

RESEARCH

Open Access

# Efficient implementation of 90° phase shifter in FPGA

Punithavathi Duraiswamy\*, Johan Bauwelinck and Jan Vandewege

## Abstract

In this article, we present an efficient way of implementing 90° phase shifter using Hilbert transformer with canonic signed digit (CSD) coefficients in FPGA. It is implemented using 27-tap symmetric finite impulse response (FIR) filter. Representing the filter coefficients by CSD eliminates the need for multipliers and the filter is implemented using shifters and adders/subtractors. The simulated results for the frequency response of the Hilbert transformer with infinite precision coefficients and CSD coefficients agree with each other. The proposed architecture requires less hardware as one adder is saved for the realization of every negative coefficient compared to conventional CSD FIR filter implementation. Also, it offers a high accuracy of phase shift.

## Introduction

Phase shifters have wide spread applications [1-3], in particular, 90° phase shifters are used in wireless communication for single side band generation, IQ modulation and image rejection. Digital implementation of 90° phase shifters is in high demand as many of the digital communication systems use FPGAs [4-6]. The FPGAs offer a wide range of performance for implementing DSP algorithms and hence it is important to efficiently map the algorithm in order to optimize the area and speed. Hilbert transformer is the most widely used 90° phase shifter implemented either by a finite impulse response (FIR) [7-9] or by an infinite impulse response (IIR) filter [10]. Since the Hilbert transformer is required to shift the phase by 90° over a wide range of frequencies, FIR filter with linear phase is preferred over IIR filter. The conventional method of implementing the FIR filter is by adders/subtractors and multipliers. The computational time of a multiplier is more than that of an adder/subtractor and also occupies more hardware. In order to increase the speed and reduce the hardware cost, the coefficients of the filter are represented in canonic signed digits (CSD), which is the representation of the coefficients in powers of two. The resulting FIR filter can be implemented using shifters and adders/subtractors without multipliers. The CSD coefficients have minimum number of non-zero values when compared

to other radix-2 representations which in turn again reduces the number of adders [11]. In this article, an efficient way of implementing the FIR filter with CSD coefficients is proposed which reduces the number of adders/subtractors compared to the conventional way of implementing the filter with CSD coefficients. In our application, this 90° phase shifter is used for digital image rejection of a demodulated RF signal on the receiver side of a wireless communication system.

## CSD representation of filter coefficients

Encoding a binary number such that it contains the fewest number of non-zero bits is called canonic signed digit (CSD). It maps a number to a ternary system  $\{-1,0,1\}$  versus a binary system  $\{0,1\}$ . The following are the properties of CSD numbers:

No two consecutive bits in a CSD number are non-zero.

The CSD representation of a number contains the minimum possible number of non-zero bits, thus the name canonic.

The CSD representation of a number is unique. Among the  $N$ -bit CSD numbers in the range  $[-1,1]$ , the average number of non-zero bits is  $N/3 + 1/9 + O(2 - N)$ .

Hence, on average, CSD numbers contain about 33% fewer non-zero bits than two's complement numbers [11]. Now the multipliers in the digital filters are realized using shifters and adders/subtractors. This results in the area reduction for digital filters. Here, the

\* Correspondence: punithavathi@intec.ugent.be  
Ghent University, INTEC/IMEC, Sint-pietersnieuwstraat 41, 9000 Ghent, Belgium

complexity of a digital filter is a function of the number of non-zero digits in the filter coefficient. Encoding the filter coefficient using the CSD representation reduces the number of partial products as well as the area and the power consumption. Hence, it is a useful technique for implementing FIR Filters with fixed coefficients. Digital FIR filters can take advantage of this CSD representation.

In order to simplify the hardware, the filter is designed with coefficients represented by powers of two (CSD). The following are the specifications used for the FIR filter design.

- Symmetry: Negative
- Sampling frequency: 140 MHz
- Pass band (normalized): 0.1-0.9
- Number of taps: 27
- Pass band ripple: 0.1 dB
- Stop band attenuation: 50 dB
- Coefficients precision: 12 bits
- CSD precision: 2

The following steps are followed for the design in MATLAB:

- (1) Determine the filter coefficients using Parks-McClellan FIR filter.
- (2) Convert the filter coefficients to finite precision coefficients.
- (3) Represent each coefficient in CSD using the algorithm in [11].

The Parks-McClellan algorithm is optimal in min-max sense, however, suffers from pass band ripples. For applications that require zero ripple in the pass band and small transition bandwidth, maximally flat FIR filters can be used which have zero ripple in the pass band [12]. The coefficient precision can be varied till

**Table 1 CSD representation of filter coefficients.**

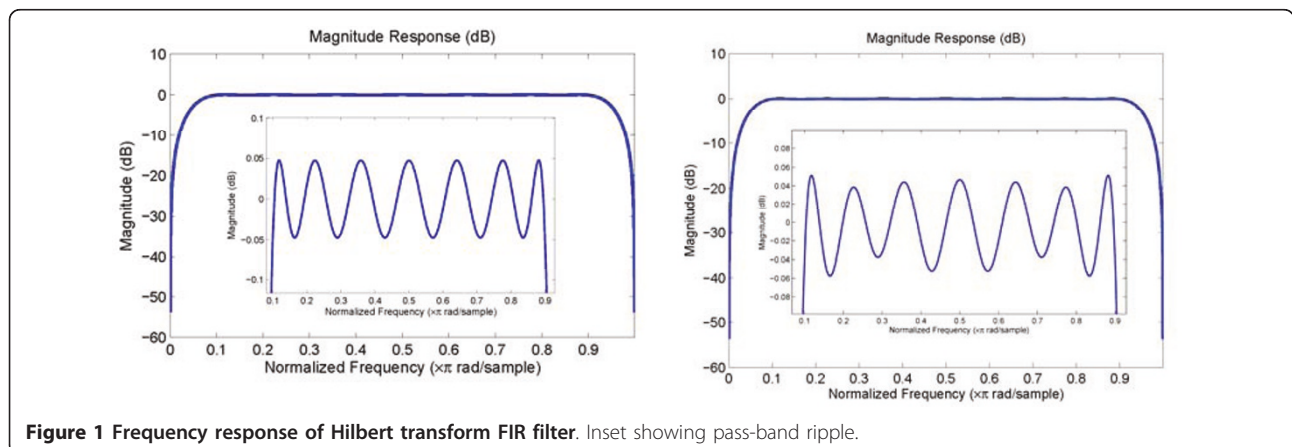
Coefficient	Value	CSD representation
$h(0) = -h(26)$	-0.00774850	$-2^{-7} + 2^{-12}$
$h(1) = -h(25)$	0	0
$h(2) = -h(24)$	-0.015821939	$-2^{-6}$
$h(3) = -h(23)$	0	0
$h(4) = -h(22)$	-0.031142897	$-2^{-5}-2^{-12}$
$h(5) = -h(21)$	0	0
$h(6) = -h(20)$	-0.056424022	$-2^{-4} + 2^{-7}-2^{-9}$
$h(7) = -h(19)$	0	0
$h(8) = -h(18)$	-0.100431833	$-2^{-3} + 2^{-5}-2^{-7} + 2^{-10} + 2^{-12}$
$h(9) = -h(17)$	0	0
$h(10) = -h(16)$	-0.195105144	$-2^{-2} + 2^{-4}-2^{-7} + 2^{-12}$
$h(11) = -h(15)$	0	0
$h(12) = -h(14)$	-0.630749788	$-2^{-1}-2^{-3}-2^{-7} + 2^{-9} + 2^{-12}$
$h(13)$	0	0

the response is close to the infinite precision frequency response. The number of taps as well as the digits in the CSD is kept minimum for efficient hardware implementation. The number of taps is varied till the desired response is obtained. The number of filter taps after optimization is 27 and the CSD coefficient precision is 12 bits. Table 1 gives the CSD coefficients generated from the algorithm in [11] from MATLAB. The frequency response of the Hilbert transform with infinite precision coefficients and CSD coefficients are shown in Figure 1 and they agree with each other.

**Hardware architecture**

The proposed hardware architecture is shown in Figure 2. The architecture consists of two basic blocks: coefficient add/sub and tap adder. The delays are modeled as Flip-Flop.

Direct Form II filter architecture is chosen for implementation. Negative powers of two is equivalent to division by powers of two and can be implemented by



**Figure 1** Frequency response of Hilbert transform FIR filter. Inset showing pass-band ripple.

simple bit shifting of the input signal  $X(n)$  to the right for a specific number of positions to get the various shifted signals which forms the input to the coefficient add/sub block. In Figure 2, the bit shifts are denoted by “>>” symbols and the input  $X(n)$  is represented in 14 bits. Making use of the symmetric nature of the coefficients,  $h(14)-h(26)$  are realized using adders/subtractors that make up the coefficient add/sub block and the coefficients  $h(0)-h(12)$  are obtained by multiplying by -1. This reduces the number of multiplications (shifters and

adders/subtractors) to half. There must be at least one positive digit in the CSD for multiplication implementation and hence  $h(14)-h(26)$  is implemented instead of  $h(0)-h(13)$ . The positive coefficients  $h(14)-h(26)$  will have at least one positive digit in the CSD representation. On the other hand, the negative coefficients may not have one positive digit in the representation like  $h(2)$  and  $h(4)$ . Such coefficients require subtraction from zero which is avoided here. The conventional way of implementing multiplication by -1 is by complementing and

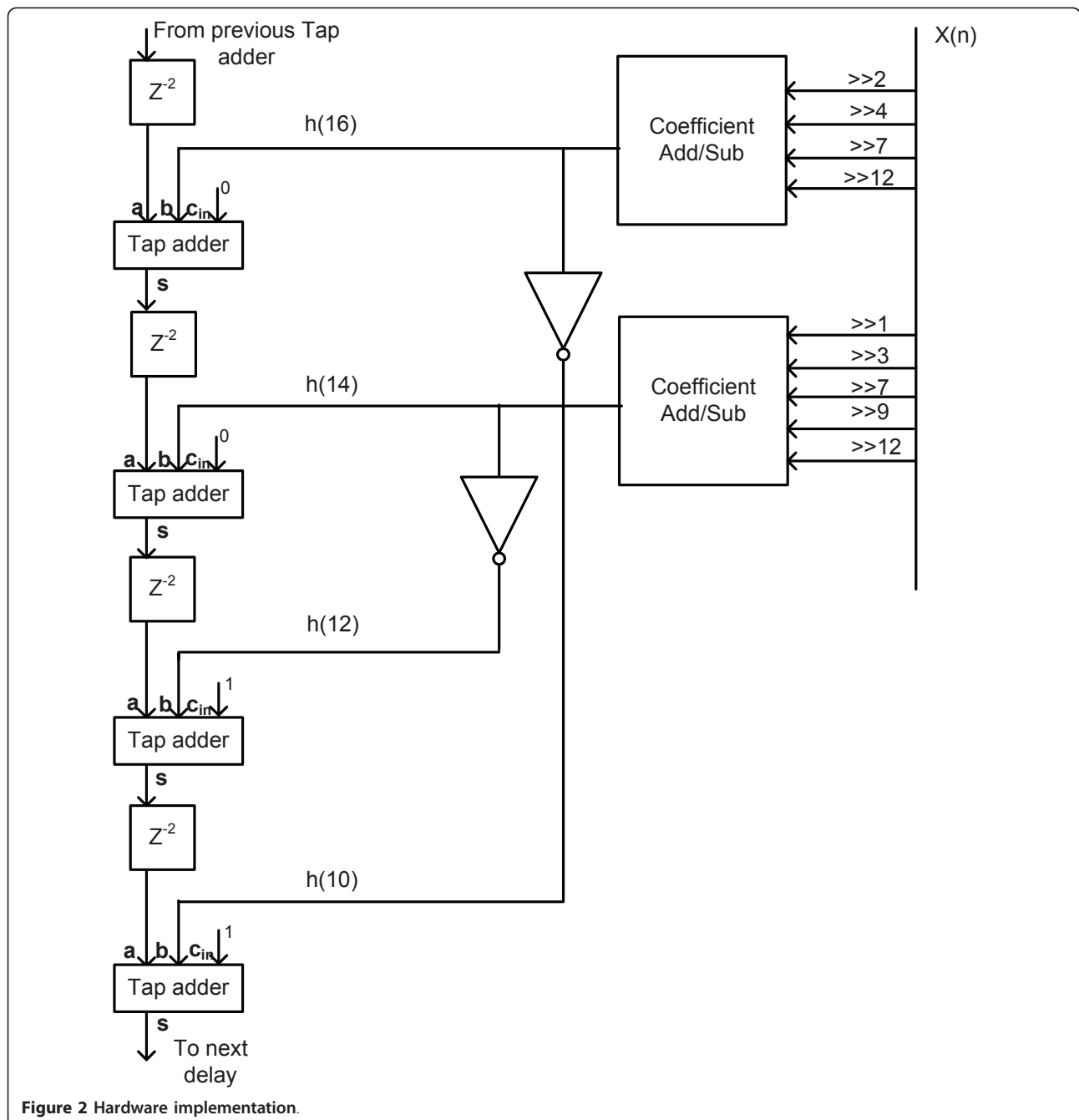


Figure 2 Hardware implementation.

**Table 2 Hardware comparison.**

Hardware	Convectional CSD	Proposed CSD
Tap adder	13	13
Delay (Flip-Flop)	26	26
Inverter (NOT gate)	7	7
Adder/subtractor	24	15

adding one. Here, we use a NOT gate for complementing and the carry of the tap adder is set instead of using an additional adder for adding one as shown in Figure 2. The setting of the carry in the tap adder eliminates the need for seven additional adders that are needed to implement the negative coefficients. One adder is saved for each negative coefficient implementation and this reduces the hardware cost compared to the convectional implementation of the filter with CSD coefficients. A comparison of the hardware cost for the convectional CSD and the proposed CSD implementation is shown in Table 2.

**FPGA implementation**

The complete filter hardware was described in VHDL and synthesized using Xilinx ISE 9.2i. The architecture has a maximum speed of 144 MHz. In order to use it for digital image rejection of demodulated RF signal, the proposed architecture is implemented in Virtex-4 FPGA (XC4VFX20-10FFG672C). For this, a board with the mentioned FPGA and a USB controller is used. The input to the FPGA is fed from 14-bit A/D converter which is clocked at 140 MHz. In order to test the performance of the architecture, it was tested with sinusoidal inputs of three different frequencies. A 7, 13, and 20 MHz sine input was given at the input of the A/D

converter. The phase shifts obtained are 90.7°, 89.98°, and 89.73°, respectively. The data from FPGA are captured in the PC and visualized in MATLAB. The phase shifted output for a 13-MHz sinusoidal input is shown in Figure 3.

**Conclusion**

The Hilbert transformer design with CSD coefficients for determining the impulse response is presented. An efficient architecture for implementing the filter in FPGA is presented. By using the symmetric nature of the filter coefficients, the hardware cost is reduced for the negative filter coefficient implementation. In the propose architecture, negative coefficient realization eliminates the need for additional seven adders compared to the conventional method of CSD FIR filter implementation.

**List of abbreviations**

CSD: canonic signed digit; FIR: finite impulse response; IIR: infinite impulse response.

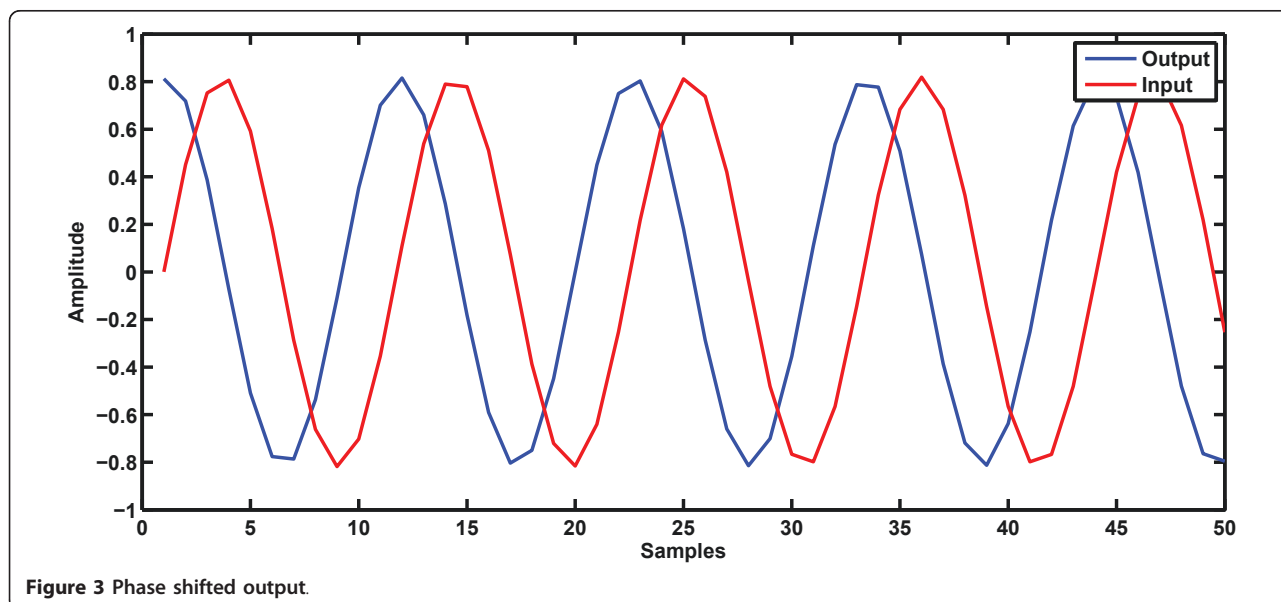
**Competing interests**

The authors declare that they have no competing interests.

Received: 7 March 2011 Accepted: 28 July 2011 Published: 28 July 2011

**References**

1. W Yu, J-m Lai, A fully digital DLLs integrated in FPGAs, in *9th International Conference on Solid-State and Integrated-Circuit Technology (ICSICT)*, 3 (2008)
2. A Asoodeh, M Atarodi, A 6-bit active digital phase shifter. *IEICE Electron Express*. **8**(3), 121–128 (2011). doi:10.1587/elex.8.121
3. D Viveiros Jr, D Consonni, AK Jastrzebski, A tunable all-pass MMIC active phase shifter. *Microwave Theory Tech IEEE Trans*. **50**(8), 1885–1889 (2002). doi:10.1109/TMTT.2002.801315



**Figure 3** Phase shifted output.

4. MS Naghmash, MF Ain, CY Hui, FPGA implementation of software defined radio model based 16QAM. *Eur J Sci Res.* **35**(2), 301 (2009)
5. H Shekhar, CB Mahto, N Vasudevan, FPGA implementation of tunable FFT for SDR receiver. *IJCSNS Int J Comput Sci Netw Secur.* **9**(5), 186–190 (2009)
6. C Thomos, G Kalivas, FPGA-based architecture of a DS-UWB Channel Estimator and RAKE Receiver employing a hybrid selection scheme, in *2010 IEEE 17th International Conference on Telecommunications (ICT)*, 903–909 (April 2010)
7. RN Gorgui-Naguib, SS Dlay, Hardware implementation of a Hilbert transformer, in *Proceedings of IEEE Mediterranean Electrotechnical Conference (MELECON'91)*, 404–407, (May 1991)
8. S He, M Torkelson, FPGA implementation of FIR filters using pipelined bit-serial canonical signed digit multipliers, in *Proceedings of IEEE Custom Integrated Circuits Conference (CICC'94)*, 81–84, (May 1994)
9. Y Takahashi, T Kitajima, K Takahashi, Hilbert transformer design using CSD FIR filter, in *Proceedings of 2001 International Technical Conference on Circuits/Systems Computers and Communications (ITC-CSCC 2001)*, pp. 921–924, (July 2001)
10. JGRC Gomes, 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.* **49**(3), 177–186 (2002). doi:10.1109/TCSII.2002.1013864
11. H Samuelli, An improved search algorithm for the design of multiplier less FIR filters with powers—of- two coefficients. *IEEE Trans Circuits Syst.* **36**, 1044–1047 (1989). doi:10.1109/31.31347
12. T Cooklev, A Nishihara, Maximally flat Hilbert transformers. *Int J Circuit Theory Appl.* **21**, 563–570 (1993). doi:10.1002/cta.4490210607

doi:10.1186/1687-6180-2011-32

**Cite this article as:** Duraiswamy et al.: Efficient implementation of 90° phase shifter in FPGA. *EURASIP Journal on Advances in Signal Processing* 2011 **2011**:32.

**Submit your manuscript to a SpringerOpen® journal and benefit from:**

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)

---