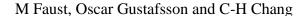
Linköping University Post Print

Fast and VLSI efficient binary-to-CSD encoder using bypass signal



N.B.: When citing this work, cite the original article.

Original Publication:

M Faust, Oscar Gustafsson and C-H Chang, Fast and VLSI efficient binary-to-CSD encoder using bypass signal, 2011, ELECTRONICS LETTERS, (47), 1, 18-19. http://dx.doi.org/10.1049/el.2010.3055

Copyright: Iet

http://www.theiet.org/

Postprint available at: Linköping University Electronic Press http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-65719

Fast and VLSI efficient binary-to-CSD encoder using bypass signal

Mathias Faust, Oscar Gustafsson, and Chip-Hong Chang

The generation of a Canonical Signed Digit (CSD) representation from a binary representation is revisited. Based on the property that each nonzero digit is surrounded by a zero digit, a hardware-efficient conversion method using bypass instead of carry propagation is proposed. The proposed method requires less area per digit and the required bypass signal can be generated or propagated with only a single NOR gate. It is shown that the proposed converter outperforms previous converters and a look-ahead circuitry to speed up the generation of bypass signals is also proposed.

Introduction: Canonical Signed Digit (CSD) representation [1] has been used for digital arithmetic operations to reduce the average and maximum numbers of nonzero digits to n/3 + 1/9 and $\lceil \frac{n+1}{2} \rceil$, respectively for an n-bit binary encoded number. In CSD no two adjacent digits can both be nonzero, which helps to reduce the number of additions required in constant and variable multiplications. Exponentiation in cryptography can also benefit from the use of CSD representation [2].

The requirement that the nonzero digits must not be adjacent makes CSD representation unique and differentiates it from the non unique Minimal Signed Digit (MSD) representation [3] which also has minimal Hamming weight. One drawback of representing a variable in CSD is that the conversion from binary to CSD can only be generated recursively, whether it is determined from the Least Significant Bit (LSB) to the Most Significant Bit (MSB), i.e., right-to-left or from the MSB to the LSB, i.e., left-to-right. Most left-to-right conversion [2], [4] and bidirectional conversion [5] methods generate a MSD encoded output as the least significant 1 can influence all digits to its left during the conversion to CSD. Although conversion to other Signed Digit (SD) representations can be executed in parallel, they do not have the minimum number of nonzero digits and could even have more nonzero digits than binary representation.

Conventional approaches [1], [6], [7] to the binary to CSD transformation use an intermediate binary carry signal. These approaches do not take advantage of the properties that $x_i \times x_{i-1} = 0$, where x_i is the i-th digit of a CSD number. In [2] a left-to-right sliding window method is proposed to convert a binary number into MSD. The sliding window scans three consecutive bits at a time and produces one or two output digits. It produces

two output digits whenever one of the outputs is 1 or -1. The simplicity of the sliding window algorithm in [2] triggered a search for a similar hardware implementation and lead to the proposed method, although only similar in that it uses the fact that $x_i \times x_{i-1} = 0$.

This letter introduces a new bypass technique for binary to CSD conversion amenable to standard cell based implementation. The proposed converter uses less logic elements than the existing carry propagating converters in ASIC or FPGA. A bypass signal is output to the next digit slice to bypass the evaluation of the inputs in the next slice if a nonzero digit is output in the current slice. This bypass signal can be generated by a single NOR gate in each digit slice. The bypass chain can be further sped up by a look-ahead circuit. The design is simple and scalable as n digit-slices can be cascaded to generate an n-digit CSD.

Processing element of Binary-to-CSD converter: In our proposed method, the two's complement number to be converted is partitioned into overlapping groups of three bits. Each digit slice of the converter recodes three binary bits, b_{i+1} , b_i and b_{i-1} into a CSD digit, x_i which is binary encoded into a sign bit, $x_{i,s}$, and a magnitude bit, $x_{i,m}$. Under the sign-magnitude encoding (equivalent to two's complement) $\{0 \equiv 00, 1 \equiv 01, -1 \equiv 11\}$. The truth table for the Processing Element (PE) of the i-th digit slice is shown in Table I, where p_i and p_{i+1} are the bypass input into and output from the PE, respectively. Without the bypass signals, redundant nonzero digits will be generated when the PEs of all digit-slices execute concurrently. For example, the binary number $[0101011]_2$ will be recoded into $[11\bar{1}1\bar{1}0\bar{1}]$ in the absence of the bypass signals while the correct CSD output based on a recursive right-to-left application of the CSD identities to make $x_i \times x_{i-1} = 0$ yields $[10\bar{1}0\bar{1}0\bar{1}]$. In this example two extraneously generated nonzero digits exist at bit positions i=3,5 (where i=0 for the LSB). This breach of the non-adjacency criterion of CSD is detected and corrected after one gate delay by cascading the PEs using the bypass signals.

When $p_i=0$, the recoded digit x_i is generated from the input bits b_{i+1} , b_i and b_{i-1} according to the first eight rows of Table I, and a bypass signal is generated for the next PE, $p_{i+1}=|x_i|$, which can be directly hardwired from $x_{i,m}$. The magnitude bit is dependent only on b_i and b_{i-1} , and $x_{i,m}=1$ iff $b_i=1$ or $b_{i-1}=1$. The sign bit is dependent on b_{i+1} and $|x_i|$, and $x_{i,s}=1$ iff $b_{i+1}=|x_i|=1$. When $p_i=1$, the outputs x_i and p_{i+1} are forced to zero irrespective of the other inputs as depicted by the don't care inputs, d, in the last row of Table I. The inputs to the PE are said to be bypassed or ignored by the PE, while setting p_{i+1} to zero allows a new bypass signal to be generated at the next PE. The PE circuit that realises Table I is very simple, as shown in Fig. 1. Comparing the proposed PE with the bit-parallel implementation in Fig. 5 of [4] the simplicity

of the circuit is distinctive. The bypass signal can be generated or propagated with only a NOR gate delay as opposed to the carry propagation delay of at least two cascaded two-input gates. The same bypass technique is also applicable to the CSD digit encoded in negative-positive binary code $\{0 \equiv 00, 1 \equiv 01, -1 \equiv 10\}$ but additional logic gates will be required.

Input paddings are required for the two end PEs at the least and most significant digit positions i.e., i=0 and i=n-1. The inputs b_{-1} and p_0 to the rightmost PE are set to zero. For the leftmost PE, if the input is signed and represented in two's complement form, the MSB b_{n-1} is sign extended to b_n . If the input is unsigned, then an additional PE will be needed to generate x_n to cover the largest positive number and the inputs b_n and b_{n+1} are set to 0. In both cases, p_{i+1} of the leftmost PE is discarded.

Lookahead circuit for bypass generation: The bypass signal p_{i+1} depends on the output of the XOR, $v_i = b_i \oplus b_{i-1}$, and the bypass signal p_i by the relation $p_{i+1} = v_i \overline{p_i}$. This can be unfolded and rearranged as it is done for the carry in [7] or carry look-ahead adder circuits. The unfolded bypass signal is given by

$$p_i = \sum_{j=0}^{i/2} \overline{v_{2j}} \prod_{k=j}^{i/2} v_{2k+1} \tag{1}$$

$$p_i = \prod_{m=0}^{i/2} v_{2m} + \sum_{j=1}^{i/2} \overline{v_{2j-1}} \prod_{k=j}^{i/2} v_{2k}$$
 (2)

where (1) and (2) are used for the even and odd i, respectively. The Boolean sum of products can be implemented as a balanced tree. Therefore, the bypass signal p_i can be calculated with the delay of $O(\log(n))$. As b_i is a common input in v_i and v_{i+1} , the following simplification is possible $\overline{v_i}v_{i+1} = \overline{b_{i-1}}\,\overline{b_i}\,b_{i+1} + b_{i-1}\,b_i\,\overline{b_{i+1}}$, but as v_i is required anyway, the simplification requires more area and the speedup is marginal.

Results and comparison: Comparison of the proposed method and a general carry propagation method for different input bit widths is shown in Table II for ASIC implementations. The results were generated from VHDL code compiled by Synopsys Design Compiler v2010.03-SP3 with the TSMC $0.18~\mu m$ standard cell library.

The carry based PE is faster than the bypass based PE due to the longer delay from the binary inputs to the digit output, but consumes about twice the area. However, for n=16 to 256, the proposed converter has approximately 40% shorter delay than the general method due to the shorter delay of the bypass signal and uses less than 50% of its area. For n=8, the savings are higher for delay and lower for area. With an average delay of approximately 0.053 ns per digit, the conversion is very fast and this opens up new opportunities to use CSD instead of Booth encoding to reduce the switching activity due to fewer nonzero digits in multiplication.

The VHDL code was also compiled by Xilinx ISE targeting Virtex 4 FPGA. The results showed that the PE of the proposed method fits into one slice using only two LUTs, while the general implementation requires three LUTs. For bit widths above 16 the number of required LUTs and required slices is reduces by more than 35%. *Conclusion*: In this letter, a new method for binary to CSD conversion using a bypass signal is proposed.

The method outperforms other carry propagation methods. It is also shown that the generation of the bypass signal to each PE can be further sped up by using relatively simple look-ahead logic.

References

- [1] Hwang K. Computer Arithmetic: Principles, Architecture and Design. John Wiley & Sons, Inc.; 1979.
- [2] Okeya K, Schmidt-Samoa K, Spahn C, Takagi T. Signed binary representations revisited. In: Advances in Cryptology. vol. 3152 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg; 2004. p. 119–142.
- [3] Phillips B, Burgess N. Minimal weight digit set conversions. IEEE Transactions on Computers. 2004 Jun;53(6):666-677.
- [4] Lim YC, Evans JB, Liu B. Decomposition of binary integers into signed power-of-two terms. IEEE Transactions on Circuits and Systems. 1991 Jun;38(6):667–672.
- [5] Backenius E, Säll E, Gustafsson O. Bidirectional conversion to minimum signed-digit representation. In: Proc. IEEE Int. Symp. Circuits Syst. Island of Kos, Greece; 2006. p. 2413–2416.
- [6] Herrfeld A, Hentschke S. Look-ahead circuit for CSD-code carry determination. Electronics Letters. 1995;31(6):434-435.
- [7] Das SK, Pinotti MC. Fast VLSI circuits for CSD coding and GNAF coding. Electronics Letters. 1996;32(7):632-634.

Authors' affiliations:

- M. Faust and C. H. Chang (Centre for High Performance Embedded Systems, Nanyang Technological University, Singapore. E-mail: {faus0001, echchang}@ntu.edu.sg)
- O. Gustafsson (Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden. E-mail: oscarg@isy.liu.se)

F	igure captions:	
1	Processing element for one digit of binary-to-CSD conversion	6

FIGURES 6

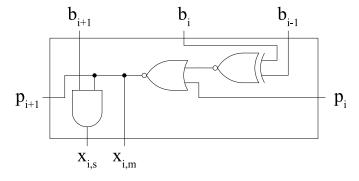


Fig. 1

FIGURES 7

-	Table captions:	
I	Truth table for CSD encoding with bypass logic	8
П	Delay optimized synthesis results based on CMOS $0.18~\mu m$ standard cell library	9

TABLES 8

TABLE I

p_i	b_{i+1}	b_i	b_{i-1}	x_i	$x_{i,s}$	$x_{i,m}$	p_{i+1}
	0	0	0	0	0	0	0
	0	0	1	1	0	1	1
	0	1	0	1	0	1	1
0	0	1	1	0	0	0	0
	1	0	0	0	0	0	0
	1	0	1	-1	1	1	1
	1	1	0	-1	1	1	1
	1	1	1	0	0	0	0
1	d	d	d	0	0	0	0

TABLES 9

TABLE II

	[1]		Prop	osed	Reduction	
Bits	$[\mu m^2]$	[ns]	$[\mu m^2]$	[ns]	Area	Delay
PE	236	0.11	136	0.15	42.3%	-40%
8	945	0.74	609	0.39	35.6%	47.3%
16	2235	1.46	1054	0.84	52.8%	42.9%
32	5212	2.83	1943	1.71	62.7%	39.7%
64	9790	5.64	4201	3.39	57.1%	39.9%
128	18714	11.32	8206	6.78	56.2%	40.1%
256	37725	22.55	16353	13.59	56.7%	39.7%