

Hardware Implementation of Canonic Signed Digit Recoding

B.V.V.Kiran, Kumar, B.Kasi babu, M.Sri Vijay

Abstract: This paper presents the implementation of convert a two's complement binary number into its canonic signed digit representation. In these CSD recoding circuits are functionally equivalent carriers H and K are described. They are computed in parallel reducing the critical path. They possess some properties that led to a simplification of the algebraic expressions minimizing the overall hardware implementation. When compared with other binary CSD with other architecture, new CSD recoding is more efficient. Power analysis and area estimation are performed for the designed CSDs.

Keywords: canonic Signed Digit (CSD)

I. Introduction

The canonic signed digit (CSD) representation is one of the existing signed digit representation. It is useful in certain DSP applications focusing on low power, efficient area and high speed arithmetic [1]. The CSD code is a ternary number with the digit set {1, 0, -1}. The corresponding representation is unique and it has two properties. 1. The number of nonzero digits is minimal, and 2. No two consecutive digits are both nonzero, i.e; two nonzero digits are not adjacent.

CSD representation has proven to be useful for the design and implementation of digital filters such as the area efficient programmable FIR digital filter architecture[2] and the low complexity algorithms for designing filters in [3], Two dimensional IIR and FIR filter design in[4]. CSD code has been largely exploited to implement efficient multipliers[5].

In 1960, Reitwiesener proposed an algorithm for converting two's complement number numbers to a minimum weight radix-2 signed digit representation [6]. All of these algorithms generate the CSD code recursively from the least significant bit (LSB) to most significant bit (MSB)[7]. In order to generate binary from CSD, the algorithm recursively generates from MSB to LSB[8]. CSD representation has no consecutive digits.

A novel efficient cell-based implementations to convert a two's complement binary number into its CSD representation based on two signals H and K functionally equivalent to two carriers is presented.

CSD Recoding

The CSD recoding of an integer number is a signed and unique representation that contains no adjacent nonzero digits. Given an n-digit binary number unsigned number

$$X = \sum_{i=0}^{n-1} x_i 2^i \quad x_i \in \{0,1\}$$

Then the (n+1) digit CSD representation Y of X is given by

$$y = \sum_{i=0}^{n-1} x_i 2^i = \sum_{i=0}^n y_i 2^i \quad y_i \in \{\bar{1}, 0, 1\}$$

This condition that all nonzero digits in a CSD number are separated by zeros implies that

$$y_{i+1} y_i = 0 \quad 0 \leq i \leq n-1$$

From this property, the probability that a CSD n-digit has a nonzero value is given by

$$P(|y_i| = 1) = \frac{1}{3} + \frac{1}{9} \left[1 - \left(\frac{-1}{2} \right)^n \right]$$

As n becomes large, the probability tends to 1/3 while this probability becomes 1/2 in a binary code, using this property, the number of additions/subtractions is reduced to a minimum number of multipliers and an overall speed-up can be achieved.

The adaptation of a ternary number system adds some flexibility to the CSD representation, it allows the number of nonzero digits to be minimized, but it requires that each digit y_i be encoded two over bits $\{y_i^d, y_i^s\}$. Since it satisfies the following relation.

$$y_i = y_i^d - y_i^s$$

Where y_i^d represents the data bit and y_i^s represents the sign bit. This encoding also allows an additional valid representation of 0 when $y_i^s = 1$ and $y_i^d = 1$ which is useful in some arithmetic implementations.

Hash main presented the binary coded CSD (BCSD) number to avoid extra data word representation. The BCSD recoding is based on a simple binary representation $B = \{b_0, b_1, \dots, b_{n-1}\}$ of a CSD code uses the same number of bits as its original two's complement representation. If the bit i in a CSD code is non zero $y_i \neq 0$, then the bit i is nonzero in the BCSD code, $b_i \neq 0$ and the following bit b_{i+1} acts as a sign bit; $b_{i+1} = 0$ means y_i is positive and $b_{i+1} = 1$ means y_i is negative. A simple conversion between a CSD recoding and BCSD coding is

$$b_i = y_i^d + y_{i+1}^s$$

$$b_{i+1} = y_{i+1}^d + y_i^s$$

The conversion from a binary representation to CSD representation is mostly based on the identity.

$$2^{i+j-1} + 2^{i+j-2} + \dots + 2^i = 2^{i+j} - 2^i$$

A conversion process from binary to CSD algorithm is shown.

$$\hat{a}_{-1} = 0$$

$$\gamma_{-1} = 0$$

$$\hat{a}_w = \hat{a}_{w-1}$$

for ($i = 0$ to $w - 1$)

$$\{\theta_i = \hat{a}_i \wedge \hat{a}_{i-1}$$

$$\gamma_i = \overline{\gamma_{i-1}} \theta_i$$

$$a_i = (1 - 2\hat{a}_{i+1})\gamma_i\}$$

The encoding is performed from LSB to MSB using two adjacent digits and a carry signal. Here, the carry-out $c_i = 1$ if and only if there is two of three 1s among the three inputs x_{i+1}, x_i and c_{i-1}

$$c_i = x_i x_{i+1} + (x_i + x_{i+1})c_{i-1} \quad c_{-1} = 0$$

The variable $D = \{d_0, d_1, \dots, d_{n-1}\}$ in which all nonzero digits in the CSD representation is defined as

$$d_i = x_i \wedge c_{i-1}$$

Since y_i takes one of the three values $\{0, 1, \bar{1}\}$, it is necessary to encode it.

$$y_i^d = \overline{x_{i+1}}(x_i \wedge c_{i-1}) = \overline{x_{i+1}}d_i$$

$$y_i^s = x_{i+1}(x_i \wedge c_{i-1}) = x_{i+1}d_i$$

In the case of an n-bit two's complement binary number X; the CSD representation is given by

$$Y = -X_{n-1}2^{n-1} + \sum_{i=0}^{n-2} x_i 2^i = \sum_{i=0}^{n-1} y_i 2^i$$

Here, only n CSD digits are necessary as the value of the binary number is limited to $[-2^n, 2^{n-1}]$. Negative integers in CSD can be obtained from their positive counterpart by changing the signs of all nonzero digits.

To convert a binary number into its CSD representation, valid for both unsigned and two's complemented binary numbers. The only difference arises in the last CSD digit. An extra sign extension, $x_n = 0$ for unsigned numbers and $x_n = x_{n-1}$ for two's complement numbers, the last section changes depending on the sign of X in the following general expression.

$$\text{For an unsigned number } X \begin{cases} y_{n-1}^d = d_{n-1} \\ y_{n-1}^s = 0 \\ y_n^d = d_n = c_{n-1} \\ \quad x_n c_{n-2} \\ y_n^s = 0 \end{cases}$$

$$\text{For a signed number } X \begin{cases} y_{n-1}^d = \overline{x_n} d_i = \overline{x_{n-1}} c_{n-2} \\ y_{n-1}^s = x_n d_i = x_{n-1} \overline{c_{n-2}} \end{cases}$$

The simulated waveform for the conversion of binary number to its CSD representation (n=6) is shown in Figure 1 and the RTL synthesis from RC is shown in Figure 2.



Figure1. Simulated Waveform for conversion of binary number to CSD representation (n=6)

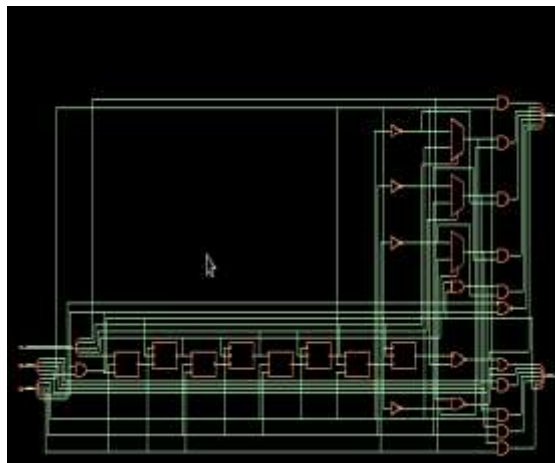


Figure 2. RTL compiler for binary to CSD code representation (n=6).

New Csd Recoding

Let X is an n-digit binary number. If we denote the two signals, the signals are $H = \{h_0, h_1, \dots, h_{n-1}\}$ and $K = \{k_0, k_1, \dots, k_{n-1}\}$ as

$$h_i = \begin{cases} x_i h_{i-1} & \text{for } i \text{ odd} \\ x_i + h_{i-1} & \text{for } i \text{ even} \end{cases}$$

$$k_i = \begin{cases} x_i + k_{i-1} & \text{for } i \text{ odd} \\ x_i k_{i-1} & \text{for } i \text{ even} \end{cases}$$

With $h_{-1} = 0, k_{-1} = 0$. Then c_i can be formally expressed in terms of odd or even index i means the following relation.

$$c_i = \begin{cases} h_i + x_{i+1} k_i & \text{for } i \text{ odd} \\ x_{i+1} h_i + k_i & \text{for } i \text{ even} \end{cases}$$

The two signals H and K have some properties.

$$\text{For } i \text{ odd } \begin{cases} h_i k_i = h_i \\ h_i \bar{k}_i = 0 \\ \bar{h}_i + k_i = 1 \\ h_i + k_i = k_i \end{cases}$$

$$\text{For } i \text{ even } \begin{cases} h_i k_i = k_i \\ k_i \bar{h}_i = 0 \\ \bar{k}_i + h_i = 1 \\ h_i + k_i = h_i \end{cases}$$

The variable D can be obtained from H and K following.

$$\text{For } i \text{ even } d_i = \overline{h_i k_i}$$

$$\text{For } i \text{ odd } d_i = \overline{h_i k_i}$$

However, the variable Y for the CSD recoding in terms of signals H and K can be expressed as

$$y_i^d = d_i \overline{x_{i+1}} = \begin{cases} \overline{k_i h_{i+1}} & \text{for } i \text{ odd, even} \\ \overline{h_i k_{i+1}} \end{cases}$$

$$y_i^s = d_i x_{i+1} = \begin{cases} \overline{k_{i+1} h_i} & \text{for } i \text{ odd, even} \\ \overline{h_{i+1} k_i} \end{cases}$$

Only the last bit changed depending on sign of X in the following general expression

For a signed number

$$y_{n-1}^d = \begin{cases} \overline{k_{n-1} h_{n-2}} & \text{for } n - 1 \text{ odd, even} \\ \overline{h_{n-1} k_{n-2}} \end{cases}$$

$$y_{n-1}^s = \begin{cases} \overline{k_{n-2} h_{n-1}} & \text{for } n - 1 \text{ odd, even} \\ \overline{h_{n-2} k_{n-1}} \end{cases}$$

For an unsigned number

$$y_{n-1}^d = \begin{cases} k_{n-1} \overline{h_{n-1}} & \text{for } n - 1 \text{ odd} \\ h_{n-1} k_{n-1} & \text{for } n - 1 \text{ even} \end{cases}$$

$$y_{n-1}^s = \begin{cases} h_{n-1} & \text{for } n - 1 \text{ odd} \\ k_{n-1} & \text{for } n - 1 \text{ even} \end{cases}$$

The simulated waveform for binary to CSD new recoding (n=6) is shown in Figure 3. The RTL synthesis from RC is shown in Figure 4.

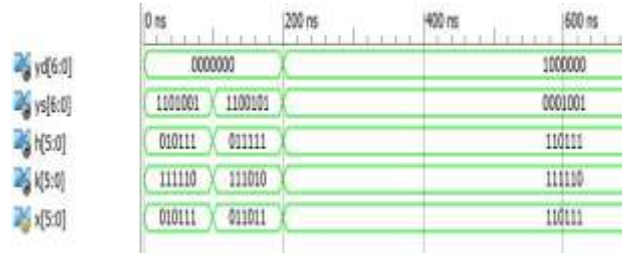


Figure 3. Simulated Waveform for new CSD recoding (n=6)

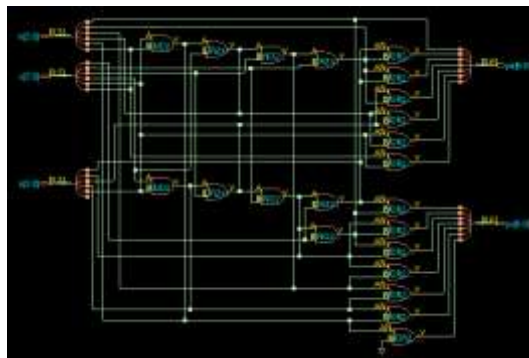


Figure 4. RC compiler for new CSD recoding (n=6)

The graphs of the Power, Area, Gates of the Binary CSD and New Binary CSD is shown in the Figure 5, Figure 6 and Figure 7. From Figure 5, it is evident that Binary CSD consumes 12.69174 pW power, New Binary CSD consumes 7.828716 pW. From Figure 6, it is evident that Binary CSD occupies 426 μm^2 , New Binary CSD occupies 293 μm^2 . From Figure 7, it is evident that Binary CSD gates are 30 and New Binary CSD gates are 22.

It is inferred that there is 38.31% reduction of power, 26.67% reduction of gates and 31.22% reduction of area for New Binary CSD compared with Binary CSD.

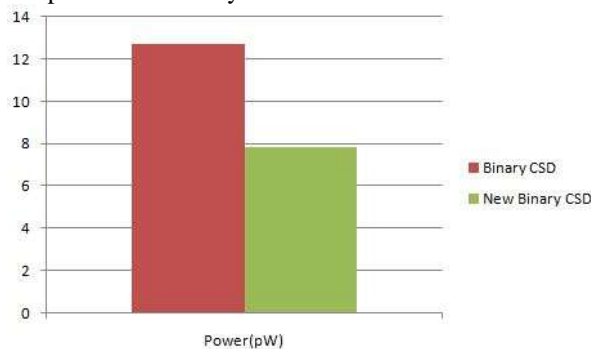


Figure 5. Power for Binary CSD and New Binary CSD (n=6)

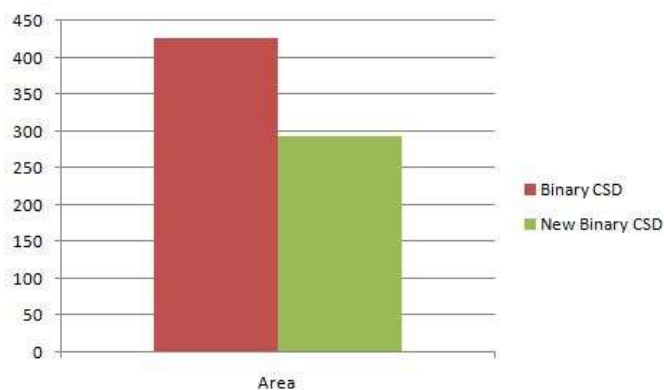


Figure 6. Area for Binary CSD and New Binary CSD (n=6)

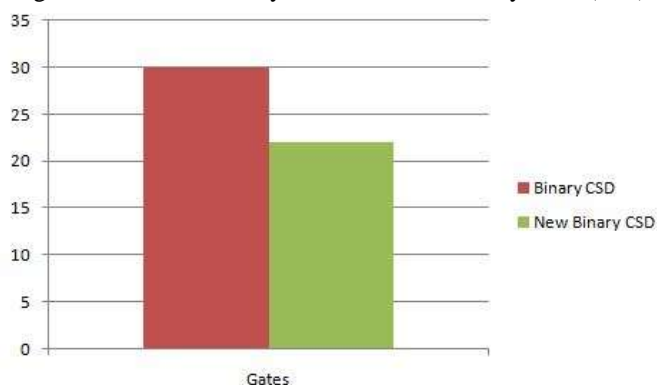


Figure 7. Gates for Binary CSD and New Binary CSD (n=6)

II. Conclusion

The conversion of a two's complement number into its canonic signed digit(CSD) representation can be efficiently implemented using two signals H and K., the New CSD recoding is better than the binary to CSD recoding because it reduces power, area and gates.

References

- [1] I. Koren, Computer Arithmetic Algorithms (Second ed.), A.K. Peters, Ltd. (Ed.), 2002.
- [2] B. Phillips, N. Burgess, Minimal weight digit set conversions, IEEE Transactions on Computers 53 (6) (2004) 666–677.
- [3] K.Y. Khoo, A. Kwentus, A.N. Wilson, A programmable FIR digital filter using CSD coefficients, IEEE Journal of Solid-State Circuits 31 (6) (June 1996) 869–874.
- [4] J. Skaf, S.P. Boyd, Filter design with low complexity coefficients, IEEE Transactions on Signal Processing 56 (7) (2008) 3162–3169.
- [5] T. Williams, M. Ahmadi, W.C. Miller, Design of 2D FIR and IIR digital filters with canonical signed digit coefficients using singular value decomposition and genetic algorithms, Circuits Systems Signal Processing 26 (1) (2007) 69–89.
- [6] G.K. Ma, F.J. Taylor, Multiplier policies for digital signal processing, IEEE ASSP Magazine 1 (1990) 6–20.
- [7] M.A. Soderstrand, CSD multipliers for FPGA DSP applications, in: Proceedings of the International Symposium on Circuits and Systems (ISCAS), pp. V-469–V-472, 2003.
- [8] D.L. Iacono and M. Ronchi, Binary canonic signed digit multiplier for high- speed digital signal processing, in: Proceedings of the 47th IEEE International Midwest Symposium on Circuits and Systems, II, pp. 205–208, 2004.