PAPER

A 70 MHz Multiplierless FIR Hilbert Transformer in 0.35 μm Standard CMOS Library

Yasuhiro TAKAHASHI[†a)], Toshikazu SEKINE[†], *Members*, *and* Michio YOKOYAMA[††], *Nonmember*

SUMMARY This paper presents the implementation of a 31-tap FIR Hilbert transform digital filter chip used in the digital-IF receivers, to confirm the effectiveness of our new design method. Our design method that we previously reported is based on a computation sharing multiplier using a new horizontal and vertical common subexpression techniques. A 31-tap FIR Hilbert transform digital filter was implemented and fabricated in 0.35 μm CMOS standard cell library. The chip's core contains approximately 33k transistors and occupies 0.86 mm^2. The chip also has an operating speed of 70 MHz over. The implementation results show that the proposed Hilbert transformer has a smallest cost factor and so that is a high performance filter.

key words: *Hilbert transformer, multiplierless, FIR filter, multiple constant multiplication (MCM), canonic signed digit (CSD)*

1. Introduction

The digital Hilbert transformer is a basic signal processing operator used in numerous application domains, e.g., telecommunications, wireless multimedia communication, speech/audio/image processing, and many more. The advantages of digital Hilbert transformer VLSI implementations include the ability to easily program the hardware to accommodate different data rates, modulation formats. For example, by simply scaling the clock frequency, the same chip set can be used across several production lines, e.g., an Inphase/Quadrature (I/Q) demodulator used in receivers of HDTV [1], and an array antenna radar system [5]. These I/Q demodulator must operate at sampling rates in excess of 70 MHz, to accommodate intermediate-frequency (IF) frequencies in a range between 20–40 MHz.

The digital Hilbert transformer used in the I/Q demodulator chip can be realized of either the finite impulse response (FIR) [2]–[5], [7] and the infinite impulse response (IIR) [6] filters. Since the Hilbert transformer is required to strictly shift the phase of signal by 90 degrees over wide range frequency, we generally will have the choice of using FIR with linear phase. However, the Hilbert transformer using FIR filter is attributable to its amplitude ripple and large circuit scale.

The primary source of complexity in Hilbert transformer using FIR digital filters is the filter coefficients and the filter word length. These coefficients have been traditionally represented in binary number format and implemented as constant multipliers [2], [3]. One way of implementing more efficient multipliers is to represent the filter coefficients in canonic signed digit (CSD) format [4], [5]. The FIR filter's latency of N clock cycles, where N is the number of coefficients, is the same as that of the Hilbert transformer design. However, comparison of the two (or more) designs assuming a constant operating time is utterly unfair. For Hilbert transformer design, operating time is dependent on the filter coefficients, since as N grows, longer interconnects will be involved and larger loads will be driven. While it is conceivable that despite the broadcast overhead increasing the operating time, the area saving might still lead to a cost-effective design, realistic analyses require an architecture and technology dependent model to justify a tradeoff. Thus we should compare various filter designs using some kind of performance function.

In this paper, we present high performance implementations for digital Hilbert transformers based on our recently proposed constant multiplier [8], [9]. The constant multiplier is designed by using common subexpression elimination (CSE) techniques and so results in low hardware complexity and a small VLSI chip area. The rest of this paper is organized in four sections. Section 2 describes the design method of digital FIR Hilbert transformer. Section 3 presents the architecture of the digital Hilbert transformer based on a computation sharing multiplier using horizontal and vertical CSE techniques. Section 4 shows the implementation results of the 31-tap digital FIR Hilbert transformer fabricated in a 0.35 μm CMOS process using a standard static logic cell. The target of the Hilbert transformer to be implemented is the I/Q demodulator with operating speed of 70 MHz over which is used in the digital-IF receivers. This section also discusses the cost-performance tradeoffs and the scaling on the proposed design. Finally, conclusions are drawn in Sect. 5.

2. Design Method of FIR Hilbert Transformer

2.1 Discrete Hilbert Transform

The Hilbert transformer converts the phase of the incoming time samples by −90 degrees for positive frequencies and by +90 degrees for negative frequencies. The frequency spec-

Manuscript received September 11, 2006.
Manuscript revised February 8, 2007.
Final manuscript received March 16, 2007.

[†]The authors are with the Department of Electrical and Electronic Engineering, Faculty of Engineering, Gifu University, Gifu-shi, 501-1193 Japan.

[††]The author is with the Department of Bio-system Engineering, Faculty of Engineering, Yamagata University, Yonezawa-shi, 992-8510 Japan.

a) E-mail: yasut@gifu-u.ac.jp

DOI: 10.1093/ietfec/e90–a.7.1376

trum of an ideal Hilbert transformer is written as

$$Y(\omega) = \hat{X}(\omega) = \begin{cases} -jX(\omega) & (\omega \geq 0) \\ +jX(\omega) & (\omega < 0), \end{cases} \tag{1}$$

where $\hat{X}(\omega)$ represents the Hilbert transform of $X(\omega)$.

The frequency relationship expressed in Eq. (1) is obtained from system whose impulse response $h(t) = -1/\pi t$. The Hilbert transform relationship for time signal is given by the convolution integral

$$y(t) = \hat{x}(t) = h(t) * x(t) = \frac{1}{\pi}\int_{-\infty}^{+\infty} \frac{x(\tau)}{\tau - t} d\tau, \tag{2}$$

where $\hat{x}(t)$ denotes the Hilbert transform of $x(t)$, $*$ represents convolution and τ is the variable of integration.

For discrete time sequences, the Hilbert transform frequency relationship is slightly different, because the spectra of discrete signals are periodic between $-\pi$ and $+\pi$. In Ref. [10], Rabiner and Schafer have given the frequency response of an ideal Hilbert transformer as:

$$H(e^{j\omega}) = \begin{cases} -j & (0 \leq \omega \leq \pi) \\ +j & (-\pi \leq \omega < 0). \end{cases} \tag{3}$$

The impulse response sequence becomes:

$$h(N) = \begin{cases} \dfrac{\sin^2\left(\dfrac{\pi N}{2}\right)}{\dfrac{\pi N}{2}} & \text{for} \quad |N| > 0 \\ 0 & \text{for} \quad N = 0. \end{cases} \tag{4}$$

Note that $h(N) = 0$ for all even N and $h(-N)$, facts which are exploited in order to reduce the required circuitry. Of course, the ideal transform is not physically realizable since it is non-causal and infinite duration. In order to create a realizable filter, the impulse response is windowed and shifted to make it causal.

2.2 Architectural Design of FIR Hilbert Transformer

The FIR Hilbert transformer is usually realized by using parallel connection of case 3 (i.e. "anti-symmetric" impulse response with "odd" length) FIR filter and delay elements. Figure 1 shows the causal FIR Hilbert transformer. In Fig. 1, $x[N]$ and $y[N]$ are input and output signal, respectively. The analytic signal written in the following formula

$$\begin{aligned} y[N] &= Re(y[N]) + jIm(y[N]) \\ &= y_{out1} + jy_{out2} \end{aligned} \tag{5}$$

can be generated by using a Hilbert transformer. In the basic non-causal system for the generation of the analytic signal, $Re(y[N]) = x[N]$. The basic causal system differs from the non-causal only by a time shift of $(N-1)/2$ sample, i.e.

$$y_{out1} = Re(y[N]) = x\left[\frac{N-1}{2}\right]. \tag{6}$$

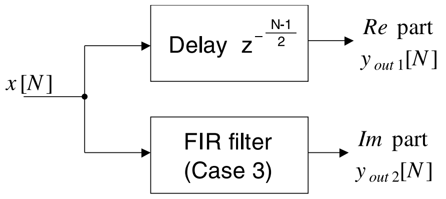


Fig. 1 Casual FIR Hilbert transformer.

	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}	2^{-9}	2^{-10}	2^{-11}	2^{-12}	2^{-13}	2^{-14}	2^{-15}	2^{-16}
h(0)	0	0	0	0	n	0	0	n	0	1	0	0	0	1	0	n
h(1)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
h(2)	0	0	0	0	0	n	0	n	0	n	0	0	0	1	0	0
h(3)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
h(4)	0	0	0	0	0	n	0	0	n	0	n	0	1	0	0	0
h(5)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
h(6)	0	0	0	0	0	0	n	0	1	0	0	0	0	0	0	n
h(7)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
h(8)	0	0	0	0	0	n	0	n	0	n	0	0	0	0	0	n
h(9)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
h(10)	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	n
h(11)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
h(12)	0	1	0	n	0	n	0	1	0	n	0	1	0	1	0	n
h(13)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
h(14)	1	0	1	0	0	0	0	0	0	n	0	n	0	n	0	1

Fig. 2 CSD represented coefficient sets of FIR Hilbert transform filter.

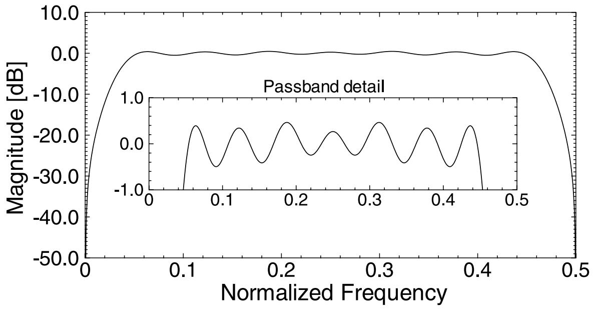


Fig. 3 Frequency response of FIR Hilbert transform filter.

This causal system can be with linear phase characteristics.

The actual filter is based on the design of a 31-tap Hilbert transform filter which was optimized for minimum stopband energy. An ideal 31-tap Hilbert transform filter was designed first using program based on the design technique presented in Ref. [10]. The resulting floating-point coefficients were then converted to an optimized CSD representation as shown in Fig. 2, where n denotes -1. The frequency response of the CSD Hilbert transform filter is shown in Fig. 3. The passband is between $0.05f_s$ and $0.45f_s$, where f_s is a sampling frequency. The passband ripple is less than 1 dB and stopband attenuation is greater than 40 dB.

3. FIR Hilbert Transformer Chip Architecture

3.1 Folded Transpose Filter Structure

Parallel architectures typically require a substantial amount of die-area and consume a large amount of power. For example, it has been shown in Ref. [4] that high-performance fixed coefficient FIR filters is realized with an architecture based on the transposed filter structure. The advantages of this structure include a small critical path (data broadcast delay + one multiplier + one adder + one register setup time). Furthermore, it is well known that for linear phase filters, the folded transpose filter structure may be used, halving the number of required general multipliers. Also, we can implement more efficient multipliers to represent the filter coefficients in CSD format.

3.2 Constant Multiplier Structure

Multiple constant multiplication (MCM) can be implemented efficiently by using dedicated shift and add multipliers. Common subexpression elimination (CSE) as a way to tackle the MCM problems is as a possible method for the optimization of finite-duration FIR filter area through the reduction of the multiplier block logic [8], [9], [11]–[16]. In general, the goal of CSE can be defined as follows.

1. Identify multiple patterns in the coefficient set.
2. Remove these patterns and calculate them only once.

Our CSE techniques [8], [9] have been proposed an efficient way to find the correct bit-patterns for horizontal and vertical CSEs. The proposed CSE is stated as the problem of minimizing the numbers of the delay and adders/subtracter blocks which are needed to perform all of the multiplications. That is, the objective cost function (CF) to be minimized is written as:

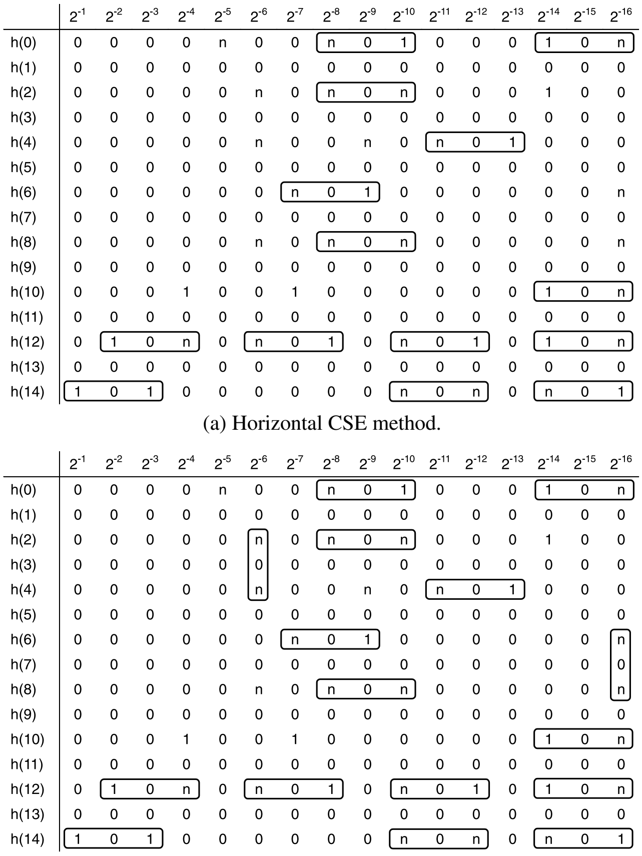
$$CF = \beta N_{reg} + N_{as} \quad \text{for} \quad 0 < \beta < 1, \tag{7}$$

where N_{reg} and N_{as} are the number of registers and adders/subtracters respectively, and β is weights. We set the parameter β to 0.15–0.2, if we assume that the FIR filters are fabricated in a 0.35 μm standard CMOS process. Using the proposed method, the MCM area of the FIR filters including the Hilbert transform filter is reduced by an average of 20% [9].

The idea of the proposed CSE can be demonstrated on a 31st Hilbert transform filter design shown in Fig. 4. The proposed technique is also described by the pseudo C language code shown in Fig. 5. The proposed CSE is completed by the following two steps.

3.2.1 Step 1: Horizontal CSE Method

In the horizontal CSE method, we must be examined all combinations of non-zero bit patterns in a coefficient. Since a bit pattern can only be eliminated once, we must also detect the occurrence of the same patterns within each other. For example, the valid non-zero bit patterns of coefficient 010n010n are summarized in Table 1, where n denotes -1. This table shows that we should select 10n as the most frequency of non-zero bit pattern. Note that 10n and 10001 can be made equivalent in their implementation to n01 and n000n. Table 2 summarizes the frequency of the valid non-zero bit patterns in 31st order Hilbert transform filter coefficients. In this case, pattern 10n is identified as most frequent for the coefficients. If two patterns have the same frequency (> 1), the smallest pattern is chosen. Because, adder/subtracter structures with a bigger wordlength cause a larger implementation area. Most common horizontal subexpressions resulting from the proposed method is extracted from the coefficient table represented in canonic signed digit (CSD) shown in Fig. 4(a).



	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}	2^{-9}	2^{-10}	2^{-11}	2^{-12}	2^{-13}	2^{-14}	2^{-15}	2^{-16}
h(0)	0	0	0	0	n	0	0	n	0	1	0	0	0	1	0	n
h(1)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
h(2)	0	0	0	0	0	n	0	n	0	n	0	0	0	1	0	0
h(3)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
h(4)	0	0	0	0	0	n	0	0	n	0	n	0	1	0	0	0
h(5)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
h(6)	0	0	0	0	0	0	n	0	1	0	0	0	0	0	0	n
h(7)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
h(8)	0	0	0	0	0	n	0	n	0	n	0	0	0	0	0	n
h(9)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
h(10)	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	n
h(11)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
h(12)	0	1	0	n	0	n	0	1	0	n	0	1	0	1	0	n
h(13)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
h(14)	1	0	1	0	0	0	0	0	0	n	0	n	0	n	0	1

(a) Horizontal CSE method.

	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}	2^{-9}	2^{-10}	2^{-11}	2^{-12}	2^{-13}	2^{-14}	2^{-15}	2^{-16}
h(0)	0	0	0	0	n	0	0	n	0	1	0	0	0	1	0	n
h(1)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
h(2)	0	0	0	0	0	n	0	n	0	n	0	0	0	1	0	0
h(3)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
h(4)	0	0	0	0	0	n	0	0	n	0	n	0	1	0	0	0
h(5)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
h(6)	0	0	0	0	0	0	n	0	1	0	0	0	0	0	0	n
h(7)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
h(8)	0	0	0	0	0	n	0	n	0	n	0	0	0	0	0	n
h(9)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
h(10)	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	n
h(11)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
h(12)	0	1	0	n	0	n	0	1	0	n	0	1	0	1	0	n
h(13)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
h(14)	1	0	1	0	0	0	0	0	0	n	0	n	0	n	0	1

(b) Final horizontal and vertical CSE method.

Fig. 4 Proposed horizontal and vertical CSE in 31st order Hilbert transform filter coefficients.

3.2.2 Step 2: Vertical CSE Method

The remaining non-zero bits are examined for optimum vertical common subexpression. Pattern identification of vertical CSE is the same as that of horizontal CSE. In this case, vertical pattern n0n as shown in Fig. 4(b) is identified as most frequent for the coefficients. Figure 4(b) also displays a final coefficient table of 31st order Hilbert transform filter processed by the pseudo code.

3.3 Adder Structure

3.3.1 Carry Save Adder

For high-speed architecture, simple carry propagate adder

Efficient horizontal and vertical CSE (EHV-CSE)

```
void main()
{
    **Eliminate** zero coefficients;
    **Merge** coefficients with the same value;
    **Construct** the initial coefficient matrix;
    for horizontal CSE
      {
        **Find** coefficients with identical pattern;
        **Extract** identical pattern;
        **Update** the coefficient matrix;
          if (identical pattern = 0) {
              break;
            }
            else {
              return;
            }
      }
    for vertical CSE
      {
        **Find** coefficients with identical pattern;
        **Extract** identical pattern;
        **Update** the coefficient matrix;
          if (similar pattern = 0) {
              **Output** signal flow graph;
              exit(0);
            }
            else {
              return;
            }
        }
}
```

Fig. 5 Pseudo C code of the proposed CSE algorithm.

Table 1 Non-zero bit patterns of coefficient 010*n*010*n*.

Bit pattern	Frequency
10*n*, *n*01	3
10001, *n*000*n*	2
100000*n*	1

Table 2 Frequency of bit patterns in 31st order Hilbert transform filter coefficients.

Bit pattern	Frequency
10*n*, *n*01	11
101, *n*0*n*	10
1001, *n*00*n*	3
1000*n*, *n*0001	7
10001, *n*000*n*	6
100000*n*, *n*000001	5
1000001, *n*00000*n*	5
10000001, *n*000000*n*	3
10000000*n*, *n*00000001	5

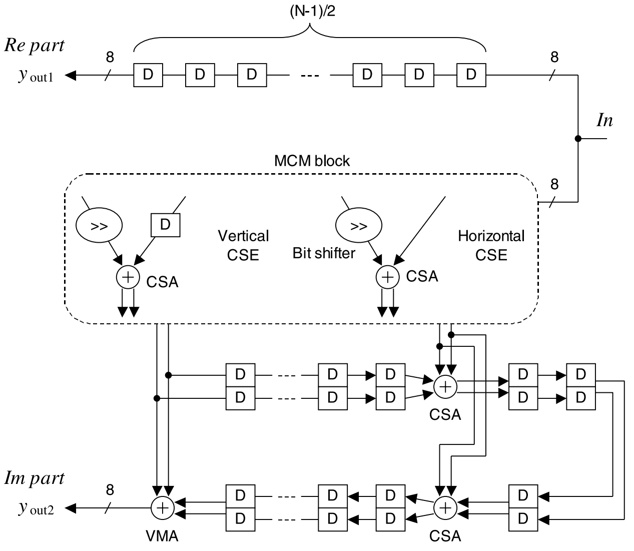


Fig. 6 Final FIR Hilbert transformer structure.

(CPA) is not sufficient to achieve the required throughput rate[†]. This problem is dealt with in one of two ways, by using a high speed CPA technique such as a conditional sum adder or carry look-ahead adder, or by eliminating the need for carry propagation through the use of a redundant addition scheme such as signed-digit arithmetic or carry save arithmetic. In this paper, carry-save-adder (CSA) is chosen rather than using a fast CPA, because of the large number of such adders which would be required in the filter.

The tradeoff is that two z^{-1} registers are required: one to delay the sum and one to delay the carry. This is an efficient tradeoff since it doubles the register hardware which leads to an area increase 10%, but results in a speed increase by factor of F:

$$F = b \times \frac{T_c}{T_s}, \tag{8}$$

where b is the wordlength, T_c is the carry-ripple delay per bit and T_s is the sum delay per bit. Typically, $T_c/T_s \leq 1.0$, this means that the CSA provides a speedup by a factor of up to b [18].

3.3.2 Vector Merge Adder

A vector merge adder (VMA) has to be used in the final stage to add the final sum and carry at the base of the carry save adder tree. The vector merge adder is a traditional adder responsible for calculating the final filter output, and is the only part of the system that has combinational delay longer than that of a single full adder. A key point is that while the natural choice in a VLSI system would be implement the VMA with a sophisticated addition technique such as carry look ahead addition and square root carry select addition to improve delay. As the filter in this paper is required high-speed architecture[†], we choose the square root

[†] For example, Zimmermann [17] has reported that the delay of 16 bit carry propagate adder and carry look-ahead adder are 12–20 ns and 8–11 ns respectively, on the other, those of 16 bit carry-select-adder (CSLA) and carry-save-adder (CSA) are 7–10 ns and 6–8 ns respectively. In this paper, the target of the Hilbert transformer to be implemented is the I/Q demodulator with operating speed of 70 MHz ($\simeq 1/14$ ns), and then we think that it is usually preferable to choose the CSLA and the CSA.

carry select adder.

3.4 Final FIR Hilbert Transformer Structure

By using the structures presented in the previous subsections, we obtain the final FIR Hilbert transformer structure shown in Fig. 6, which requires a smaller chip size, a faster speed, and has less power dissipation after ASIC implementation.

4. Hilbert Transformer VLSI Implementation

4.1 Considered Application Example: Wideband Radar Receiver

In this application example, we assume that the receiver with FIR Hilbert transformer is to be used in an array antenna radar system [5]. This receiver consists of two parts: a RF part with only one mixer stage and an I/Q demodulation part. The signal is then bandpass sampled by the analog-digital-converter (ADC) at IF. The I/Q demodulation is performed in the digital domain and is based on the FIR Hilbert Transformer (HT) in a receiver structure. Such receiver is shown in Fig. 7. The input signal to the receiver is in the range of 8–12 GHz. The IF is 360 MHz and the sample frequency of ADC is 80 MHz. After the I/Q demodulation the sample frequency is 40 MHz.

4.2 Implementation and Testing Results

We designed an experimental chip for the 31st order FIR Hilbert transformer chip as shown in Fig. 8. The chip uses 0.35 μm 3.3 V CMOS provided by the VLSI Design and Education Center (VDEC), the University of Tokyo. The target library is the VDEC EXD library for Rohm 0.35 μm CMOS technology. The Resistor Transfer Level (RTL) and gate-level netlists are written in Verilog-HDL. We use Synopsys Design Compiler for logic synthesis, Avant! Apollo for physical-layout and Cadence Verilog-XL for simulation. Table 3 summarizes main specifications of the FIR Hilbert transformer. From the implementation results, the core size is 0.86 mm^2, and it integrates about 33k transistors.

In order to verify the operation of the chip, a pseudo noise signal which is generated from a data generator (Tektronix DG-2040) is inputted into the proposed FIR Hilbert transformer. The resulting waveforms were digitized with a Tektronix TDS-7054 digital oscilloscope. Figure 9 shows a digital oscilloscope trace of an MSB output of a chip operating at 71 MHz. The power dissipation is also 263 mW at

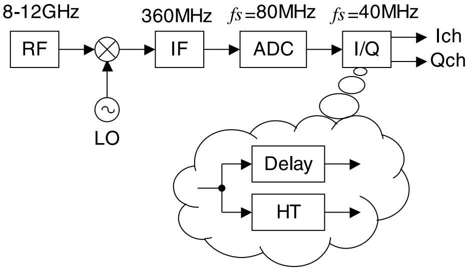


Fig. 7 Receiver structure with digital I/Q demodulation.

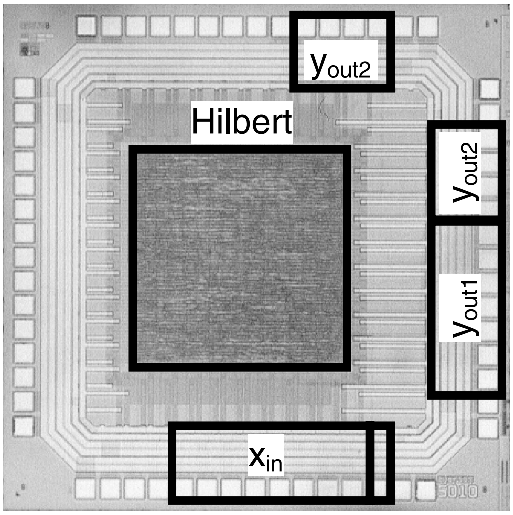


Fig. 8 Photomicrograph of the 31st order FIR Hilbert transformer chip.

Table 3 FIR Hilbert transformer chip specifications.

Feature	Value
Technology	0.35 μm CMOS, 3-layer metal, 2-layer poly
Power supply	3.3 V
Chip size	2.4 × 2.4 mm^2
Core size	0.86 mm^2
Number of transistors	32988
Filter order	31 taps
Normalized passband frequency	$[0.05f_s,\ 0.45f_s]$
Passband ripple	0.92 dB
Stopband attenuation	41.9 dB
I/O word	8 bits
Internal word	16 bits
Clock frequency	71 MHz
Power dissipation	263 mW @ 70 MHz

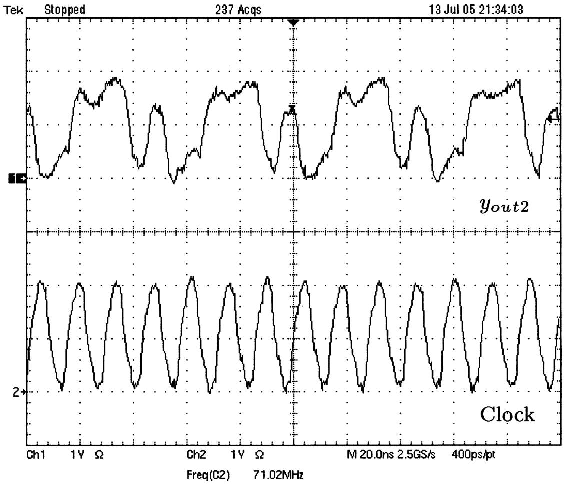
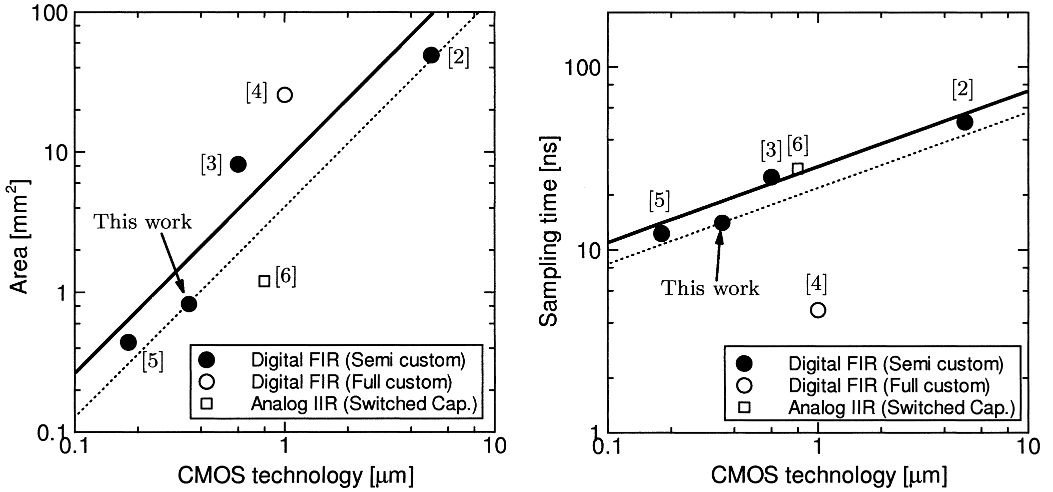


Fig. 9 MSB output at 71 MHz.



(a) Relationship of area between CMOS technology.

(b) Relationship of sampling time between CMOS technology.

Fig. 10 Technical roadmap of Hilbert transformer. The numbers in the brackets are references.

Table 4 Comparison of different Hilbert transformers.

year	Area [mm^2]	Time [ns]	N	b	Cost ($\times 10^{-3}$)	Feature
1991 [2]	49.2	53.4	15	8	28.5	5 μm Gate Array 2's comp. representation FIR filter Power dissipation: n/a
1994 [3]	8.20	100	31	15	6.70	0.6 μm FPGA XC3090 2's comp. representation FIR filter Power dissipation: 780 mW @ 10 MHz
1995 [4]	25.6	3.33	43	18	2.50	1.0 μm CMOS full custom design CSD representation FIR filter Power dissipation: 1.8 W @ 300 MHz
2002 [5]	0.44	12.3	17	11	3.90	0.18 μm CMOS semi custom design CSD representation FIR filter Power dissipation: n/a
2002 [6]	1.20	55.6	3	5	122	0.8 μm CMOS full custom design Analog Switched Capacitor IIR filter Power dissipation: 51 mW @ 18 MHz
2005 (proposed)	0.82	14.1	31	16	2.10	0.35 μm CMOS semi custom design CSD representation, CSE FIR filter Power dissipation: 263 mW @ 70 MHz

70 MHz.

Figure 10 shows the technical roadmap has been reported over the past decade or so. The straight-line represents the trend for the semi-custom LSI design. If the Hilbert transformer which uses a new approach breaks the trendline, we can say that Hilbert transformer is characterized by high-bit-rate, small circuit scale. Trends regarding the chip area of a Hilbert transformer are summarized in Fig. 10(a). We found that this is the minimum reported area for a Hilbert transformer which is fabricated by using semi-custom design, and we can estimate a 47% reduction in the area compared with the earlier Hilbert transformer made using the same fabrication technology. Figure 10(b) shows the trends regarding the sampling time of a Hilbert transformer. From this figure, we found that this is the minimum reported area for a Hilbert transformer and we can estimate a 24% speed up compared with the earlier Hilbert transformer made using the same fabrication technology.

As the reported Hilbert transformers have been fabricated by various integrated circuits (e.g. FPGA, ASIC, Gate Array) and have been designed from analog/digital FIR/IIR filters, it is unfair to evaluate these transformers by the coverage of one condition. Therefore in order to compare various filter designs, we define the cost factor C for Hilbert transformer evaluation considering the performance-cost ratio as the following equation:

$$C = \frac{\log(A \times T)}{N \times b}, \tag{9}$$

where $A\times T$ is a term of area-time product and $N\times b$ is a term of taps-wordlength product. This equation means that as C decreases, the area-time product of Hilbert transformer per taps-wordlength product gets smaller. Table 4 summarizes the comparison of different Hilbert transformers. From this table the results show that the proposed Hilbert transformer has a smallest cost factor and so that is a best performance filter regardless of process, design, etc.

5. Conclusions

We have presented the design and implementation of a 31-tap FIR Hilbert transform digital filter chip. The architecture of the filter has been based on a computation sharing multiplier using horizontal and vertical common subexpression elimination techniques. The chip has been implemented by using a 0.35 μm CMOS process technology with the area of 0.86 mm^2. The chip has a clock frequency of 71 MHz and a power consumption of 263 mW at 70 MHz.

Acknowledgment

The Hilbert transformer chip in this paper has been designed with CAD tools of Synopsys, Inc. and Cadence Design Systems, Inc., and has been fabricated through the chip fabrication program of VLSI Design and Education Center (VDEC), the University of Tokyo with the collaboration by Rohm Corporation and Toppan Printing Corporation.

References

[1] Advanced Television System Committee, “ATSC, digital television standard for HDTV transmission,” Sept. 1995.

[2] R.N. Gorgui-Naguib and S.S. Dlay, “Hardware implementation of a Hilbert transformer,” Proc. IEEE Mediterranean Electrotechnical Conf. (MELECON’91), pp.404–407, May 1991.

[3] S. He and M. Torkelson, “FPGA implementation of FIR filters using pipelined bit-serial canonical signed digit multipliers,” Proc. IEEE Custom Integrated Circuits Conf. (CICC’94), pp.81–84, May 1994.

[4] R. Hawley, T. Lin, and H. Samueli, “A 300 MHz digital double-sideband to single-sideband converter in 1 μm CMOS,” IEEE J. Solid-State Circuits., vol.30, no.1, pp.4–10, June 1995.

[5] H. Ohlsson and L. Wanhammar, “A digital down converter for a wideband radar receiver,” Proc. National Radio Science Conf. (NRSC 2002), pp.478–481, June 2002.

[6] J.G.R.C. Gomes and A. Petraglia, “An analog sampled-data DSB to SSB converter using recursive Hilbert transformer for accurate I and Q channel matching,” IEEE Trans. Circuit & Syst. II, vol.49, no.3, pp.177–186, March 2002.

[7] Y. Takahashi, T. Kitajima, and K. Takahashi, “Hilbert transformer design using CSD FIR filter,” Proc. 2001 Int. Tech. Conf. on Circuits/Systems Computers and Communications (ITC-CSCC 2001), pp.921–924, July 2001.

[8] Y. Takahashi, K. Takahashi, and M. Yokoyama, “Synthesis of multiplierless FIR filter by efficient sharing of horizontal and vertical common subexpression elimination,” Proc. ITC-CSCC 2004, pp.7C2L-4-1–7C2L-4-4, July 2004.

[9] Y. Takahashi and M. Yokoyama, “New cost-effective VLSI implementation of multiplierless FIR filter using common subexpression elimination,” Proc. IEEE Int. Symp. on Circuits and Systems (ISCAS 2005), pp.845–848, May 2005.

[10] L.R. Rabiner and R.W. Schafer, “On the behavior of minimax FIR digital Hilbert transformers,” Bell Syst. Tech. J., vol.53, no.2, pp.363–380, Feb. 1974.

[11] R. Hartley, “Optimization of canonic signed digit multipliers for filter design,” Proc. ISCAS’91, pp.1992–1995, June 1991.

[12] M. Mehendale, S.D. Sherlekar, and G. Venkatesh, “Synthesis of multiplier-less FIR filters with minimum number of additions,” Proc. IEEE/ACM Int. Conf. on Computer-Aided Design (ICCAD’95), pp,668–671, Nov. 1995.

[13] M. Potkonjak, M.B. Srivastava, and A. Chandrakasan, “Multiple constant multiplications: Efficient and versatile framework and algorithms for exploring common subexpression elimination,” IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.15, no.2, pp.151–165, Feb. 1996.

[14] R. Paško, P. Schaumout, V. Derudder, S. Vernalde, and D. Ďuračková, “A new algorithm for elimination of common subexpressions,” IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.18, no.1, pp.58–68, Jan. 1999.

[15] Y. Jang and S. Yang, “Low-power CSD linear phase FIR filter structure using vertical common sub-expression,” Electron. Lett., vol.38, no.15, pp.777–779, July 2002.

[16] A.P. Vinod, E.M-K. Lai, A.B. Premkumar, and C.T. Lau, “FIR filter implementation by efficient sharing of horizontal and vertical common subexpressions,” Electron. Lett., vol.39, no.2, pp.251–253, Jan. 2003.

[17] R. Zimmermann, Binary Adder Architectures for Cell-Based VLSI and Their Synthesis, Ph.D. Dissertation, Swiss Federal Institute of Technology (ETH), Zurich, 1997.

[18] R. Jain, P. Yang, and T. Yoshino, “FIRGEN: A computer-aided design system for high performance FIR filter integrated circuits,” IEEE Trans. Signal Process., vol.39, no.7, pp.1655–1668, July 1991.

Yasuhiro Takahashi was born in Yamagata, Japan, in July 1977. He received the B.E., M.E. and Ph.D. degrees in electronic engineering from Yamagata University, Japan, in 2000, 2002 and 2005, respectively. From 2005 to 2007, he was a Research Associate in the Faculty of Engineering, Gifu University, Japan, where he is currently an Assistant Professor at the same university. His research interests include VLSI hardware architecture for digital signal processing and communication systems with particular emphasis in low-complexity and low-power design. Dr. Takahashi is a member of IEEE.

Toshikazu Sekine received the B.E. M.E. and Ph.D. degrees in electronic engineering from Yamagata University in 1974, 1976 and 2002, respectively. Since 1976, he has been with the Faculty of Engineering, Gifu University and currently he is an Associate Professor. His current research interests include electromagnetic compatibility (EMC) analysis, lossy transmission line modeling, and microwave system and high-speed PCB signal integrity analysis. Dr. Sekine is a member of IEEE.

Michio Yokoyama received the B.E. degree in electrical engineering from Yamagata University in 1989, and the M.E. degree in electrical and communication engineering and the Ph.D. degree in electronic engineering from Tohoku University, Japan, in 1991 and 1994, respectively. From 1994 to 2001, he joined Research Institute of Electrical Communication, Tohoku University, where he engaged in research on the design and development of RF–CMOS devices such as power amplifier module for digital cellular phone system. Since 2001, he has joined Yamagata University and engaged in development of RF CMOS circuits and RF system packages. Dr. Yokoyama is a member of IEEJ, JIEP and JSAP.