IMPLEMENTATION OF POWER EFFICIENT PARALLEL CHIEN SEARCH ARCHITECTURE USING A TWO-STEP APPROACH

Girish Pechetti[1], Bonam Vijaya Lakshmi[2], Patnala Hima Bindu[3]
[1,2,3]Department of Electronics and Communication Engineering,
Bonam Venkata Chalamayya Institute of Technology & Science,
Amalapuram, Andhra Pradesh, India.

ABSTRACT:
The short horizontal Bose caudari Hocquenghem (BCH) Chien search for signs of a new power-saving (CS) structure is proposed. For syndrome-based decoding, CS plays an important role in identifying the areas of error, but incurs a huge waste of exhaustive computation power consumption. The proposed architecture, the process of searching for the binary representation of the matrix is decomposed in two steps. This is neither new low power architecture for parallel CS provided. By reducing access to the second stage of the conventional CS to achieve significant power savings is decomposed in two steps. Error operate under the same ownership, the less energy the size of the CS in the construction sector in different configurations, and error correction capability of the horizontal factor compared to traditional construction. Power saving horizontal factor or increase the size of the field will become more and more important. Further this project is enhanced by replacing by multiplier architecture with radix-n modified booth multiplication algorithm for more power and area reduction. Radix4 modified booth encoding algorithm produces 50 percent reduction in partial different systems.

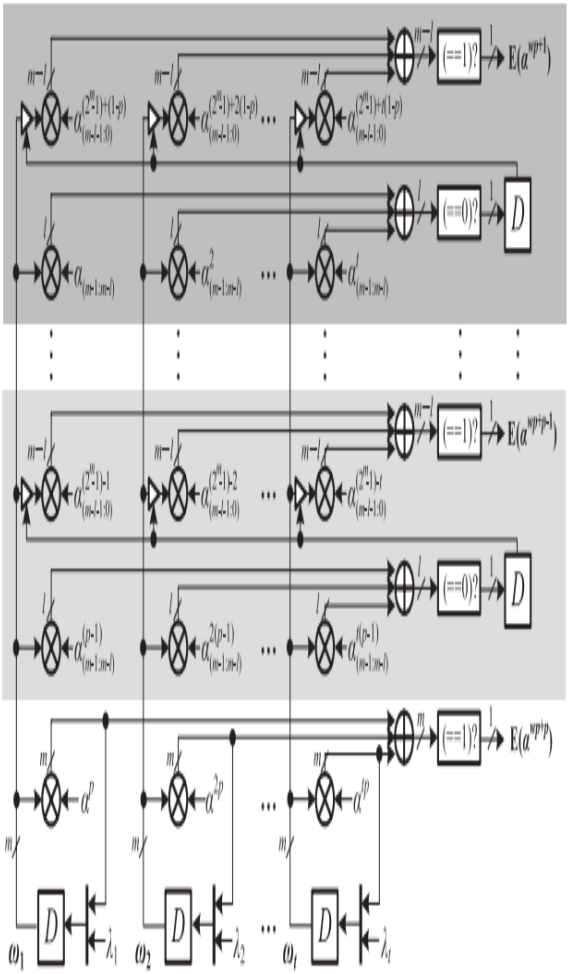
INTRODUCTION:
Satisfying the huge computational capacity of high throughput and strong error correction is inevitable, therefore, becomes more and more important power saving structure of the BCH decoding. In general, a BCH decoder to correct the bits T at the peak of the three main blocks, namely, the syndrome calculation (SC), the key-equation solving (KES) has, and Chien search (CS) [1], [2]. Receiving a code word for a given R (x) Compute syndromes SC 2T and KES (X) using the syndromes of the error locator polynomial Λ. Finally, the error is E (X) Λ sources (X) CS determined by the algorithm is based on the finding. In a parallel BCH decoder, CS main cause of power consumption and total electricity consumption [6] and can take up to a half. Numerous studies have demonstrated the ability to reduce the power consumption of CS proposed structures. Early termination of the methods presented in [6] and [7] After finding an error in the past to eliminate redundant computations are. An additional error counter is incremented when an error is found, and the counter KES downsides found in the CS is turned off matches. BCH decoder dealing with a small number of errors early in the implementation of the common and effective drug, though, when the power saving small insignificant error correction capability. [8], is a more effective method in order polynomial reduction (POR) when the error was found in the error locator polynomial of the proposed reform. Locator polynomial order one at a time, errors are detected by the decline and eventually becomes zero. POR [8] at a time, gradually power down circuitry associated with a polynomial factor makes it impossible for the CS. POR for serial BCH decoders are successful, however, because it is difficult to apply the technique of complex polynomial update parallel architecture. Furthermore, all of the previous power saving algorithms, including early termination, [6], [7] and the POR [8], depending on the position of the errors. For example, if faults at the end of the term of the code, as in the case of power savings is significant that in the beginning of errors. In this brief, we have a new approach, which is parallel to the CS proposed two stages of decomposition. In order to have access to each of the first step, but the first step to access the second stage will be activated only when a less successful result. The proposed two-step method [9] that is conceptually similar. The two-step approach, in general, lead to an increase in the critical path delay and delay, the losses can be solved simply by employing an efficient pipelined architecture. Unlike previous architectures [6] - [8], regardless of the error, the location of the proposed construction of the power consumption can be saved

TWO-STEP CS ARCHITECTURE:

As indicated in above, the p-parallel CS examines p error positions simultaneously, each of which generates a 1 × m binary matrix denoting a Galois field (GF) element by computing

$$Y(\alpha^{wp+i}) = \sum_{j=1}^{t} \mathrm{FFM}_{ij} = \sum_{j=1}^{t} \Omega_j A_{ij} = [\Omega_1\ \Omega_2 \cdots \Omega_t] \begin{bmatrix} A_{i1} \\ A_{i2} \\ \vdots \\ A_{it} \end{bmatrix}$$

Where i ranges from 1 to p. The CS determines the presence of an error when Y (αwp+i) is 1, which implies that αwp+i is a root of the error locator polynomial. In the GF of dimension m, the multiplicative identity element, α0 or α2m−1, is defined as 1, i.e., 0(m−1:1)1(0), more precisely. The main idea comes from the fact that the absence of errors is guaranteed if some bits ofY (αwp+i) are not equal to those of 0(m−1:1)1(0). In the case of GF(24), for example, no presence of errors is guaranteed if Y (αwp+i)(3:2) _= 0. Similar to [9], a two-step approach is employed for early detection.

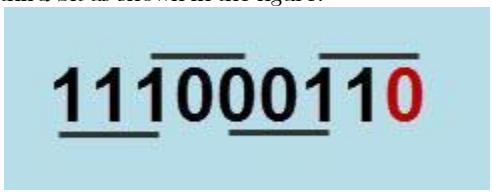


illustrates the low-power CS architecture based on the proposed two-step approach. According to (8), the m-bit FFMs in the conventional CS are replaced with the pipelined two partial FFMs except for those in the pth row. Given the intermediate values from the registers, the first partial FFM processes the l MSBs and activates the second partial FFM responsible for the remaining m − l LSBs at the next clock cycle only when the output of the former is 0. Otherwise, we can reduce the dynamic switching power by disabling the latter partial FFMs. Since each intermediate register can hold one of all possible GF elements, the latter partial FFM is activated once every 2l clock cycles on the average. Furthermore, it is worth noting that the sum of the hardware complexity for the former and the latter partial FFMs is almost the same as the conventional FFM. Therefore, additionally required in the proposed architecture are the p 1-bit registers and the (p − 1)t m-bit buffers.

Modified Booth Algorithm Encoder

This modified booth multiplier is used to perform high-speed multiplications using modified booth algorithm. This modified booth multiplier's computation time and the logarithm of the word length of operands are proportional to each other. We can reduce half the number of partial product. Radix-4 booth algorithm used here increases the speed of multiplier and reduces the area of multiplier circuit. In this algorithm, every second column is taken and multiplied by 0 or +1 or +2 or -1 or -2 instead of multiplying with 0 or 1 after shifting and adding of every column of the booth multiplier. Thus, half of the partial product can be reduced using this booth algorithm. Based on the multiplier bits, the process of encoding the multiplicand is performed by radix-4 booth encoder.

The overlapping is used for comparing three bits at a time. This grouping is started from least significant bit (LSB), in which only two bits of the booth multiplier are used by the first block and a zero is assumed as third bit as shown in the figure.



The figure shows the functional operation of the radix-4 booth encoder that consists of eight different types of states. The outcomes or multiplication of multiplicand with 0, -1, and -2 are consecutively obtained during these eight states.

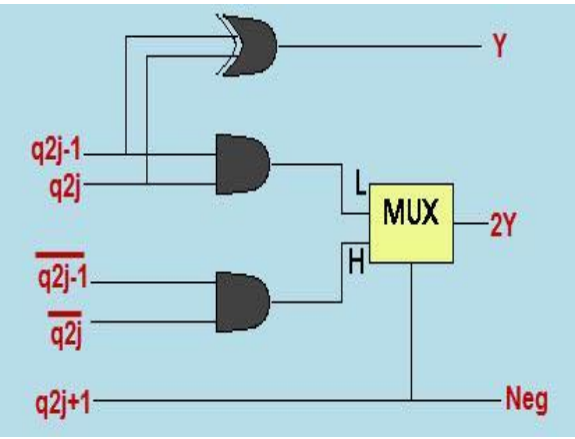
Booth recoding table for radix-4

Multiplier Bits Block			Recoded 1-bit pair		2 bit booth	
i+1	i	i-1	i+1	i	Multiplier Value	Partial Product
0	0	0	0	0	0	Mx0
0	0	1	0	1	1	Mx1
0	1	0	1	-1	1	Mx1
0	1	0	1	0	2	Mx2
1	0	0	-1	0	-2	Mx-2
1	0	1	-1	1	-1	Mx-1
1	1	0	0	-1	-1	Mx-1
1	1	0	0	0	0	Mx0

Booth Recoding

The steps given below represent the radix-4 booth algorithm:

- Extend the sign bit 1 position if necessary to ensure that n is even.
- Append a 0 to the right of the least significant bit of the booth multiplier.
- According to the value of each vector, each partial product will be 0, +y, -y, +2y or -2y.



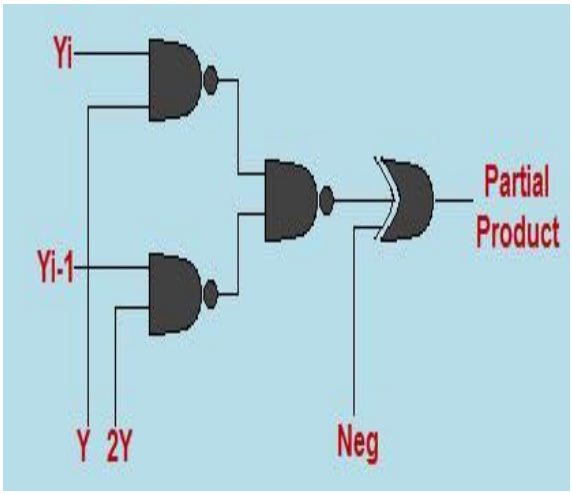
Booth's Encoder

Modified booth multiplier's (Z) digits can be defined with the following equation:

$Zj = q2j + q2j-1 - 2q2j+1$ with $q-1 = 0$

The figure shows the modified booth algorithm encoder circuit. Now, the product of any digit of Z with multiplicand Y may be -2y, -y, 0, y, 2y. But, by performing left shift operation at partial products generation stage, 2y may be generated. By taking 1's complement to this 2y, negation is done, and then one is added in appropriate 4-2 compressor. One booth encoder shown in the figure generates three output signals by taking three consecutive bit inputs so as to represent all five possibilities -2X, -X, 0, X, 2X.

Partial Product Generator

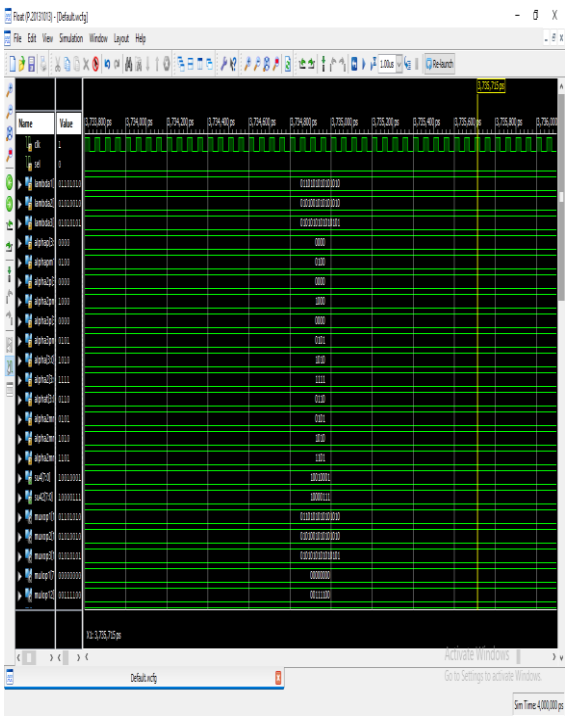


Partial Product Generator

If we take the partial product as -2y, -y, 0, y, 2y then, we have to modify the general partial product generator. Now, every partial product point consists of two inputs (consecutive bits) from multiplicand and, based on the requirement, the output will be generated and its complements also generated in case if required. The figure shows the partial product generator circuit.

The 2's complement is taken for negative values of y. There are different types of adders such as conventional adders, ripple-carry adders, carry-look-ahead adders, and carry select adders. The carry select adders (CSLA) and carry-look-ahead adders are considered as fastest adders and are frequently used. The multiplication of y is done by after performing shift operation on y – that is – y is shifted to the left by one bit.

RESULT



CONCLUSION:

This is new low-power architecture for parallel CS provided. By reducing access to the second stage of the conventional CS to achieve significant power savings is decomposed in two steps. Error operate under the same ownership, the less energy the size of the CS in the construction sector in different configurations, and error-correction capability of the horizontal factor compared to traditional construction. Final implementation with Radix-n modified booth encoding algorithm yields reduction in density and power consumption.

REFERENCES:

[1] Jiun-Ping Wang, Shiann-Rong Kuang, Shish Chang Liang, "High Accuracy Fixed Width Modified Booth Multipliers for Lossy Applications", IEEE Trans 2011.

[2] Pouya Asadi and Keivan Navi, "A new low power 32×32 bit multiplier", World Applied Sciences Journal 2 (4): 341-347,2007.

[3] A.D. Booth, "A signed binary multiplication technique", Quart. J. Mech. Appl. Marh, vol.4, pp. 236-240, 1951. (Reprinted in [8, pp. 100-104])

[4] Razaidi Hussin, Ali Yeon Md. Shakaff, Norina Idris, Zaliman Sauli, Rizala fande CheIIsmail, and Afzan Kamaraudin, "An efficient modified Booth multiplier architecture", International Conference on Electronics Design, 978-1-4244-2315-6/08, 2008 IEEE.

[5] Ramya Muralidharan, Chip Hong Chang," Radix-4 and Radix-8 Booth encoded multi-modulas multipliers" IEEE Trans, 2013.

[6] Vignesh Kumar R., Kamala J., "High Accuracy Fixed Width Multipliers using Modified Booth Algorithm",International Conference on Modeling Optimization and Computing, Procedia Engineering 38(2012) 2491-2498.

[7] Y. Wu, "Low power decoding of BCH codes," in Proc. IEEE ISCAS, May 2004, pp. II-369–II-372.

[8] S. Wong, C. Chen, and Q. M. Wu, "Low power Chien search for BCH decoder using RT-level power management," IEEE Trans. Very Large Scale Integr. Syst., vol. 19, no. 2, pp. 338–341, Feb. 2011.

[9] H. Weingarten, E. Sterin, O. A. Kanter, and M. Katz, "Low Power Chien-Search Based BCH/RS Decoding System for Flash Memory, Mobile Communications Devices and Other Applications," U.S. Patent 2010 013 1831 A1, May 27, 2010.

[10] Y. Chen and K. K. Parhi, "Small area parallel Chien search architectures for long BCH codes," IEEE Trans. Very Large Scale Integr. Syst., vol. 12, no. 5, pp. 545–549, May 2004.

[11] J. Cho and W. Sung, "Strength-reduced parallel Chien search architecture for strong BCH codes," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 55, no. 5, pp. 427–431, May 2008.

[12] Y. Lee, H. Yoo, and I.-C. Park, "Low-complexity parallel Chien search structure using two-dimensional optimization," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 58, no. 8, pp. 522–526, Aug. 2011.

[13] H. Choi, W. Liu, and W. Sung, "VLSI implementation of BCH error correction for multilevel cell NAND Flash memory," IEEE Trans. Very Large Scale Integr. Syst., vol. 18, no. 5, pp. 843–847, May 2010.