A Polyphase Decimation Filter for Time-Interleaved ADCs in Direct-RF Sampling Receivers

Yuma Isobe, Takao Kihara, Tsutomu Yoshimura
Osaka Institute of Technology, Japan
October 27, 2018
Outline

Background and objectives
Decimation filter for Conventional direct-RF sampling receiver with
$N$-channel time-interleaved ADC (TI-ADC)

Proposed direct-RF sampling receiver
Polyphase decimation filter with a decimation factor twice as large as $N$ ($D = 2 \times N$)

Simulations and comparisons
Output spectrum
Power consumption
Chip area

Summary
Direct-RF sampling receiver
- Downconverts and filters the RF signals in digital domain.
- Benefits from technology scaling and design automation.
- Reduces the design cost and time to market.

Decimation filter
- Decreases high data rates (>3 GS/s) to low ones (MS/s).
- Reduces the quantization noise of the ADC.

LNA: Low-noise amplifier, ADC: Analog-to-digital converter, CIC: Cascaded integrator-comb, D: Decimation factor
Objectives

Conventional decimation filter
(E.Martens, et al., JSSC 2012)

- Decimation factor, $D$
  - $D = \text{Channel number of TI-ADC, } N$

- Adders need to work at 1 GS/s for 4-GS/s 4-channel TI-ADC.

- Difficulty in realizing medium-bit (6-10) additions for 1 GS/s.

We present a polyphase decimation filter to relax the speed requirement of the adders.

$f_s$: Sampling frequency
I/Q mixer: In-phase/quadraturephase downconversion mixer

4-channel TI-ADC ($N = 4$)

2nd-order polyphase decimation filter ($D = N = 4$)
Proposed Polyphase Filter for 4-Channel TI-ADC

\[ D = 2 \times N (= 8) \]

- Adders work at 500 MS/s for 4-GS/s 4-channel TI-ADC.

We can realize higher-speed decimation filter with little additional power consumption and chip area.
I/Q Mixer

- Downconverters the digitized signal to baseband with a complex signal, \( e^{-j(2\pi f_s/4)t} \).

\[
e^{-j(2\pi f_s/4)t} \Big|_{t = nT_s} = \cos \left( \frac{\pi n}{2} \right) - j \sin \left( \frac{\pi n}{2} \right)
\]

- Eliminate all mixers using 0 and realize I/Q mixing only with multiplications of 1 and -1.

This greatly reduces the hardware cost.
Polyphase Decimation Filter

2nd-order CIC filter with $D = 8$

- Transfer function
  \[ H_{CIC}(z) = \left( \frac{1}{8} \cdot \frac{1 - z^{-8}}{1 - z^{-1}} \right)^2 \]

- Polyphase decomposition
  \[ H_{CIC}(z) = \frac{1}{8} \left[ (1 + 7z^{-8}) + z^{-4}(5 + 3z^{-8}) ight. \]
  \[ + z^{-1}\left( (2 + 6z^{-8}) + z^{-4}(6 + 2z^{-8}) \right) \]
  \[ + z^{-2}\left( (3 + 5z^{-8}) + z^{-4}(7 + z^{-8}) \right) \]
  \[ + z^{-3}\left( (4 + 4z^{-8}) + 8z^{-4} \right) \]

- First three delays and four decimations with $D = 4$
  can be removed in a 4-channel TI-ADC.

Decimators with $D=2$ are inserted before the adders, reducing the operating speeds to $f_s/8$. 
Simulations of Direct-RF Sampling Receiver

- Process: Renesas 65 nm SOTB CMOS
- HDL: Verilog HDL
- Performances: Output spectrum, chip area, power consumption
- Tools: Mathworks MATLAB/Simulink, Candence Verilog-XL, Synopsys Design Compiler, IC Compiler

SOTB: Silicon on thin buried oxide
Comparison of Polyphase Decimation Filters

$D = N = 4$ (conventional)

$D = 2 \times N = 8$ (proposed)

$D = 4 \times N = 16$ (comparative)
Output Spectrum (Proposed)

| TI-ADC | SNR = 74.5 dB @BW = 1.0 × 10⁻³ |
| Decimation filter | SNR = 75.7 dB @BW = 1.0 × 10⁻³ |

Simulation conditions on MATLAB/Simulink

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of data: ( N_D )</td>
<td>( 2^{18} (= 262,144) )</td>
</tr>
<tr>
<td>Sampling frequency: ( f_s )</td>
<td>1</td>
</tr>
<tr>
<td>Frequency of input signal: ( f_{in} )</td>
<td>( f_s/4 + 4/N_D )</td>
</tr>
<tr>
<td>Amplitude of input signal: ( A_{in} )</td>
<td>0.2</td>
</tr>
</tbody>
</table>

BW: Bandwidth

- Desired signal was downconverted around DC.
- Output SNR did not change.

The filter combined with the mixers works properly.
Conventional filter \((D = 4)\)
- Lowest power consumption at 0.77 GHz.
- Cannot operate at more than 0.77 GHz due to the adders.

Proposed filter \((D = 8)\)
- Operates at 1.67 GHz.
- More power consumption than \(D = 4\).

Comparative filter \((D = 16)\)
- Works at 1.82 GHz, limited by D-flip flops.
- Most power consumption.

\[\text{Conditions} \]
\[\begin{align*}
\text{TT, } 25 \degree \text{C,} \\
V_{DD} &= 0.75 \text{ V, } B_{in} = 7
\end{align*}\]

\(V_{DD}\): Supply voltage
\(B_{in}\): Input bit width
The proposed filter consumes twice or more chip area of the conventional one.
Summary

Polyphase decimation filter with $D = 2 \times N$ for N-channel TI-ADCs in direct-RF sampling receivers

- Decimators with $D = 2$, inserted before adders, enable the filter to operate at twice the operating frequency of the conventional method ($D = N$).

- A 7-bit polyphase decimation filter with $D = 8$ and I/Q mixers are designed in a 65-nm SOTB CMOS process.

- Simulations show that the proposed filter works properly at 1.67 GHz with 1.7 mW.
Time-Interleaved ADC (TI-ADC)

4-channel TI-ADC
- Converts analog signals to digital ones with four ADCs.
- Decreases the required sampling rate for one ADC to $f_s/4$.
- The sampled signals are multiplexed together to generate a signal sampled at $f_s$.
- This multiplexer can be realized by the addition in the decimation filter.

![Diagram of 4-channel TI-ADC]
Polyphase Decimation Filter with 4-channel TI-ADC (2nd-order CIC filter with D = 16)

The transfer function:

$$H_{CIC}(z) = \left( \frac{1}{16} \cdot \frac{1 - z^{-16}}{1 - z^{-1}} \right)^2$$

Polyphase decimation filter can be obtained by polyphase decomposition of the transfer function.

First three delays and four decimations with $D=4$ can be removed in a 4-channel TI-ADC.

This Filter is used comparative architecture.

Reducing the operating speeds of the adders to $f_s/16$. 

```
    f_s/4  f_s/8  f_s/16
       \downarrow    \downarrow      \downarrow
       z^{-1}      z^{-1}        z^{-1}      z^{-1}
    \downarrow        \downarrow        \downarrow        \downarrow
    12          12                  12          12
    \downarrow        \downarrow        \downarrow        \downarrow
    10          10                  10          10
    \downarrow        \downarrow        \downarrow        \downarrow
    8           8                   8           8
    \downarrow        \downarrow        \downarrow        \downarrow
    6           6                   6           6
    \downarrow        \downarrow        \downarrow        \downarrow
    4           4                   4           4
    \downarrow        \downarrow        \downarrow        \downarrow
    2           2                   2           2
    \downarrow        \downarrow        \downarrow        \downarrow
    1           1                   1           1
    \downarrow        \downarrow        \downarrow        \downarrow
    0           0                   0           0
```