論文 88-25-2-13

# Implementation of High Reliable Fault-Tolerant Digital Filter Using Self-Checking Pulse-Train Residue Arithmetic Circuits

# (自己檢査 Pulse 列 剩餘數演算回路를 이용한 高信賴化 Fault Tolerant 디지탈필터의 構成에 관한 硏究)

金 汶 洙* 孫 東 仁** 田 龜 濟***

(Moon Soo Kim, Dong In Shon and Koo Chae Chun)

**要 約**

剩餘數系는 剩餘數 表現이 各段 獨立的이며 並列演算이 可能하고 誤差의 檢出과 訂正이 可能하다는 特徵을 가지므로 高信賴化 디지탈 信號處理系에의 應用이 期待되어지고 있다. 本 論文에서는 剩餘數系를 基本으로한 高信賴化 디지탈 信號處理系를 效率的으로 構成하는 方法으로 誤差檢出은 基本 回路內部 에서 各 段마다 實施하고 誤差訂正은 剩餘數系의 並列性을 이용해서 行하는 誤差檢出 訂正方法을 提示하였다. 應用例로서 誤差訂正回路가 非冗長系에 直列로된 設計方法을 提案하였으며 設計대로 2次 巡回形 Fault Tolerant 디지탈 필터가 基本回路數를 大幅 減少시켜 構成되고 正確히 動作하며 信賴度 역시 크게 向上됨을 提示하였다. 또한 單一雜音 注入試驗에 의해 構成된 하드웨어의 Fault Tolerant 能力을 確認하였다.

**Abstract**

The residue number system offers the possibility of high-speed operation and error detection/correction because of the separability of arithmetic operations on each digit. A compact residue arithmetic module named the self-checking pulse-train residue arithmetic circuit is effectively employed as the basic module, and an efficient error detection/correction algorithm in which error detection is performed in each basic module and error correction is performed based on the parallelism of residue arithmetic is also employed.

In this case, the error correcting circuit is imposed in series to non-redundant system. This design method has an advantage of compact hardware. Following the proposed method, a 2nd-order recursive fault-tolerant digital filter is practically implemented, and its fault-tolerant ability is proved by noise injection testing.

*正會員, 國立 釜山開放大學 電子工學科
(Dept. of Elec. Eng., Nat'l Busan Open Univ.)

**正會員, 釜山專門大學 電子通信科
(Dept. of Elec. Communication, Busan Junior College.)

***正會員, 東亞大學校 電子工學科
(Dept. of Elec. Eng., Busan Dong-A Univ.)

接受日字 : 1987年 2月 25日

## I. Introduction

During these days, digital signal processing systems are increasingly being used in the range of instrumentation, control engineering, space-communication, and electronic medical devices. These systems are working on the very important positions and sometimes working under the bad conditions. When there is

any fault on the system, it will make a big trouble. There is an obvious need for highly-reliable digital signal processing systems.

The residue number system(RNS)[1-3] offers the possibility of high-speed operation and error detection/correction because of the separability of arithmetic operations on each digit. The implementation of digital filters based on the residue number system has been studied [8-10]. The ROM table-look-up method is usually proposed for the implementation of residue arithmetic circuits [4]. Much memory is needed in this ROM method, so that the hardware of the digital filter is increased. Also, the error detection/correction based on the RNS have been studied mainly from the mathematical view point [5-7] and little attention has been given to the efficient hardware implementation.

On the other hand, we have proposed a compact residue arithmetic circuit named the self-checking pulse-train residue arithmetic circuit [10] in which the ring-counter [8] is used as the major component and arithmetic operations are performed by counting pulse and code conversion [8]. We have also proposed a new efficient error detection and correction method in which error detection is performed in each self-checking pulse-train residue arithmetic circuit and error correction is performed based on the parallelism of residue arithmetic [9]. According to the proposed method, a 2nd-order recursive fault-tolerant digital filter is practically implemented and a noise testing is done on the implemented digital filter. In this case, the error correction circuit is imposed in series to non-redundant system. This design method has an advantage of compact hardware. It is found that this hardware complexity is 70% of the one of the conventional tripple modular redundancy(TMR), and the error detection/correction can be perfectly performed under the condition of a single noise injection.

## II. Overview of the Self-checking Pulse-Train Residue Arithmetic Circuit and its Error Detection and Correction

Since the detailed explanation about the RNS is found in [2], it will be discusse briefly here.

In the residue number system, an integer X is represented by Eqs. (1) and (2), where $m_1, m_2, ..., m_N$ are moduli which are selected from mutually prime integers [2].

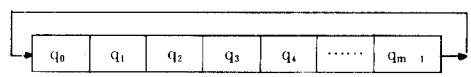
$$X=(x_1, x_2, ..., x_N) \quad (1)$$

$$x_i = |X\ m_i\ (i=1, 2, ..., N) \quad (2)$$

The symbol $|x|$ denotes the remainder of X, when divided by the modulus. The dynamic range of X represented by Eq. (1) is

$$0 \leqq X < M$$
$$M=m_1\ m_2\ ...\ m_N \quad (3)$$

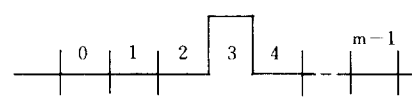
where X is a non-negative integer.




(a) model of the ring counter

| X | $q_0$ | $q_1$ | $q_2$ | $q_3$ | $q_4$ | ..... | $p_{m-1}$ |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | 0 | ..... | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | | 0 |
| 2 | 0 | 0 | 1 | 0 | 0 | | 0 |
| 3 | 0 | 0 | 0 | 1 | 0 | | 0 |
| 4 | 0 | 0 | 0 | 0 | 1 | | 0 |
| ⋮ | ⋮ | ⋮ | | | | | ⋮ |
| m − 1 | 0 | 0 | 0 | 0 | 0 | | 1 |

(b) code of the number X




(c) output signal

**Fig. 1.** State of the ring counter.

Addition and multiplication of residue numbers X, Y are given by

$$X + Y = (x_1 \oplus y_1, x_2 \oplus y_2, ..., n_N \oplus y_N) \quad (4)$$

$$X \times Y = (x_1 \odot y_1, x_2 \odot y_2, ..., x_N \odot y_N) \quad (5)$$

where $\oplus$ and $\odot$ are modulo-$m_i$ addition and modulo-$m_i$ multiplication respectively on each digit, as given by

$$x_i \oplus y_i = |x_i + y_i|_{m_i}$$

$$x_i \odot y_i = |x_i\, y_i|_{m_i}$$

Eqs. (4) and (5) imply that addition and multiplication in the residue number system can be performed completely in parallel on each digit.

Division in the residue number system is very complex. Scaling[2], which is division by a constant K, is needed later in the recursive digital filter to prevent overflow.

### *1. Self-checking pulse-train residue arithmetic circuit and its error detection.*

Fig. 2 shows the symbol of the basic module[8] used in this paper. The module is an arithmetic circuit on a single residue digit. Let inputs be A and B, the output X, the multiplication constant K, and the modulus m, then the arithmetic function of the basic module is expressed as follows:

$$X(kT0) = |(A((k-1)To) - B((k-1)To))K|_m \quad (6)$$

where to denotes the operating time of the basic module and k the descrete time. From Eq.(6), we see that the basic module has all of the arithmetic functions of addition, multiplication and delay which are needed in digital signal processing. Fig. 3 illustrates the schematic circuit of the basic module, which is composed of the difference ring-counter, the product ring-counter, the line exchanger and the shift control circuit. The 1 of N code is used in the ring-counter. The operation of A−B is performed by applying shift-pulses to the difference counter. The multiplication by K is performed by the code conversion between the difference counter and the product counter.

Two kinds of errors are possible to be generated in the basic module[9]. One is the code-error which generates some code different from 1 of N code, and the other is the shifting-error of the ring-counter. Error detection of

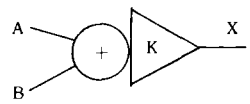



**Fig. 2.** Symbol of the basic module.

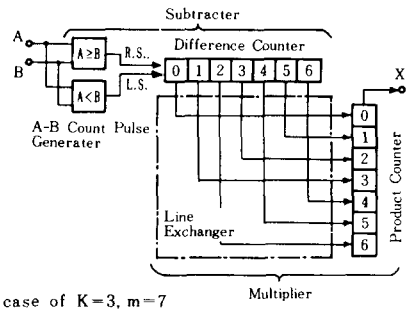



**Fig. 3.** Schematic circuit of the basic module.

the basic module is performed by the combination of the code error detection and shifting error detection. In this paper, following conditions are assumed:

[Condition 1] Any error other than the shifting error of the ring-counter takes the form of the code-error.

[Condition 2] A shifting error of the ring-counter is a single bit error. The code error can be detected by checking the number of bits with level 1. A shifting error is detectable by checking coincidence between the ring-counter and the binary counter which operates so as to hold Q modulo-2, where Q denotes the content in the ring-counter. In Fig. 5, the schematic circuit of the shifting error

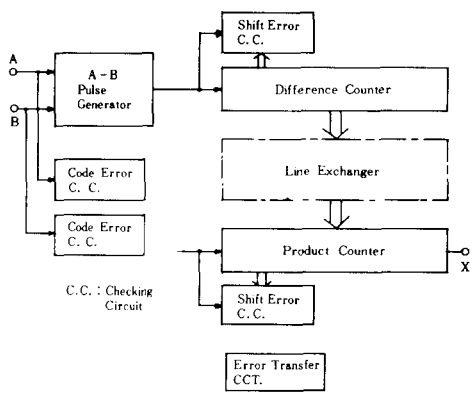



**Fig. 4.** Self-checking basic module.

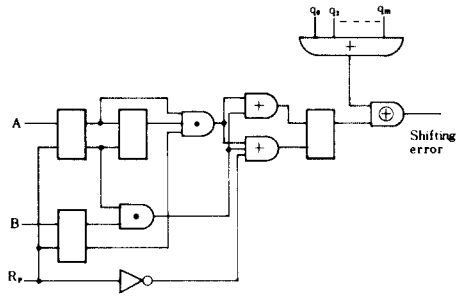



**Fig. 5.** Shifting error detecting circuit.

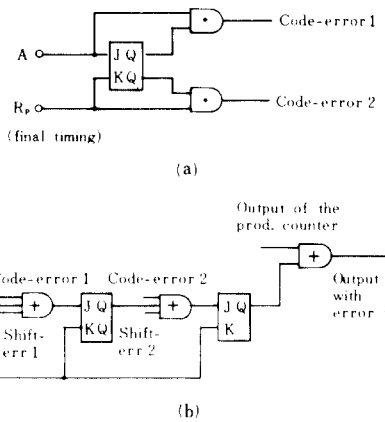



**Fig. 6. (a)** Code error detecting circuit. **(b)** Error transmitting circuit.

and in Fig. 6, code error detection circuit and error transmitting circuit are shown.

*2. Error correction based on the puls-train residue arithmetic circuit.*

When error is detected by using self-checking pulse-train residue arithmetic circuits, the error correction is realizable by the error correction algorithm based on the base extension in the residue arithmetic [9]. Next property is known on this error correction method.

[Property] If R redundant digits are added, then R error digits are correctable.

In this paper, following condition is assumed:

[Condition 3] In an operating cycle of the basic module, error occurs in a single digit.

Under this condition, only one redundant digit is required. Fig. 9 shows an error correction circuit [9].

## III. Design Method of the Fault-Tolerant Digital Filters

As well known, the transfer function H(Z) of the digital filter is generally expressed by

$$H(Z) = \frac{a_0 + a_1 Z^{-1} + a_2 Z^{-2} + \ldots + a_m Z^{-m}}{1 + b_1 Z^{-1} + b_2 Z^{-2} + \ldots + b_n Z^{-n}} \quad (7)$$

If all of $b_1, \ldots, b_n$ are 0, the digital filter is called the non-recursive digital filter, otherwise the recursive digital filter.

*1. Design method of the non-recursive fault-tolerant digital filter.*

Consider the non-recursive digital filter expressed by

$$H_1(Z) = a_0 + a_1 Z^{-1} + \ldots + a_m Z^{-m} \quad (8)$$

The operation of this digital filter is represented by

$$y(kT) = a_0\, x(kT) + a_1\, x((k-1)T) + \ldots + a_m\, x((k-m)T) \quad (9)$$

where x, y denote the input and the output respectively.

Following to the above Eq.(9), the non-recursive fault-tolerant digital filter can be implemented as shown $H_1(z)$ in Fig. 7.

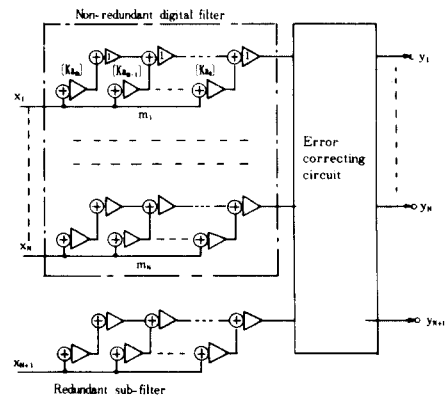



**Fig. 7.** Non-recursive fault-tolerant digital filter.

The digital filter is composed of a non-redundant digital filter, a redundant sub-filter and the error correction circuit.

The basic modules used here are the self-checking basic modules. Error detection is performed in each modules. When an error is detected in a basic module, the 'ERROR' signal is sent to the next module together with the normal output signal. Error correction is performed at the final stage.

### *2. Design method of the recursive fault-tolerant digital filter*

Consider the design of the recursive digital filter expressed as

$$H_2(Z) = \frac{1}{1 + b_1 Z^{-1} + b_2 Z^{-2} + \ldots + b_n Z^{-n}} \tag{10}$$

The operation of the digital filter is expressed by

$$y(kT) = x(kT) - b_1 y((k-1)T) - b_2 y((k-2)T) - \ldots - b_n y((k-n)T). \tag{11}$$

The Eq.(11) is integerized as

$$y(kT) = [Kx(kT) - Kb_1 y((k-1)T) - Kb_2 y((k-2)T) - \ldots - Kb_n y((k-n)T)] / K. \tag{12}$$

Following be above Eq.(12), the fault-tolerant recursive digital filter can be implemented as shown in Fig. 8. The digital filter is composed of a non-redundant digital filter, a redundant sub-filter and the error correction circuit. The scaling circuit in Fig. 10 performed 1/K operation to prevent to the overflow. Because no digits in the scaling circuit are independent, an error in a digit is transmitted to all digits on the scaling circuit. To avoid this problem, the error correction circuit is performed before the scaling circuit. Errors which occur in the scaling circuit can be corrected if the duplex hardware of the scaling circuit is provided.

The recursive digital filter with H(Z) expressed by Eq.(7) is implemented as follows:

First, H(Z) is writen as

$$H(Z) = H_1(Z) * H_2(Z). \tag{13}$$

The digital filter with H(Z) can be constructed by connecting the recursive digital filter with $H_2(Z)$ to the non-recursive digital filter with $H_1(Z)$ in series. If the connection is made as $H_1(Z) \Rightarrow H_2(Z)$, then the error correction circuit in the non-recursive digital filter can be omitted. The digital filter can be constructed as shown in Fig. 8.

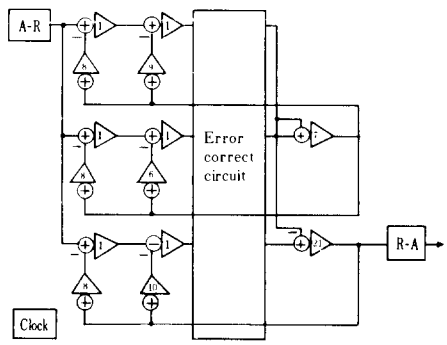



**Fig. 8.** Fault-tolerant digital filter designed.

## VI. Implementation of the Fault-Tolerant Digital Filter

A 2nd-order recursive digital filter is practically implemented, which is the most basic one. TTL ICs are used as the components.

### *1. Implementation of the self-checking pulse-train residue arithmetic circuit*

Fig. 4 shows the block-diagram of the self-checking basic module which is constructed by adding 2 code error detection circuits, the shifting error detection circuit of the difference counter, the shifting error detection circuit of the product counter and the error transmitting circuit to the standard basic module. The code error detection circuits are connected to the input terminals. By this connection, errors which will occur on the interconnections between modules are also detectable. Fig. 5 showes the shifting error detection circuit. Fig. 6 showes the code error

detection circuit and the error transmitting circuit. Rp is the final timing pulse in an operating cycle.

### 2. *Implementation of the error correction circuit*

Fig. 9 shows the error correction circuit. This circuit is implemented using the base extension theory. Basic modules belonging to the n-th column have to be disabled when an error is detected in the n-th digit. The disabling operation is realized by using multiplexers provided in parallel to each basic modules. The error digit is corrected by using two legitimate digits.

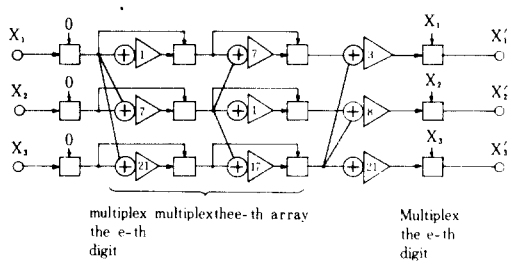



**Fig. 9.** Error correcting circuit.

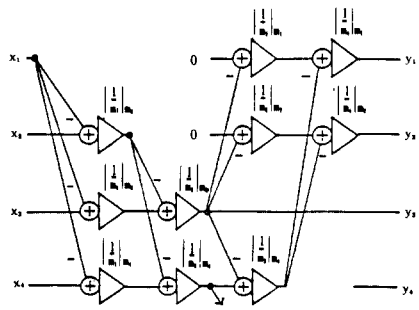


**Fig. 10.** Scaling circuit.

### 3. *Implementation of the fault-tolerant digital filter*

At first, the parameters of fault-tolerant digital filter is determined as Table 1. In Fig. 8, the whole block-diagram of the digital filter is illustrated. The duplex hardware of the scaling circuit is omitted. Fig. 12 (a) and (b) show the normal output waveform after error correction and the operation with error respectively.

Fig. 11 shows the step response obtained by simulation using a personal computer. Because the wave-form in Fig.12(a) exactly coincides with the one in Fig. 11, we recognize that the digital filter fabricated is normally operating and error detection/correction circuits are perfectely working.

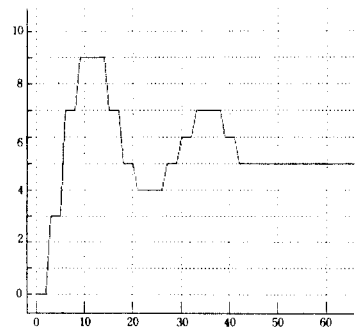
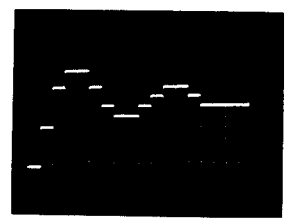
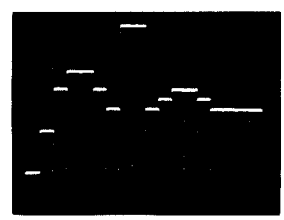



**Fig. 11.** Step response of the digital filter (by simulation).



(a) Normal operation after error correction.



(b) Response with error correction.

**Fig. 12.** Step responses of the digital filter (partical wave form).

**Table 1.** Parameters of the fabricated digital filter.

| | |
|---|---|
| non-redundant moduli | $m_1=11$, $m_2=19$ |
| redundant modulus | $m_3=23$ |
| scaling factor | $K=m_1=11$ |
| dynamic range(overall) | $M=m_1*m_2=209$ |
| dynamic range(after scaling) | $M/K=m_2=19$ |
| filter coefficients | $b_1=-1.1$, $b_2=.8$ |
| | $Kb_1=-13$, $Kb_2=8$ |

## V. Conclusion

The design, implementation and noise testing of a high reliable digital signal processing system has been discussed here using the self-checking pulse-train residue arithmetic circuits. It is made clear that a selfchecking pulse-train residue arithmetic circuit can be constructed with TTL ICs, and that a 2nd-order recursive fault-tolerant digital filter with 3 moduli can be constructed with 23 self-checking pulse-train residue arithmetic circuits. Table 2 showes the comparison of digital filters. It is seen that the hardware of the digital filter in this paper is 60-70% of the one of typical TMR digital filter. The experimental results confirm the feasibility of the use of self-checking pulse-train residue arithmetic circuits based RNS in the high reliable digital signal processing. The anti-practical noise testing of the digital filter will be the next problem.

**Table 2.** Comparison of the hardware complexity of digital filters.

| | proposed | TMR |
|---|---|---|
| N=4, m=n=2 | 127 | 138 |
| N=4, m=n=4 | 190 | 258 |
| N=4, m=n=8 | 315 | 498 |

N=non-redundant digits
n=orders of digital filter

## References


[1] A.V. Oppenheim, editor: *"Application of digital signal processing"*, Prentice-Hall, Englewood Cliffs, New Jersey, 1978.

[2] N.S. Szabo and R.I. Tanaka: *"Residue arithmetic and its application to computer technology"*, McGraw-Hill, New York, 1967.

[3] W.K. Jenkins, B.J. Leon: "The use of residue number systems in the design of finite impulse response digital filters", *IEEE Trans. Circuit and Syst.*, CAS-24, pp.191-201, April 1977.

[4] G.A. Jullien: "Residue number scaling and other operations using ROM arrays", *IEEE Trans. Comput.*, C-22, pp.325-336, April 1978.

[5] H.M. Etzel and W.K. Jenkins: "Redundnat residue number systems for error detection and correction in digital filters", *IEEE Trans. Acoust. Speech and Signal Processing*, ASSP-28, pp.538-545, Oct. 1980.

[6] M. Kameyama and T. Higuchi: "Design of fault-tolerant digital filter based on residue number system", *Technical report on Information Theory and Its Application*, pp. 12-19, Nov. 1979.

[7] S. Yau and Y. Liu: "Error correction in redundant number systems", *IEEE Trans. Comput.*, C-22, pp.5-11, Jan. 1973.

[8] N. Tomabechi, M. Kameyama and T. Higuchi: "Pulse rate arithmetic circuit based on residue number system and its application to digital filter", *IECE Japan Trans.*, vol. J65-D, pp. 147-153, Feb. 1982.

[9] N. Tomabechi, M. Kameyama and T. Higuchi: "Highly reliable digital signal processing system using pulse-train residue arithmetic circuits", *IECE Japan Trans.*, vol. J66-D, pp. 340-347, March 1983.

[10] N. Tombechi and M.S. Kim: "Implementation of a fault-tolerant digital filter using pulse-train residue arithmetic circuit", *IECE Japan Technical Report*, FTS89-21, Nov. 1986. *