

A high-level synthesis based dual-module redundancy with multi-residue detection (DMR-MRD) fault-tolerant method for on-board processing satellite communication systems^①

Yang Wenhui (杨文慧)^{*}, Chen Xiang^{②**}, Wang Yu^{***}, Zhao Ming^{****}, Wang Jing^{****}

(^{*} Communication Engineering of Xiamen University, Xiamen 361005, P. R. China)

(^{**} Aerospace Center, Tsinghua University, Beijing 100084, P. R. China)

(^{***} Department of Electronic Engineering, Tsinghua University, Beijing 100084, P. R. China)

(^{****} Wireless and Mobile Communication Technology R&D Center, Tsinghua University, Beijing 100084, P. R. China)

Abstract

On board processing (OBP) satellite systems have obtained more and more attentions in recent years because of their high efficiency and performance. However, the OBP transponders are very sensitive to the high energy particles in the space radiation environments. Single event upset (SEU) is one of the major radiation effects, which influences the satellite reliability greatly. Triple modular redundancy (TMR) is a classic and efficient method to mask SEUs. However, TMR uses three identical modules and a comparison logic, the circuit size becomes unacceptable, especially in the resource limited environments such as OBP systems. Considering that, a new SEU-tolerant method based on residue code and high-level synthesis (HLS) is proposed, and the new method is applied to FIR filters, which are typical structures in the OBP systems. The simulation results show that, for an applicable HLS scheduling scheme, area reduction can be reduced by 48.26% compared to TMR, while fault missing rate is 0.15%.

Key words: single event upset (SEU), residue code, triple modular redundancy (TMR), high-level synthesis (HLS), fault missing rate

0 Introduction

Different to the traditional Bent Pipe (BP) satellite communication systems, the on-board processing (OBP) ones become a more adaptive alternative due to the development of satellite communication applications, the demand for communication quality and the increased capacity^[1]. Baseband OBP is commonly referred as a fully processing satellite platform, so all the physical layer techniques, including demodulation/modulation, decoding/encoding and channel estimation/equalization should be performed for the signal regeneration for each sub-channel^[2,3]. The Thuraya system is a good example of current satellite communication systems with a baseband OBP platform^[4]. Since complete switching is required in baseband OBP, FIR and FFT are still the basic modules. As introduced in Refs[2,5,6], the orthogonal frequency division multiplexing (OFDM) technology is an attractive candidate

when targeting high quality and high flexibility in future mobile multimedia satellite communication systems, so FFT is still a necessary module. In addition, almost all the current mobile satellite communication systems and future ones, no matter digital channelization or fully processed base band platform, would apply digital beam-forming (DBF) for multiple-beam coverage^[4,7-11], so DBF is also a necessary DSP module on OBP platforms^[12,13].

In general, off-the-shelf SRAM-FPGA is chosen as the alternative for OBP implementation because of its high density, high performance, reduced development cost and re-configurability, the last of which is quite useful for remote update and maintenance of the OBP satellite systems^[14]. However, SRAM-FPGA are sensitive to the radiation in space radiation environments, so they are not reliable enough for spatial applications without any protection. Among the radiation effects, the single event upset (SEU) is one of the major one, and the SEU-tolerant method is a key issue for

① Supported by the National S&T Major Project (No. 2011ZX03003-003-01, 2011ZX03004-004) and the National Basic Research Program of China (No. 2012CB316002).

② To whom correspondence should be addressed. E-mail: chenxiang@tsinghua.edu.cn
Received on Apr. 16, 2013

the feasibility of the SRAM-FPAG based application on the OBP platform. The triple modular redundancy (TMR) is a classic and effective SEU-tolerant method, which applies three identical modules to perform the same process, and the results are processed by a majority voter to produce a single output. When a single module fails, the majority voter can help masking the faults and give the correct results^[15]. However, large area overhead is a fatal defect of TMR. Ref. [16] shows that, space based rader requires 100s of giga floating point operations of on-board processing and 10s of Gbps data links to accomplish mission goals. A TMR approach for such a program would create a system that weighs hundreds of pounds and requires kilowatts of power, which are unbearable for an OBP platform. Therefore, low-cost fault-tolerant design is in urgent need.

To reduce the area overhead of TMR, many prior fault tolerant methods have been proposed based on residue code^[17], residue number system (RNS) and a redundant RNS^[18,19], which is an extended algorithm based on RNS. In these existing residue code based fault tolerant methods, however, some shortcomings reduce their feasibility. In RRNS, several CRT modulus are added for fault detection of parallel computation branches, but the CRT modulus themselves are not protected from SEUs. What's more, the dynamic range of the operand is limited by the modulus. FIR plus two checking modules based on residue code was given in Ref. [17], which located the upset position according to the remainders' patterns. However, this method is effective for single bit error in the output, and this kind of errors occupies only a small part of the fault model. In addition, the fault missing problems are ignored in this method. To overcome these shortcomings, Ref. [20] proposed a single-sample checking based dual modules plus check module based on residue code (SSC-DM-CRC). But in SSC-DM-CRC method, a small modulus leads to a high fault missing rate, while large modulus leads to a quite large area overhead.

Another cost-effective method is the high level synthesis (HLS). Refs [21] and [22] considered SEU-tolerant in HLS. Ref. [22] presented a method to HLS of data path with concurrent error detection, which copied the entire flow graph and scheduled both of the copies in order to increase the reliability and reduce the area overhead. Ref. [21] presented another HLS method which centered the reliability as well as concerning about the area and delay boundary. In this method, the resources with different areas, delay and reliability are selected via iteration to maximum the re-

liability on the premise of inbounding latency and area constraints. Although these methods reduce the area overhead and a relatively high reliability, but the reliability cannot meet our requirement yet.

In this paper, a new method, in which residue code and HLS are used, is proposed. In this paper, we use a group of relative prime number as moduli to replace the module branch, in order to ensure a lower fault missing rate, and use HLS to cut down the area overhead.

The focus of this paper is the low cost SEU-tolerant design for the DSPs with the structure expressed as

$$y = \sum_0^{L-1} x(l) \times h(l) \quad (1)$$

in which $x(l)$ and $h(l)$, $l = 0, 1, \dots, L-1$ are input data for the current operation and coefficients, respectively. Since only multiplications and additions are involved, this structure is called multiply and accumulation (MAC). The common used DSPs, such as FIR, FFT and DBF all belong to MAC structure. For FIR, $h(l)$ is the filter coefficient. For FFT, $h(l)$ is the rotation factor. And for DBF, $h(l)$ is the weighting coefficient. To facilitate the description and analysis, this paper will focus on the SEU-tolerant FIR design. The analysis method and theoretical results can be easily applied to other DSPs with MAC structure^[20].

Our key contributions in this paper include three points:

- (1) A fault-tolerant structure named DMR-MRD based on residue code is proposed.
- (2) A resource-limited ASAP scheduling algorithm is designed.
- (3) A set of test method including fault injection and fault missing rate estimating are designed.

1 Background

The residue code has been widely used to DSPs to reduce the hardware overhead. It converts the operands to their remainders by dividing a given modulus^[20]. The interesting character of the residue code is that, it maintains the arithmetical and logical properties of operand and invariant for linear operations, including additions/subtractions and multiplications. For operand X , Y and modulus m , they have the following relationship.

$$(X \text{ op } Y)_m = ((X)_m \text{ op } (Y)_m)_m \quad (2)$$

where $(\cdot)_m$ represents the mod m operation.

The residue number system (RNS) is based on residue code. It is defined by a set of p pairwise relatively prime integers, $\{m_1, m_2, \dots, m_p\}$. The dynamic range of RNS is: $M = m_1 \cdot m_2 \cdot \dots \cdot m_p$. Then

an integer $X \in [0, M)$ can be uniquely expressed in RNS as

$$X \xrightarrow{RNS} \{(X)_{m_1}, (X)_{m_2}, \dots, (X)_{m_p}\} \quad (3)$$

where $(X)_{m_i}$ means $X \bmod m_i$. In RNS, linear operation between X and Y , including additions, subtractions and multiplications in two's complement system, can be transformed to a set of operations on residues as

$$Z = X \text{ op } Y \rightarrow \begin{cases} (Z)_{m_1} = ((X)_{m_1} \text{ op } (Y)_{m_1})_{m_1} \\ (Z)_{m_2} = ((X)_{m_2} \text{ op } (Y)_{m_2})_{m_2} \\ \dots \\ (Z)_{m_p} = ((X)_{m_p} \text{ op } (Y)_{m_p})_{m_p} \end{cases} \quad (4)$$

where result Z can be recovered based on $(Z)_{m_i}$ by a Chinese remainder theorem (CRT) as Ref. [23]

$$Z = CRT\{(Z)_{m_1}, (Z)_{m_2}, \dots, (Z)_{m_p}\} \quad (5)$$

HLS targets optimizing the register transfer level (RTL) hardware performance, area and power requirements. In general, HLS includes the following steps^[24]:

Step 1 Compile the specification: this step transforms the input description to a formal description. The formal mode is usually the data flow graph (DFG)/control data flow graph (CDFG). In most cases, the input of HLS flow is represented in C/C++.

Step 2 Constraint: resource allocation is to define the number and type of resources according to the design constraints.

Step 3 Resource scheduling: in this step, the resources are allocated to different control steps (CSTEPS), according to different scheduling algorithms, such as the as soon as possible (ASAP) and the as late as possible (ALAP) algorithms.

Step 4 Binding: this step includes function units binding and registers binding, every operation and variable need to be bounded to a relative hardware element.

Step 5 RTL generation: RTL code is generated after the above four steps.

The area overhead is decided by the resource allocation under the condition of time constraints. It is a tradeoff between area and time constraints, if area requirement is strict, then we have to relax the time constraints and vice versa. Thus, we can find a compatible scheme according to the requirements of our design.

2 HLS based DMR-MRD method

The proposed method is called HLS based dual-modular redundancy multi-residue detection (DMR-MRD), which can be divided into two steps. Firstly, the DMR-MRD method is used to protect the design

and provide a lower fault missing rate while it introduces more hardware area overhead. Secondly, HLS is utilized to schedule the hardware resources, giving a desirable resource number. The FIR filter, which is a common module in many systems, is taken as an example for applying the DMR-MRD method.

2.1 The DMR-MRD method

The structure of an FIR filter module after using the DMR-MRD method is shown in Fig. 1. It consists of two normal FIR filter modules, denoted as M_1 and M_2 , and three residue modules, denoted as RM_1 , RM_2 and RM_3 . The outputs of five modules are processed by the synthesis logic, which is expected to output the correct result if an error caused by SEU occurs in one of the branches. In this structure, the sample data x is fed to FIR filter modules M_1 and M_2 directly, and to residue modules RM_1 , RM_2 and RM_3 after the mod operation. The outputs of M_1 and M_2 are y_1 and y_2 , respectively, while the outputs of RM_1 , RM_2 and RM_3 are rm_1 , rm_2 and rm_3 , respectively. r_{ij} ($i = 1, 2, j = 1, 2, 3$) is the remainder of y_i under the modulus m_j .

The flow diagram of the synthesis logic is given by Fig. 2, and further explained as follows.

(1) If $y_1 = y_2$, y_1 is chosen as the output; namely, $y = y_1$.

(2) If $y_1 \neq y_2$, rm_1 , rm_2 and rm_3 are used to check the correctness of y_1 and y_2 . Eight cases exist here.

- a. $r_{11} = rm_1$, $r_{21} \neq rm_1$: y_1 is chosen as output;
- b. $r_{11} \neq rm_1$, $r_{21} = rm_1$: y_2 is chosen as output;
- c. $r_{11} = rm_1$, $r_{21} = rm_1$ (or $r_{11} \neq rm_1$, $r_{21} \neq rm_1$), $r_{12} = rm_2$, $r_{22} \neq rm_2$: y_1 is chosen as output;
- d. $r_{11} = rm_1$, $r_{21} = rm_1$ (or $r_{11} \neq rm_1$, $r_{21} \neq rm_1$), $r_{12} \neq rm_2$, $r_{22} = rm_2$: y_2 is chosen as output;
- e. $r_{11} = rm_1$, $r_{21} = rm_1$ (or $r_{11} \neq rm_1$, $r_{21} \neq rm_1$), $r_{12} = rm_2$, $r_{22} = rm_2$ (or $r_{12} \neq rm_2$, $r_{22} \neq rm_2$), $r_{13} = rm_3$, $r_{23} \neq rm_3$: y_1 is chosen as output;
- f. $r_{11} = rm_1$, $r_{21} = rm_1$ (or $r_{11} \neq rm_1$, $r_{21} \neq rm_1$), $r_{12} = rm_2$, $r_{22} = rm_2$ (or $r_{12} \neq rm_2$, $r_{22} \neq rm_2$), $r_{13} \neq rm_3$, $r_{23} = rm_3$: y_2 is chosen as output;
- g. $r_{11} = rm_1$, $r_{21} = rm_1$ (or $r_{11} \neq rm_1$, $r_{21} \neq rm_1$), $r_{12} = rm_2$, $r_{22} = rm_2$ (or $r_{12} \neq rm_2$, $r_{22} \neq rm_2$), $r_{13} = rm_3$, $r_{23} = rm_3$. In this case, the error cannot be identified;

h. $r_{11} = rm_1$, $r_{21} = rm_1$ ($r_{11} \neq rm_1$, $r_{21} \neq rm_1$), $r_{12} = rm_2$, $r_{22} = rm_2$ (or $r_{12} \neq rm_2$, $r_{22} \neq rm_2$), $r_{13} \neq rm_3$, $r_{23} \neq rm_3$. In this case, all branches fail, which is out of the range of this paper.

Three residue branches are used in this structure to check the error, so if and only if $y_1 \neq y_2$, and both

y_1 and y_2 are congruent to all the three moduli, the fault missing event happens (case g). The probability of the fault missing events is called fault missing rate. In the

next sub-section, fault missing rate of the proposed method for L -tap FIR filters is analyzed.

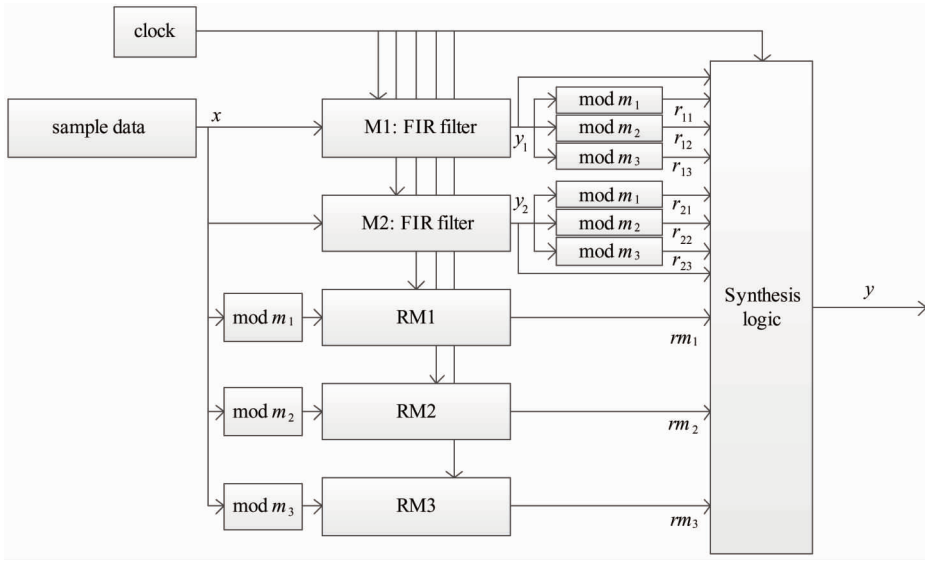


Fig. 1 Structure of the proposed method

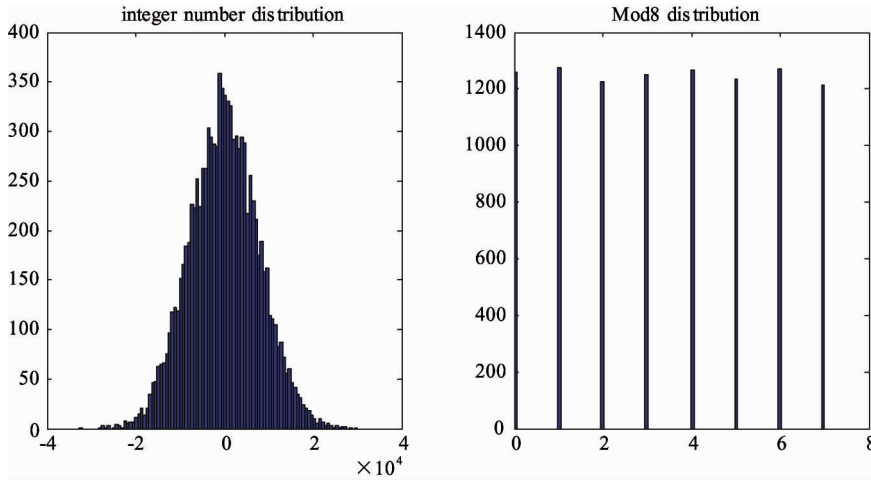


Fig. 2 Data with Gaussian distribution and the its remainder with uniform distribution

2.2 Analysis of the fault missing rate

For an L -tap FIR filter, the output can be expressed as

$$y = x_0 \times h_0 + x_1 \times h_1 + \cdots + x_l \times h_l + \cdots + x_{L-1} \times h_{L-1} \quad (6)$$

where x_l is the l th sample data, h_l is the l th coefficients. Since only multipliers and adders exist in the MAC structure, the fault models can be classified into two classes, multiplier-input fault (MIF) models and adder-input fault (AIF) models.

Assume that coefficient h_l experiences SEU in the MIF model. Since sample data x and coefficient h are equivalent in the MAC structure, the case that x_l experiencing SEU is not discussed separately. When the q th bit of h_l upsets, the filter's output becomes

$$\begin{aligned} y_e &= x_0 \times h_0 + x_1 \times h_1 + \cdots + x_l \times (h_l \pm 2^q) \\ &+ \cdots + x_{L-1} \times h_{L-1} \\ &= x_0 \times h_0 + x_1 \times h_1 + \cdots + x_l \times h_l + \cdots \\ &+ x_{L-1} \times h_{L-1} \pm x_l \times 2^q \end{aligned} \quad (7)$$

where $x_l \cdot 2^q$ is the error caused by SEU. Compare Eq. (6) with Eq. (7), we have

$$y_e - y = \pm x_l \times 2^q \quad (8)$$

When fault missing event happens, the following equation always holds.

$$\begin{aligned} (y_e - y)_m &= \pm (x_l \times 2^q)_m \\ &= \pm ((x_l)_m \times (2^q)_m)_m = 0 \end{aligned} \quad (9)$$

Moduli in form of $m = 2^n - 1$ are considered in this paper, then Eq. (9) has the following possible cases:

a. $(x_l)_m = 0$: observing from Fig. 2 that for random inputs x_l with Gaussian distribution, $(x_l)_m$ is uniformly distributed in interval $[0, m - 1]$, so $Prob((x_l)_m = 0) = 1/m$.

b. $(2^q)_m = 0, (x_l)_m \neq 0$: since $(2^q)_m \neq 0$, $Prob((2^q)_m = 0, (x_l)_m \neq 0) = 0$.

c. $(x_l)_m \neq 0, (2^q)_m \neq 0, (x_l)_m \cdot (2^q)_m = km, k \in N$. In this case, $Prob((x_l)_m \neq 0, (2^q)_m \neq 0, (x_l)_m \cdot (2^q)_m = km) = 0$, whose computation process is given by Ref. [20].

Therefore, the fault missing rate of the MIF model is

$$P_{MIF} = \frac{1}{m} = \frac{1}{2^n - 1} \quad (10)$$

Three pairwise relatively prime moduli m_1, m_2, m_3 are used in this paper. It can be observed from Fig. 3 that the following condition must be satisfied when the fault missing event happens.

$$\begin{aligned} &y_1 \neq y_2, (y_1 \equiv y_2) \bmod m_1, \\ &(y_1 \equiv y_2) \bmod m_2, (y_1 \equiv y_2) \bmod m_3. \end{aligned} \quad (11)$$

Namely, according to the above discussion, we have

$$(x_l)_{m_1} = 0, (x_l)_{m_2} = 0, (x_l)_{m_3} = 0. \quad (12)$$

Since m_1, m_2 and m_3 are pairwise relative prime integers, if x_l satisfies $(x_l)_M = 0, M = m_1 \cdot m_2 \cdot m_3$, then the fault missing event occurs. Consequently, the MIF model fault missing rate of the proposed method is reduced to

$$P_{MIF_PROPOSED} = \frac{1}{M} = \frac{1}{m_1} \times \frac{1}{m_2} \times \frac{1}{m_3} \quad (13)$$

In the AIF model, the filter's output is

$$\begin{aligned} y_e = &x_0 \times h_0 + x_1 \times h_1 + \dots x_l \times h_l \\ &+ x_{L-1} \times h_{L-1} \pm 2^q \end{aligned} \quad (14)$$

where 2^q is the error caused by SEU. Based on case b for Eq. (9), we have $(2^q)_m \neq 0$. Thus, the fault missing rate of the AIF model is

$$P_{AIF} = 0 \quad (15)$$

Therefore, the AIF model fault missing rate of the proposed method is

$$P_{AIF_PROPOSED} = 0 \quad (16)$$

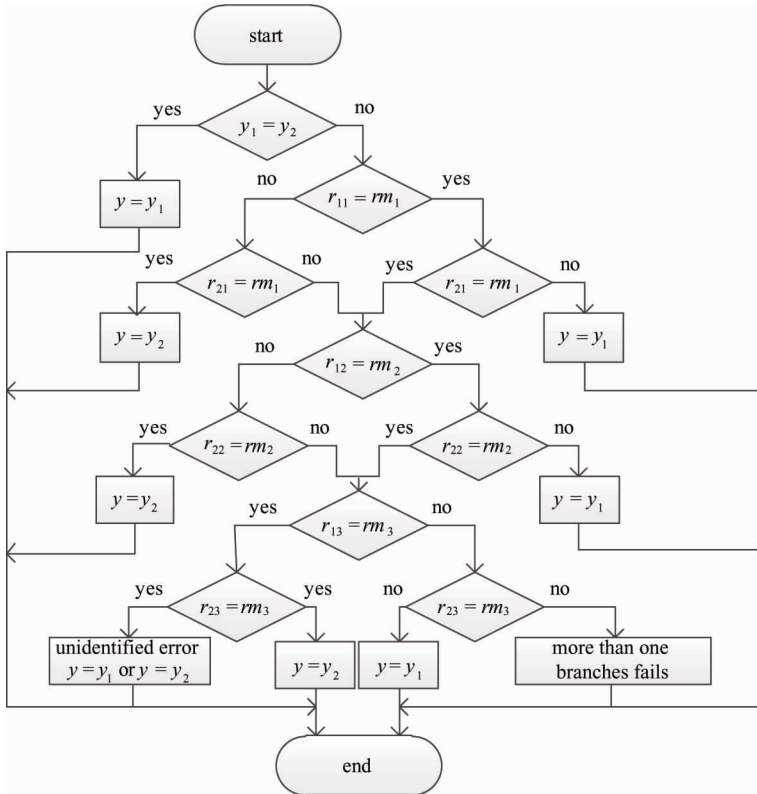


Fig. 3 Flow diagram of the proposed method

2.3 High level synthesis for the proposed structure

Although the structure of multiple residue code branches can reduce the fault missing rate dramatically, it needs more multipliers and adders, which in-

crease the hardware size greatly. In this subsection, a resource-constrained as soon as possible (ASAP) algorithm based HLS is applied to the structure to explore a proper tradeoff between area overhead and delay.

The flow graph of ASAP scheduling is explained

in Fig. 4. The ASAP scheduling is designed in the C++ program. The inputs of the scheduling program are data flow graph (DFG), operator types (such as adder and multiplier) and their numbers. It performs topological sorting to the tasks firstly, and then computes the earliest start time and the latest start time of each task. The tasks whose earliest start time equals to their latest start time are defined as critical tasks, the rest of them are defined as non-critical tasks. In each control step, the critical tasks are distributed firstly, and then the non-critical tasks are distributed if and only if all critical ones have been distributed and there are remainder operators in this control step. The process is repeated until all of the tasks are distributed. The outputs of the program are scheduling results, including the execution sequence of each task.

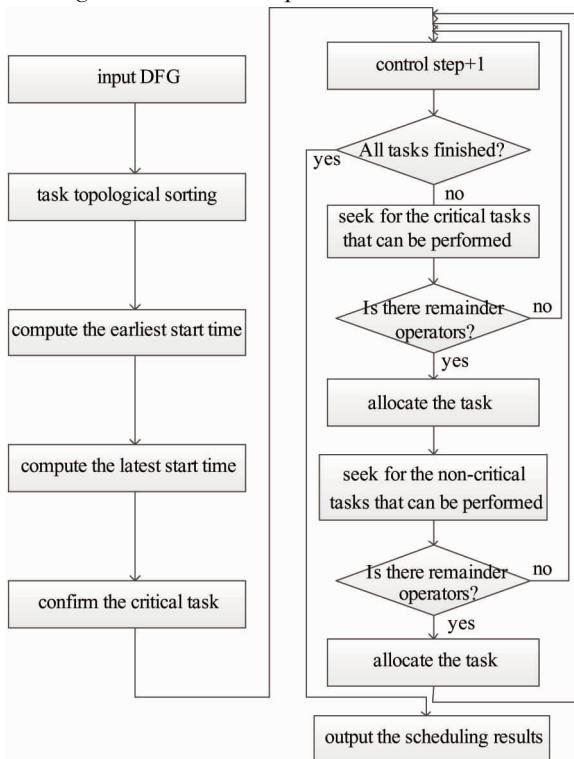


Fig. 4 Flow graph of the ASAP scheduling

To reduce the area overhead of the proposed structure and keep all branches running synchronously, the same scheduling scheme is performed to each branch. Notice that, the area reduction is established on the basis of increase of control steps, which means that a smaller area overhead will lead to more control steps. Therefore, the best tradeoff between them according to the design requirement should be made.

3 Simulation results

In this section, the simulation results of the HLS based DMR-MRD method, including the area reduction and fault missing rate, are presented. A 15-tap FIR filter with 8-bit sample data and 8-bit coefficients is implemented in ISE12.3 for Xilinx FPGA Virtex-4 device.

Firstly, the high level synthesis is applied to the DMR-MRD structure. The DFG of the 15-tap FIR filter is shown in Fig. 5. Observe that direct synthesis produces 16 multipliers and 15 adders, and four control steps are needed. Similarly, the DFG of the residue branches can also be described by Fig. 5, in which the multipliers and adders just need to be replaced by modular multipliers and modular adders, respectively. By applying the resource-constraint ASAP scheduling, different low cost scheduling schemes can be obtained. Some examples of them are given in Table 1. Considering the tradeoff between the delay (control step) and area overhead, the scheduling scheme with 6 multipliers and 6 adders, whose DFG is shown in Fig. 6, is selected. In this case, 6 control steps are required. The scheduling scheme is applied to both of the FIR filter branches and the residue code branches.

The area overheads for the 15-tap FIR filter protected by TMR, SSC-DM-CRC scheme with $m = 3$, $m = 7$, $m = 15$, $m = 31$, $m = 63$, and proposed method are presented in Table 2. Since multiple branches are used in the proposed method, and the total area overhead of three residue code branches may be higher than

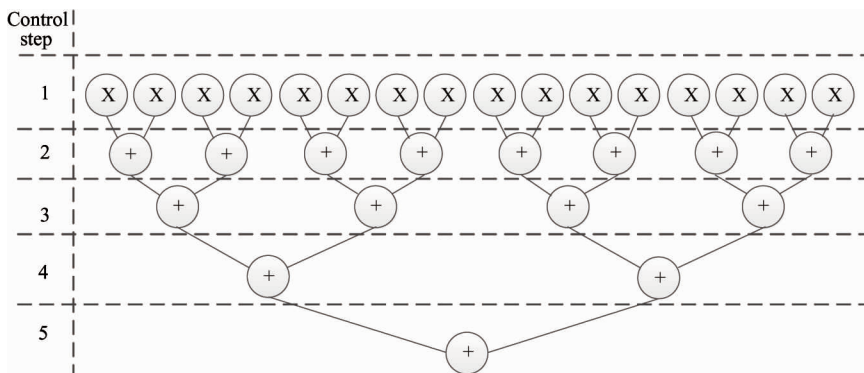


Fig. 5 DFG of 15-tap FIR filter

Table 1 part of the scheduling schemes

Control steps	Multipliers	Adders
6	8	8
6	8	7
7	6	6
7	6	5
8	5	4
10	3	3
12	2	2

one branch of regular filter, especially when the moduli are large integers, the proposed structure generates more area overhead than TMR. However, the resource-constraint HLS cuts down the resource number, and makes the area overhead of the proposed method much lower than that of TMR and SSC-DM-CRC. Notice that, the area reduction has a close relationship with the scheduling schemes.

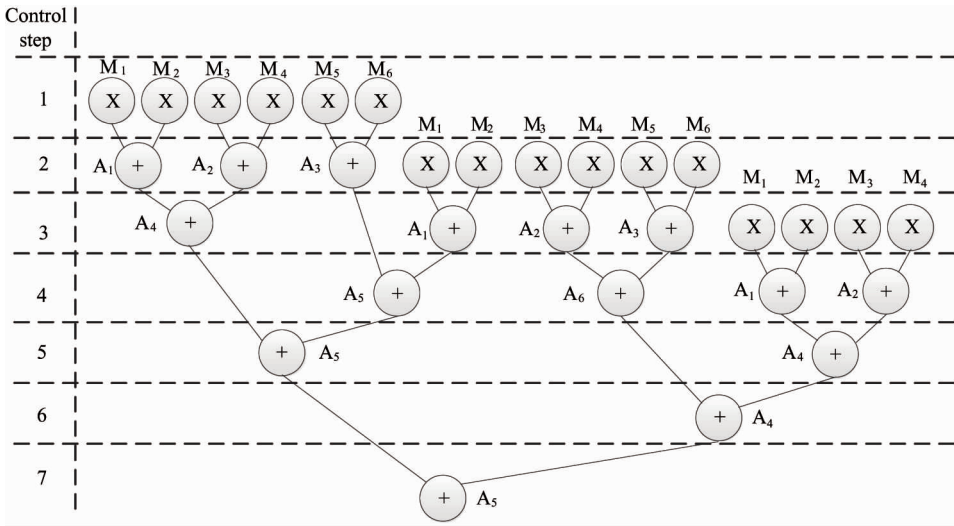


Fig. 6 DFG of 15-tap FIR filter after scheduling

Table 2 Area overhead comparison

	TMR	SSC					Proposed
		$m = 3$	$m = 7$	$m = 15$	$m = 31$	$m = 63$	
Number of slices	1085	742	846	902	1028	1089	656
4-input LUTs	1537	1175	1344	1156	1535	1892	957

To estimate the fault missing rate of the proposed fault-tolerant scheme, 136000 samples are input to the two 15-tap FIR filters M_1 and M_2 , and SEU is injected into the coefficients of M_1 or M_2 randomly. Since the moduli discussed in this paper are pairwise relatively prime integers in the form of $m = 2^n - 1$, $n = 2$, $n = 3$ and $n = 5$ are used, that is $m_1 = 3$, $m_2 = 7$ and $m_3 = 31$. Fig. 7 shows the fault missing rate and the area ratio to TMR of the proposed scheme and the SSC-DM-CRC scheme with $m = 3$, $m = 7$, $m = 15$, $m = 31$, $m = 63$. Notice that for the SSC-DM-CRC scheme, the fault missing rate decreases with the increase of m , with the increasing area overhead. The fault missing rate for the proposed method, is even lower than that of the SSC-DM-CRC scheme with $m = 63$, and the area reduction is higher than that of SSC-DM-CRC scheme with $m = 3$.

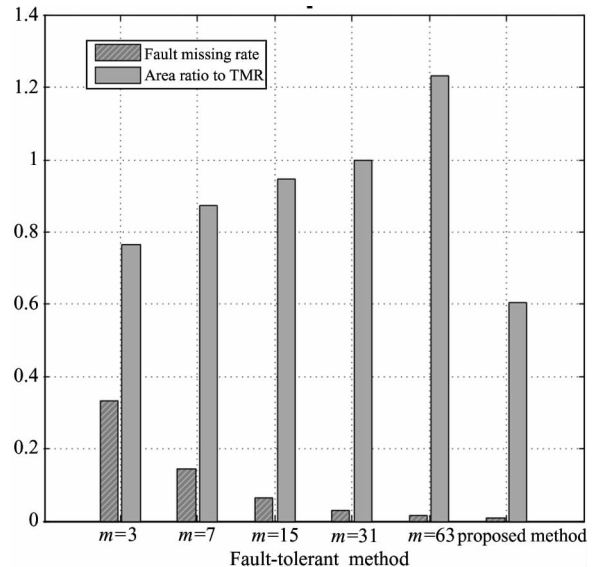


Fig. 7 Fault missing rate and the area ratio to TMR for SSC-DM-CRC and the proposed method

4 Conclusion

Traditional TMR is not suitable for on-board processing (OBP) because of its high area overhead. To cut down the area overhead of TMR without reducing reliability, this paper proposes a new scheme which uses a high-level synthesis (HLS) based dual-module redundancy with multi-residue detection (DMR-MRD). The new scheme reduces fault missing rate dramatically compared to the SSC-DM-CRC scheme. At the same time, it costs less FPGA resources as well. Since the area reduction has a close relationship to the scheduling schemes, the area reduction is not a constant value. The simulation results in this paper show that the area overhead can be reduced by 48.26% with the fault missing rate as low as 0.15% for an applicable scheduling scheme. Although only FIR filters are simulated in this paper, the DMR-MRD fault-tolerant method is also available to other DSPs with the MAC structure, such as FFT and digital beam forming.

Reference

- [1] Jolfaei MA, Jakobs KRA. Concept of on-board-processing satellites. In: Proceedings of Universal Personal Communication, Dallas, USA, 1992. 14.04/1-14.04/4
- [2] Ibnkahla M, Rahman Q, Sulyman A, et al. Safwat A: High-speed satellite mobile communications: technologies and challenges. *Proceedings of the IEEE*, 2004,92(2): 312-339
- [3] Kolawole MO; Satellite Communication Engineering. Marcel Dekker Publishers, 2002. 36-90
- [4] Sunderland D, Duncan G, Rasmussen B, et al. Megagate ASICs for the Thuraya Satellite Digital Signal Processor. In: Proceedings of International Symposium on Quality Electronic Design, San Jose, USA, 2002. 479-486
- [5] Fernando W, Rajatheva R. Performance of COFDM for LEO satellite channels in global mobile communications. In: Proceedings of Vehicular Technology, Ottawa, Canada, 1998. 1503-1507
- [6] Papathanassiou A, Salkintzis A, Mathiopoulos P. A comparison study of the uplink performance of W-CDMA and OFDM for mobile multimedia communications via LEO satellites. *IEEE Personal Communications*, 2001,8(3): 35-43
- [7] Wiemann K. The ACeS digital channelizer-The ACeS digital channelizer-the next generation in regional digital satellite telephone communications. In: Proceedings of the 19th Digital Avionics Systems, Philadelphia, USA, 2000. 8B3/1-8B3/6
- [8] Kumar R, Taggart D, Monzingo R, et al. Wideband gap-filler satellite (WGS) system. In: Proceedings of the IEEE Aerospace, Big Sky, USA, 2005. 1410-1417
- [9] Wang L D, Hamilton B A, Ferguson D. WGS Air-Interface for AISR Missions. In: Proceedings of the IEEE Milcom, Orlando, USA, 2007. 1-7
- [10] Sadowsky J S, Lee D K. The MUOS-WCDMA Air Interface. In: Proceedings of the IEEE Milcom, Orlando, USA, 2007. 1-6
- [11] Ghysel P. Mobile User Objective System (MUOS). Navy Communications Satellite Program Office (PMW 146), 2012
- [12] Litva J, Lo T K. Digital Beamforming in Wireless Communications. Artech House Publisher, 1996
- [13] Chiba I, Miura R, Tanaka T, et al. Digital beam forming (DBF) antenna system for mobile communications. *Aerospace and Electronic Systems Magazine*, 1997,12(9):31-41
- [14] Kastensmidt F L, Carro L, Reis R. Fault-tolerance Techniques for SRAM-based FPGAs. Springer Publisher, 2006. 1-8
- [15] Carmichael C. Application Note: Triple Module Redundancy Design Techniques for Virtex FPGAs (XAPP197 (v1.0.1)). XILINX INC. 2006
- [16] Murray P L, VanBuren D. Single event effect mitigation in re-configurable computers for space applications. In: Proceedings of Aerospace, Big Sky, USA, 2005. 1-7
- [17] Rodriguez-Navarro J J, Gansen M, Noll TG; Error-tolerant FIR filters based on low-cost residue codes. In: Proceedings of IEEE Symposium on Circuits and Systems, Kobe, Japan, 2005. 5210-5213
- [18] Etzel M H, Jenkins W K. Redundant residue number systems for error detection and correction in digital filters. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 1980,28(5):538-544
- [19] Li L, Hu J. Error correcting properties of redundant residue number systems. *IEEE Transactions n Nuclear Science*, 2010,C-22(3):307-315
- [20] Yang W H, Gao Z, Chen X, et al. Residue code based low cost SEU-tolerant FIR filter design for OBP satellite communication system. *Journal on Wireless Communications and Networking*, 2012. 174:1-11
- [21] Tosun S, Mansouri N, Arvas E, et al. Reliability-centric high-level synthesis. In: Proceedings of the Design, Automation and Test in Europe Conference and Exhibition. Munich, Germany, 1258-1263
- [22] Antola A, Piuri V, Sami M. High-level synthesis of data paths with concurrent error detection. In: Proceedings of IEEE International Symposium on Defect and Fault Tolerance in VLSI System, Austin, USA, 1998. 292-300
- [23] Cardarilli G C, Nannarelli A, Re M. Residue number system for low-power DSP applications. In: Proceedings of Asilomar Conference on Signals, Systems and Computers, Pacific Grove, USA, 2007. 1412-1416
- [24] Coussy P, Gajski D, Meredith M, et al. An introduction to high-level synthesis. *IEEE Design and Test of Computers*, 2009,26(4):8-17

Yang Wenhui, born in 1985. She received his B. S. degrees in Electronic Engineering Department of Xiamen University in 2008. She is now studying for a Ph. D in Xiamen University and Tsinghua University. Her research interest is the reliability design for on-board processing satellite communication systems.