized as follows.

Let $F(z)$ be the desired rational transfer function with numerator $P(z)$ and denominator $Q(z)$. $P(z)$ and $Q(z)$ are polynomials of degree N with real coefficients. Let $\Omega = \{\omega_i, i = 1, N\}$ be the collection of all zeros of $P(z)$. Notice the number of real zeros and pairs of complex conjugate zeros in Ω will correspond to the number of degree–1 and degree–2 sections in the synthesized filter structure.

Algorithm
INPUT: $Q(z), \Omega$
OUTPUT: Filter structure with parameter set Φ
while (Ω is not empty) do{
  pick $\omega_i \in \Omega$
  if (ω_i is real)
   then { degree–1 reduction{
      input: $Q_N(z), \omega_i$
      output: $Q_{N-1}(z), \{\phi_1, \phi_2\}\}$
     $Q_N(z) = Q_{N-1}(z)$, $\Omega = \Omega - \{\omega_i\}$
     $\Phi = \Phi + \{\phi_1, \phi_2\}\}$
   else { degree–2 reduction{
      input: $Q_N(z), \omega_i, \omega_i^*$
      output: $Q_{N-2}(z), \{\phi_1, \phi_2, \phi_3, \phi_4\}\}$
     $Q_N(z) = Q_{N-2}(z)$, $\Omega = \Omega - \{\omega_i, \omega_i^*\}$
     $\Phi = \Phi + \{\phi_1, \phi_2, \phi_3, \phi_4\}\}$
} □

Example 1.

Consider a fourth–order transfer function

$$F(z) = \frac{0.102 + 0.517z^{-1} + 0.830z^{-2} + 0.517z^{-3} + 0.102z^{-4}}{1.000 + 2.267z^{-1} + 1.821z^{-2} + 0.580z^{-3} + 0.070z^{-4}}$$

Here, $\Omega = \{-2.727, -0.367, -0.996 + i0.090, -0.996 - i0.090\}$. Following the above degree reduction procedure, we obtain

$$\begin{aligned} Q_4(z) &= [1.000\ 2.267\ 1.821\ 0.580\ 0.070] \\ Q_3(z) &= [0.367\ 0.696\ 0.405\ 0.052] \\ Q_2(z) &= [0.364\ 0.589\ 0.257] \\ Q_0(z) &= [0.313]. \end{aligned}$$

Here, Q_3, Q_2, Q_0 are the resulting polynomials after two degree–1 and one degree–2 reductions. Fig. 4 shows the detailed filter structure.

$$\begin{aligned} \Phi &= \{(\phi_{01}, \phi_{02}), (\phi_{11}, \phi_{12}), (\phi_{21}, \phi_{22}, \phi_{23}, \phi_{24})\} \\ &= \{(1.195, 3.122), (0.124, 1.943), \\ &\quad (0.536, 0.073, 0.531, -0.153)\} \end{aligned}$$

contains the filter parameters in radians of the degree–1 and degree–2 sections in the figure from the right to the left.

3. PIPELINING OF CORDIC BASED IIR DIGITAL FILTERS

From Fig. 4, we can see that the critical path of the filter goes forward and backward through the entire filter structure, which contains 7 multiplication and 7 addition operations. As the cascaded stages increases, the critical path will also increase. There are almost no pipelining in this structure. Although we can use the *retiming* procedure to transfer part of the delay around the loop, the maximum sample rate can't be increased, since multiple clock cycles are needed to process one sample. One approach to reducing the feedback loop computation time is to introduce a number of dummy transfer zeros at $z = 0$, so that the feedback loop is cut short by the degree–1 section corresponding to $z = 0$ in [6]. But, the pipelining level achieved using this method is limited. In order to achieve arbitrary level pipelining, the following transformation is considered.

Let $F(z^M)$ be the desired filter transfer function of the form z^M. If we replace all the z^M in $F(z^M)$ by z', we obtain a new transfer function $F(z')$. Now we apply the filter synthesis algorithm of section 2 to the transformed transfer function $F(z')$, and obtain the filter structure $\Gamma(z')$, which consists of only rotation blocks and storage elements z'. Due to the computability of the IIR digital filters, there exists at least one storage element in every feedback loop. Notice $F(z')$ is derived from $F(z^M)$ by simply replacing all the z^M by z', the filter structure corresponding to the original transfer function $F(z^M)$ can be derived by simply replacing all the z' in $\Gamma(z')$ by z^M. This leads to the filter structure $\Gamma(z^M)$, which contains at least M delays in every feedback loop. These M delays can be redistributed to appropriate positions using the retiming procedure to achieve pipelining by M levels.

The design of a transfer function which is a function of z^M is given in [2], which is based on the constrained filter design method and the polyphase decomposition technique. The pipelining algorithm is summarized as follows.

Step 1. For given filter specification and pipelining level M, design the filter using any constrained filter design method, which constrains the denominator to be a polynomial in z^M. Therefore, the transfer function can be represented as

$$F(z) = \frac{P(z)}{Q(z^M)} \qquad (1)$$

Step 2. Use the polyphase decomposition technique to

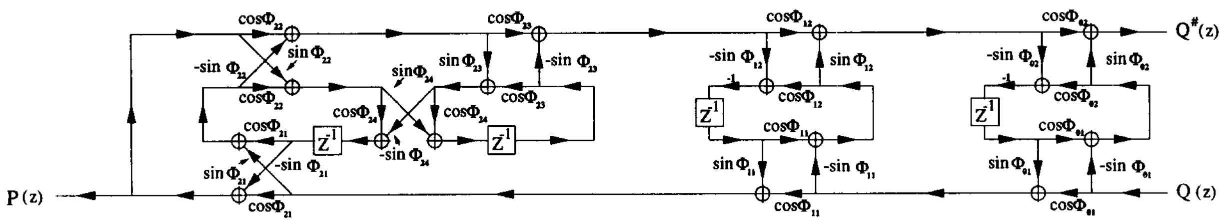

Figure 4. A fourth–order Cordic IIR digital filter

decompose the the transfer function as

$$F(z) = z^{-i} \sum_{i=0}^{M-1} F^{(i)}(z^M), \tag{2}$$

where

$$F^{(i)}(z^M) = \frac{P^{(i)}(z^M)}{Q(z^M)}, \tag{3}$$

$$P^{(i)}(z^M) = \sum_{j=0}^{k} a_{i+jM}\, z^{-jM}, \tag{4}$$

and k is an integer which satisfies the following condition

$$N - M < i + kM \leq N. \tag{5}$$

Step 3. For each i, derive $F^{(i)}(z')$ from $F^{(i)}(z^M)$ by replacing z^M by z'. Synthesize $F^{(i)}(z')$ using the algorithm in section 2. Replacing all the z' back by z^M in the resulting filter structure gives us the M–level pipelined filter architecture of $F^{(i)}(z^M)$.

Step 4. Add the output of each subfilter $F^{(i)}(z^M)$ with appropriate delay element according to (2), and the desired M level pipelined cordic IIR digital filter structure with transfer function $F(z)$ is obtained.

The pipelining scheme diagram of cordic based IIR digital filters is given in Fig. 5.

Example 2.

Consider the low pass filter $F(z)$ with maximum passband ripple of 1 dB, cutoff frequency 0.3π, and minimum stopband attenuation of -20 dB, A pipelinable transfer function with pipelining level 3 is synthesized using Martinez–Parks Decimation method [2] with $M = 3$. The frequency response is shown in Fig. 6. The obtained transfer function is

$$Q(z^3) = [1.000\ 0\ 0\ 2.267\ 0\ 0\ 1.821\ 0\ 0\ 0.580\ 0\ 0\ 0.070]$$
$$P(z) = [0.102\ 0.177\ 0.257\ 0.517\ 0.620\ 0.698\ 0.830\ 0.698\ 0.620\ 0.517\ 0.257\ 0.177\ 0.102].$$

The transfer function $F(z)$ with degree $N = 12$ is decomposed by polyphase decomposition technique. Since $M = 3$, we have three sets of decomposed transfer functions:

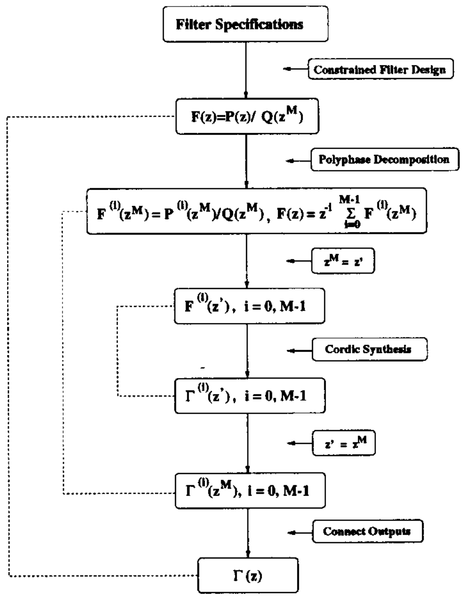

Figure 5. Pipelining scheme of cordic based IIR digital filters

$$Q(z^3) = [1.000\ 0\ 0\ 2.267\ 0\ 0\ 1.821\ 0\ 0\ 0.580\ 0\ 0\ 0.070]$$

$$P^{(0)}(z^3) = [0.102\ 0\ 0\ 0.517\ 0\ 0\ 0.830\ 0\ 0\ 0.517\ 0\ 0\ 0.102]$$
$$P^{(1)}(z^3) = [0.177\ 0\ 0\ 0.620\ 0\ 0\ 0.698\ 0\ 0\ 0.257]$$
$$P^{(2)}(z^3) = [0.257\ 0\ 0\ 0.698\ 0\ 0\ 0.620\ 0\ 0\ 0.177].$$

After the $z' = z^3$ transformation, we have the new transfer functions

$$Q(z') = [1.000\ 2.267\ 1.821\ 0.580\ 0.070]$$

$$P^{(0)}(z') = [0.102\ 0.517\ 0.830\ 0.517\ 0.102]$$
$$P^{(1)}(z') = [0.177\ 0.620\ 0.698\ 0.257]$$
$$P^{(2)}(z') = [0.257\ 0.698\ 0.620\ 0.177].$$

The Φ–parameters are computed as

$$F^{(0)}(z^3):\ \Phi = \{(1.195, 3.122), (0.124, 1.943), (0.536, 0.073, 0.531, -0.153)\}$$
$$F^{(1)}(z^3):\ \Phi = \{(0.910, 3.060), (0.008, 1.571), (0.428, -0.015, 0.620, -0.201)\}$$
$$F^{(2)}(z^3):\ \Phi = \{(0.082, 2.232), (0.008, 1.571), (0.620, -0.010, 0.428, -0.198)\}.$$

Notice example 1 realized the subfilter $F^{(0)}(z')$. Therefore, the filter structure for $F^{(0)}(z^3)$ is essentially the same

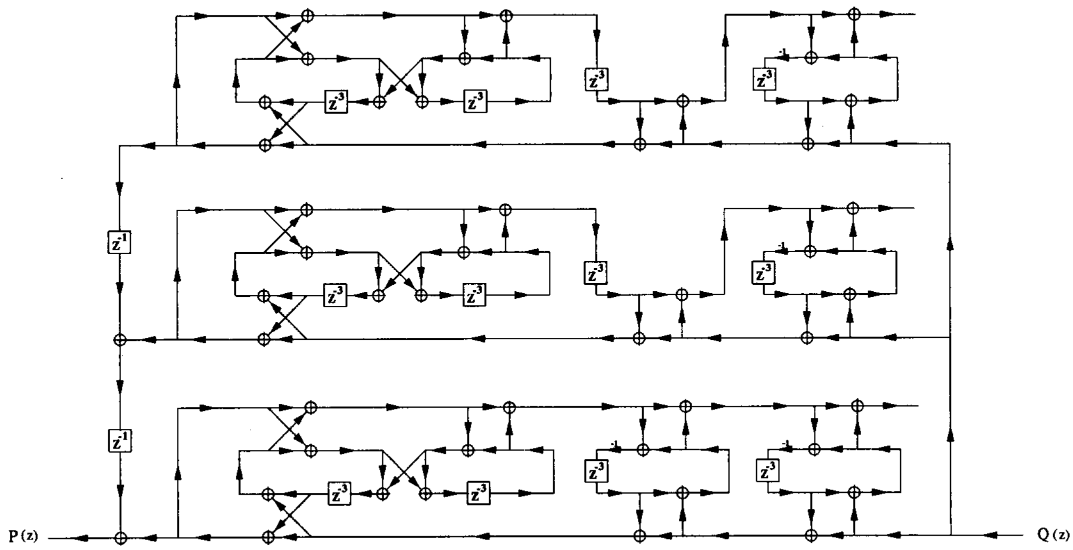

Figure 8. A 3–level pipelined Cordic IIR digital filter

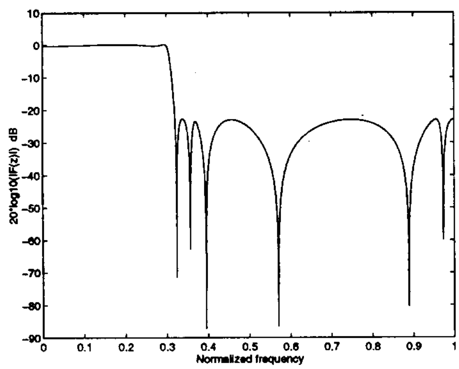

Figure 6. Frequency response of a 3–level pipelined Cordic IIR digital filter. 1.0 **corresponds to** π **for horizontal axis.**

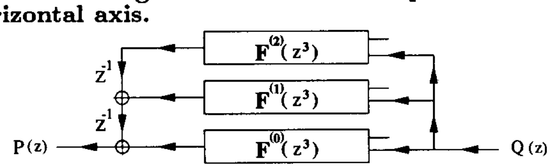

Figure 7. Overall scheme of a 3–level pipelined Cordic IIR digital filter

as in Fig. 4 except all the z^{-1} are replaced by z^{-3}, which is a 3–level pipelined structure. Subfilter $F^{(1)}(z^3)$ and $F^{(2)}(z^3)$ can be similarly synthesized. The final output for $F(z)$ is obtained by adding the output of the three subfilters. Fig. 7 shows the overall scheme of the 3–level pipelined Cordic IIR digital filter. The complete filter architecture of example 2 is shown in Fig. 8, where the multiplication operations are omitted for clarity purpose. In every feedback loop of Fig. 8, there are at least 3 delay elements z^{-1}. These delay elements can be redistributed to appropriate positions using the retiming procedure to achieve pipelining by 3 levels. Each of the three subfilters is itself a 12th order cordic IIR digital filter. Their outputs are connected according to (2) to obtain the final 3–level pipelined 12th order cordic IIR digital filter architecture.

4. CONCLUSION

A pipelining method for the cordic based IIR digital filters is proposed using the constrained filter design methods and the polyphase decomposition technique. Using this method, the maximum filter sample rate can be increased to any desired level.

REFERENCES

[1] A.H. Gray, Jr. and J.D. Markel, "A normalized digital filter structure", *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP–23, pp. 486–494, Oct. 1976.

[2] J.G. Chung and K.K. Parhi, *Pipelined Lattice and Wave Digital Recursive Filters*, Kluwer Academic Publishers, 1996.

[3] S.K. Rao and T. Kailath, "Orthogonal digital filters for VLSI implementation", *IEEE Trans. on Circuits and Systems*, pp. 933–945, Nov. 1984.

[4] P. Rao, E. Deprettere, and P. Dewilde, "Orthogonal Filter Theory and Design", Technical Report, Delft University of Technology, Delft, The Netherlands, 1985.

[5] P. Dewilde, E. Deprettere, and R. Nouta, "Parallel and pipelined VLSI implementation of signal processing algorithms", *VLSI and Modern Signal Processing*,(S. Kung, H. Whitehouse, and T. Kailath, eds.), Englewood Cliffs, NJ: Prentice Hall, Inc., 1984.

[6] E. Deprettere, "Synthesis and Fixed Point Implementation of Pipelined True Orthogonal Filters", *Proc ICASSP–83*, pp. 217–220, 1983.