

硕士学位论文

JPEG-LS 无损压缩编码器抗 SEU 加固设计

**SEU RADIATION HARDENED BY DESIGN
TECHNIQUE FOR LOSSLESS JPEG-LS IMAGE
COMPRESSOR**

马建宁

哈尔滨工业大学

2019 年 6 月

国内图书分类号：TN406

学校代码：10213

国际图书分类号：621.3

密级：公开

工程硕士学位论文

JPEG-LS 无损压缩编码器抗 SEU 加固设计

硕士研究生：马建宁

导师：霍明学 教授

申请学位：工程硕士

学科：集成电路工程

所在单位：航天学院

答辩日期：2019 年 6 月

授予学位单位：哈尔滨工业大学

Classified Index: TN406

U.D.C: 621.3

Dissertation for the Master Degree in Engineering

**SEU RADIATION HARDENED BY DESIGN
TECHNIQUE FOR LOSSLESS JPEG-LS IMAGE
COMPRESSOR**

Candidate :	Ma Jianning
Supervisor :	Prof. Huo Mingxue
Academic Degree Applied for :	Master of Engineering
Speciality :	Integrated Circuit Engineering
Affiliation :	School of Astronautics
Date of Defence :	June, 2019
Degree-Conferring-Institution :	Harbin Institute of Technology

摘要

随着航天事业的发展,星载图像捕捉设备分辨率日益提高,图像数据信息量不断地增大,图像压缩编码器在航天图像处理应用中越显重要。图像压缩编码器需要依照一定标准进行执行,JPEG-LS (Joint Photographic Experts Group-Lossless) 是一种主流的图像无损/近无损压缩标准。它不仅有良好的压缩效果,更重要的是具有运算复杂度低、便于硬件实现的特点,节省硬件资源,更适合用于航天设备中。空间环境中存在大量的辐射粒子,工作在其中的图像压缩编码器易发生单粒子翻转效应 (Single Event Upset, SEU), 最终导致整幅图像数据压缩错误而无法恢复出原图像。因此需要对图像压缩编码器进行抗 SEU 加固设计,从而提高编码器的可靠性,降低图像的失真程度。

本文首先介绍了图像压缩的基本理论知识。接着对 JPEG-LS 标准进行了分析,并重点对无损压缩的流程做了深入研究。在此基础之上,设计了 JPEG-LS 无损压缩编码器的硬件结构。提出了以下优化方案:采用有限状态机 (Finite State Machine, FSM) 的方式实现选择常规与游程模式的功能;利用简单的移位寄存器即可完成码流拼接任务,节省硬件资源;设计输入控制器配合编码器工作,采用流水线技术对常规模式编码器处理,提高了编码速率。最终编码器可对 512×512 8bit 灰度图像正确压缩。

综合考虑速度、面积开销及编码器结构特点后,从以下几个方面提出了加固方案:考虑到模式选择器内的 FSM 在整个设计中的重要地位,使用独热码 (One-Hot) 结合三模冗余的方式对其加固;利用三模冗余和超时检测技术对核心编码器加固;采用 EDAC (Error Detection And Correction) 检纠错技术对参数更新模块加固;根据 JPEG-LS 编码器特点,采用帧间图像独立处理的方式减少 SEU 的累积。

本文通过仿真验证了编码器功能的正确性及加固方案的有效性,并利用故障注入工具对编码器整体的加固效果进行了评估,结果显示加固后的编码器整体错误率降低 19.48%。最后本文利用 Xilinx 的 FPGA 对设计进行了硬件实现,验证了编码器功能的正确性。

关键词: JPEG-LS; 无损图像压缩; 软错误; 抗 SEU 加固

Abstract

With the development of aerospace industry, the amount of image data information is increasing. Image compressor is becoming more and more important in the application of space image processing. Image compressor needs to be executed in accordance with certain standards, JPEG-LS (Joint Photographic Experts Group-Lossless) is a mainstream image lossless/near lossless compression standard. It not only has good compression effect, but also has the characteristics of low computational complexity, saving hardware resources and being more suitable for use in aerospace equipment. There are a large number of radiation particles in the space environment. The image compressor working in the space environment suffers from Single Event Upset (SEU), which eventually leads to the whole image data compression error and cannot recover the original image. Therefore, it is necessary to design radiation hardened image compressor to reduce the distortion of the image.

In this paper, firstly, the basic theory of image compression is introduced. After that, the JPEG-LS standard is analyzed, and the process of lossless compression is studied in details, based on which, the hardware structure of JPEG-LS lossless compressor is designed. The following optimization schemes are proposed: the function of choosing regular mode or run mode is implemented by using Finite State Machine(FSM); Simple shift register is used to complete the task of code stream splicing and save hardware resources; The input controller is designed to cooperate with the encoder, and the pipeline technology is used to process the regular mode encoder, which improves the compression efficiency. Finally, the encoder can compress 512×512 8 bit gray image correctly.

Considering the speed, area overhead and the structure characteristics of the encoder, the harden methods are proposed from the following aspects: Using One-Hot and three mode redundancy (TMR) to harden mode selector; Hardening core encoders with TMR and timeout detection techniques; Hardening the parameter update module with EDAC; Compression independently to reduce SEU accumulation.

Simulation results shows that the encoder function and hardening methods proposed in this paper are correct and effective. The hardening methods is assessed by Fault Injection Tool. The result shows that the overall error rate of the hardened compressor is reduced by 19.48%. Finally, the design is implemented by using Xilinx's FPGA, which verifies the correctness of the compressor.

Keywords: JPEG-LS, lossless image compression, soft error, SEU radiation harden

目 录

摘 要	I
ABSTRACT	II
第 1 章 绪 论	1
1.1 课题背景及研究目的和意义	1
1.2 国内外研究现状	2
1.2.1 JPEG-LS 研究现状	2
1.2.2 抗 SEU 加固设计研究现状	4
1.3 论文研究内容	6
第 2 章 图像压缩理论与 JPEG-LS 压缩标准研究	8
2.1 图像压缩基本理论	8
2.1.1 图像压缩信息论基础	8
2.1.2 主要图像编码技术	9
2.2 JPEG-LS 图像压缩标准研究	10
2.2.1 上下文建模	11
2.2.2 常规模式编码	12
2.2.3 游程模式编码	15
2.3 本章小结	16
第 3 章 JPEG-LS 压缩编码器设计与实现	17
3.1 图像预处理模块设计	17
3.2 像素输入控制器模块设计	18
3.3 基于 JPEG-LS 的无损压缩编码器设计	20
3.3.1 模式选择器设计	22
3.3.2 常规模式编码器设计	23
3.3.3 游程模式编码器设计	30
3.3.4 参数更新模块设计	32
3.3.5 输出码流控制器设计	33
3.4 本章小结	34
第 4 章 编码器抗 SEU 加固设计	35
4.1 模式选择器的抗 SEU 加固设计	35
4.1.1 有限状态机编码	35

4.1.2 有限状态机加固	36
4.2 核心编码器的抗 SEU 加固设计	37
4.2.1 三模冗余技术	37
4.2.2 超时检测技术	38
4.2.3 编码器加固	39
4.3 参数更新模块的抗 SEU 加固设计	39
4.3.1 汉明码检纠错技术	40
4.3.2 参数更新模块的 EDAC 设计	40
4.4 帧间图像独立处理	42
4.4.1 独立处理分析	42
4.4.2 复位信号设计	42
4.5 本章小结	44
第 5 章 功能验证与加固效果分析	45
5.1 常态工作模式功能仿真验证	45
5.1.1 仿真验证平台设计	45
5.1.2 功能验证结果	46
5.2 抗 SEU 加固测试	47
5.2.1 加固仿真测试	47
5.2.2 加固效果评估	50
5.3 FPGA 验证	51
5.3.1 FPGA 硬件平台搭建	51
5.3.2 结果验证与性能分析	52
5.4 本章小结	54
结 论	55
参考文献	56
攻读硕士学位期间发表的论文及其它成果	60
哈尔滨工业大学学位论文原创性声明和使用权限	61
致 谢	62

第1章 绪 论

1.1 课题背景及研究目的和意义

随着互联网技术的发展,信息交互的应用场景变得越来越常见。图像是人类感知世界、认识世界的视觉基础,是人类获取信息、表达信息的重要载体之一。统计资料表明,人类视觉系统获取到的信息可以达到总信息量的 70%^[1]。一方面,特定情况下图像信息的获取更加直接、方便;另一方面,图像表达出的信息内容更加直观,便于双方接受。图像作为信息传播载体的重要性日益提高,对图像处理技术的要求也越来越高。随着航天事业的发展以及航天应用技术的提高,数字图像处理技术尤其是图像压缩技术的需求量正逐渐增大。

我国航空航天事业飞速发展,如今俨然已成为一个航天大国,正在朝航天强国的目标迈进。航天技术的提高促进了信息科学、生物科学、材料科学等多个学科的进步,为人们带来了生活便利的同时,也为国防事业提供了安全保障。但是在迈向航天强国的道路上,航天技术的发展也面临着各种挑战。无论是探索宇宙星体奥秘还是观测地球资源分布等等,分辨率日益提高的星载图像捕捉设备为科学研究发现提供了有力的保障,但也表现出了数据量大的特点^[2]。传统的处理方式是将未处理的遥感图像信息传输到地面后再进行处理,这样就对星载存储器与图像传输带宽提出了更高要求,如果在星载系统中利用图像压缩算法将图像数据进行压缩处理,那么将很好地解决这个矛盾^[3]。图像信息数据是由有用数据与冗余数据构成的,图像压缩可以减少甚至清除冗余数据。在不损坏图像有用数据的情况下,利用比较少的数据表现图像信息,从而实现图像数据量减少,以达到利用有限星载资源存储、传输图像信息的要求。

无损压缩要求将有效的信息全部留存,不可以损失,存在压缩率低的缺点;有损压缩可以允许丢失一定的信息,具有压缩率高的优点。在航天图像处理领域,图像信息经过压缩后恢复出的图像不能丢失有效信息,这种情况下,无损压缩方法是星载图像处理设备的第一选择。此外,考虑到星载图像处理系统构造复杂、造价昂贵,功耗、体积、重量都受到严格限制,需要选择一种简单高效、便于硬件实现、压缩率比较高,且图像不易失真的图像压缩标准。

JPEG-LS 是一种契合上述需求的无损/近无损压缩算法。它是 ISO/ITU 新标准

中的一种图像压缩算法^[4],是基于 HP(惠普)实验室 Weinberger 等人提出的 LOCO-I (Low Complexity Lossless Compression for Image) 的一种压缩方法^[5,6]。与 JPEG2000、SPIHT 等算法相比较, JPEG-LS 算法不需要进行离散余弦变化、不使用算术编码等复杂运算,算法运算过程复杂度低,硬件实现较为简单^[7,8],对各类图像均具有良好的适用性,因而通常被应用于航天图像处理、生命科学等领域^[9,10]。

太空中复杂的电磁和辐射环境,导致电子电路系统产生各类辐照效应。对于图像压缩编码器来讲,单粒子翻转效应对其危害程度不容小觑。在压缩编码器进行压缩过程中,易受到各种粒子的影响而引起单粒子翻转效应,从而导致编码器逻辑状态跳变,引发可恢复的错误。如果星载图像压缩编码器在工作过程中发生逻辑错误,就极大可能出现压缩后的星地传输图像数据重要内容丢失,甚至不能恢复出原图像的问题。进行冗余设计来降低造成错误的概率是一种主流加固方式,也可以根据编码器的具体功能加入其它保护电路减少逻辑错误对功能的影响。通过进行 JPEG-LS 编码器设计实现及抗 SEU 加固的研究,可以提高星载图像压缩编码器输出码流的正确性,保证其能够恢复出原图像,为科学研究提供真实可靠的高清图像。因此,开展基于 JPEG-LS 的图像压缩编码器实现及抗 SEU 加固设计研究具有十分重要的意义。

1.2 国内外研究现状

1.2.1 JPEG-LS 研究现状

图像压缩算法是图像处理过程中的重要组成部分之一。当前,主流的国际图像压缩标准主要有 JPEG 标准、JPEG2000 标准和 JPEG-LS 标准。

1998 年,静止图像国际压缩标准的大家庭中添加了一位新成员 JPEG-LS,随后它便越来越多地出现在人们视野中。1999 年,该标准的第一部分面世,2003 年,标准的第二部分对外公开,且引入了算法编码。由于 JPEG-LS 算法压缩性能的优越性,逐渐取代了 JPEG 标准的无损压缩部分,自其诞生之日起便引起了国内外很多领域的关注。2000 年, M. Weinberger 探讨了 LOCO-I 设计的基本原理,并将其标准化为 JPEG-LS^[11]。Z Wang 等人提出采用并行处理技术设计 JPEG-LS,采用 n 个基本处理单元(EPU)来同时压缩 n 个图像,将 JPEG-LS 组成中各个子模块作为基本处理单元。结果表明,这样设计的电路吞吐量提升 $1.2 \times n$ 倍,相同吐出量时,消耗的硬件资源较原始电路减少 25%^[12]。S. Bedi 等人提出 JPEG-LS 简单的边缘检

测技术会在对角线边缘的位置产生较大的预测误差,通过改进该预测技术,可以精确检测对角线边缘并有效预测像素,最终结果表明可以降低预测误差达 15%^[13]。2013 年, LihJen Kau 等人提出对编码器采用全流水线结构设计,分为 13 个阶段,最终在 Altera SOPC 平台上实现的时钟频率可达到 113.03MHz^[14]。2017 年, Yakup Murat Mert 提出通过改进预处理单元来提高 JPEG-LS 的吞吐量的方法,同时又能保持算法本身所具有的低复杂度的优势,并且利用重新映射算法提高了图像的压缩效果,最终结果表明使用很少的硬件资源便可以对高分辨率图像进行实时压缩^[15]。

自 JPEG-LS 压缩算法提出以来,国内也对其进行了诸多研究,而且在卫星遥感领域较为突出。关于遥感图像高保真高速实时的问题,2001 年,北京航空航天大学利用 JPEG-LS 压缩技术获得了一定研究成果,主要是运用该算法原理对星载图像捕捉设备捕获的图像进行压缩处理^[16],与原始算法没有太大区别。刘荣科等人指出 JPEG-LS 无损压缩算法采用预测编码方式,压缩结果像素间相关性大,单个像素点压缩结果错误导致后续码流无法正常解码^[17]。2010 年,牛慧卓等人实现了 JPEG-LS 压缩编码器的设计,对硬件运行过程中的缓存设置、定时控制和边界点问题进行了处理,缺点是需要一定的纠错方法保证编码器的准确性,但是表明了 JPEG-LS 无损压缩算法在图像处理领域有一定的优势^[18];太原理工大学的戴鑫利用 C 语言对 JPEG-LS 无损压缩编码器和解码器进行了实现,并且以此为参考模型实现了编码器的硬件设计,将其封装成了 IP 核,编码器运行频率为 100MHz^[19]。2016 年,海南大学寇计萍提出将 JPEG-LS 无损压缩与加密算法相结合,完全重现原始图像的同时不影响算法的压缩性能,并且从多个方面对 JPEG-LS 编码器进行加密研究,最终从压缩率、密钥灵敏度、信息熵等方面说明了方案的可行性以及对信息的保密^[20]。2017 年,阙恒等人在分析 JPEG-LS 原理和强像素级依赖的基础上,提出可以利用帧缓存分块和推迟上下文更新的方法来提高压缩速率,减少压缩延迟,结果表明压缩所用时钟周期明显降低^[21]。

通过对国内外 JPEG-LS 算法研究现状的分析发现,自 JPEG-LS 标准公布以来,无论是国内还是国外,研究者们都对该算法的实现做了大量的研究。针对 JPEG-LS 算法硬件实现过程中存在的反馈延时问题,有研究者提出了改进的 JPEG-LS 算法来解决延时造成的算法时钟频率低的问题。针对 JPEG-LS 算法其中的部分顺序处理结构采用流水线方式提高该算法的处理图像数据的速度。采用分块和推迟上下文更新的方式减少压缩延迟。国内研究者又将 JPEG-LS 算法与混沌系统相结合,

实现了压缩图像数据的保密性。从相关文献^[7-9]中总结提炼出 JPEG 标准、JPEG2000 标准、JPEG-LS 标准三种常见的图像压缩算法优缺点及应用领域如表 1-1 所示。通过比较分析可知，JPEG-LS 压缩效果比无损的 JPEG 标准好。尤其是 JPEG-LS 算法具有复杂度低、便于硬件实现的特点，将节省星载系统硬件资源，因此 JPEG-LS 适合用于星载图像压缩编码器，但是其抗差错能力还有待解决，压缩码流出现单个像素点错误就会导致后续码流无法恢复出原图像^[17]。

表 1-1 三种图像压缩标准比较

	JPEG	JPEG2000	JPEG-LS
优点	压缩比较高； 可部分压缩图像；	压缩比高； 支持无损及有损压缩； 码率可调	算法复杂度低； 支持无损及近无损压缩； 适用于各类图像； 计算时间消耗低；
缺点	无损压缩模式特别不理想；误差修复能力弱	算法复杂度高；	码率不可控；
应用领域	彩色传真、可视通讯 印刷出版、新闻图片	图像打印、彩色传真 医学图像、遥感图像	遥感图像、医学图像 无线通信、网络传输

1.2.2 抗 SEU 加固设计研究现状

随着航天事业的发展，世界各个机构对航天用电子元器件的可靠性越来越重视，尤其是对器件的抗单粒子翻转加固设计做了深入的研究。

抗单粒子翻转加固设计可以分为在防护层抗辐照加固（SL-RHBD: Shielding Level Radiation Hardened By Design）、器件层抗辐照加固（DL-RHBD: Device Level Radiation Hardened By Design）和应用层抗辐照加固（AL-RHBD: Application Level Radiation Hardened By Design）等三个层面上进行实现^[22]。防护层层面，主要方法是为设计完成的电路或者系统等加入抗辐照屏蔽罩，降低空间粒子直接射入功能单元的可能性，从而达到抗单粒子翻转的效果，这方面的研究，国内外各个机构都涉及的比较少，且防护层抗辐照加固措施的性能有限^[23,24]。

器件层抗辐照加固是从逻辑设计、工艺和材料角度采取一系列措施防止单粒子翻转的加固方法。可通过设计特殊的器件结构、使用具有可靠性能高的材料或者利用特殊工艺使器件达到抗单粒子翻转的目的。当前，在工艺方面考虑器件级抗辐照加固的方式是采用绝缘体上硅（SOI: semiconductor on insulator）工艺^[25]。

在逻辑设计中考虑器件级抗辐照加固的方法一般有以下几种：降低存储节点对单粒子翻转的敏感度、采用冗余技术、存储器检纠错设计等。美国德州农工大学设计了一种新的 CMOS 标准单元库，该标准单元库经过特殊设计后具有一定的抗辐照能力，利用它替换掉原有的标准单元库之后具备了抗单粒子翻转的能力，结果表明虽然面积比原来增大了 62%，延迟增加 29%，但是该方法却可以减少 90% 的软错误率^[26]。Shayan M 提出一种新的 SRAM 基本单元结构，由 12 个基本单元构成，结果表明该结构具有很好的抵抗单粒子翻转效应的效果^[27]。A. Lopes Filho 等人向 JPEG-LS 编码器内进行故障注入，指出了 JPEG-LS 无损压缩编码器敏感部位^[28]，为抗 SEU 加固研究奠定了基础。

在图像压缩编码器的抗辐照研究方面，主要是考虑在器件逻辑设计层面进行抗辐照加固设计，主流的加固方式是采用冗余设计^[29-31]。西安电子科技大学谭新宇提出了采用单帧独立处理方法、超时检测处理方法和冗余保护处理方法对 JPEG2000 算法硬件实现进行抗辐照设计^[32]，结果表明以上方法可以有效减少辐照效应对压缩过程的影响。王波提出采用图像源异常处理、冗余处理及图像单帧处理的方法对 SPIHT 算法进行抗辐照设计^[33]，经过验证上述方法提升了编码器的抗单粒子效应的能力。北京理工大学董珊介绍了星载图像压缩编码器抗辐照加固可以从工艺级、器件级、电路结构级和系统容错级进行设计，对专用芯片从功能逻辑电路、时钟电路、复位电路和片内存储电路等方面分别设计了加固处理方案，加固效果显著^[34]。西安电子科技大学赵静介绍了前端可靠性设计的方案，并对 JPEG2000 算法进行三模冗余等加固方式，实现了对图像压缩编码器的加固，抗辐照作用显著^[35]。Tiwari Anurag 和 A. Tomko Karen 提出了状态机状态编码加固设计，可以有效的解决状态机受到单粒子效应而自动进行状态转移的问题^[36]。针对图像压缩编码器容易受到单粒子翻转效应的问题，S. Baloch 等人提出了相应的检测与加固方法：时间冗余设计^[37]。

应用层抗辐照加固主要是需要设计者从电子系统的整体结构出发，设计一些抗辐照加固措施。应用层抗辐照加固的核心思想是，允许器件或者系统发生单粒子翻转现象的发生，当发生错误之后，通过应用层硬件、软件的设计，比如配置回读检测和刷新等措施，避免或者消除单粒子翻转造成的影响^[38]。

此外，通过对国内外抗 SEU 加固设计研究现状的分析发现，国内外研究者对 FPGA 都进行了深入研究，分析了 FPGA 受辐照效应的原理及方式，并且给出了相应的解决办法。美国航空航天局在 FPGA 辐照研究领域处于世界先进水平^[39]，SEE

联盟（由 NASA 和 Xilinx 公司研究员组成）专门针对辐照效应设计了可靠性能高的宇航级 SRAM 型 FPGA，但是目前他国无法从正常渠道获取这种高等级器件，国内在此方面较为落后。

总结国内外关于 JPEG-LS 和抗 SEU 加固设计的研究情况，国内外学者大多研究 JPEG-LS 算法执行效率、压缩码率和安全性问题，鲜有对 JPEG-LS 压缩编码器的抗 SEU 加固设计有所涉及，一旦 SEU 导致单像素点压缩码流错误，后续码流将无法恢复原图像。有研究者利用三模冗余等技术对诸如 JPEG2000 编码器、SPIHT 编码器进行了抗辐照加固设计的研究，取得了不错的研究成果。但是，以上图像压缩算法复杂度高，硬件实现将耗费很大的星载硬件资源，而且编码器功耗高，消耗星载系统的能源，不适合用于航天图像处理设备。JPEG-LS 相较以上算法具有硬件实现复杂度低、功耗低等特点，符合航天用星载无损图像压缩编码器的要求。因此，本文将对 JPEG-LS 无损压缩编码器进行抗单粒子翻转加固设计。

1.3 论文研究内容

本文的主要任务是设计并实现基于 JPEG-LS 算法抗 SEU 加固编码器。JPEG-LS 是一种基于自适应预测、上下文模式和 Golomb 编码算法的无损/近无损的图像压缩算法。本文中编码器采用无损压缩模式。

论文的主要研究内容如下：

(1) 图像预处理模块及输入控制器模块设计。利用 MATLAB 工具对 512×512 8bit 灰度图像进行预处理，将 bmp 格式的图像转化成二进制码流文件。采用 Verilog HDL 语言设计输入控制器模块，将获得的二进制数据输送至 JPEG-LS 编码器中，要求该模块与 JPEG-LS 算法的执行流程相契合，从而保证编码器能够及时地、正确地获得图像信息进行压缩。

(2) 基于 JPEG-LS 的无损压缩编码器设计。设计实现模式选择器，控制图像数据输送到常规模式编码器或游程模式编码器中；常规模式编码器采用合适的流水线、并行处理方案，实现速率较快、面积较小的电路；解决编码过程中的众多参数更新问题，避免反馈逻辑对编码速率造成的影响；设计实现输出码流控制器，保证输出码流拼接结果正确。对 512×512 的 8bit 灰度图像进行压缩，分析比较确认整体功能的正确性。

(3) 编码器抗单粒子翻转加固设计。研究 JPEG-LS 算法的图像压缩流程及辐照效应产生的机理，基于已完成的 JPEG-LS 编码器，利用冗余技术、EDAC 技术及

独热码编码等技术对模式选择器、常规模式编码器、游程模式编码器、参数更新模块等各个模块加固。

(4) 抗 SEU 编码器功能验证与性能评估。搭建编码器仿真验证平台，验证编码器输出结果的正确性。仿真模拟单粒子翻转对编码器的影响，再次对加固后的编码器进行仿真，验证加固方案的有效性。利用故障注入工具对编码器整体加固效果进行评估。利用 Vivado 软件进行综合与布局布线后，生成比特流，在 FPGA 上验证加固前后编码器的设计。

第2章 图像压缩理论与 JPEG-LS 压缩标准研究

本章将通过对图像压缩理论的简单介绍，系统地梳理图像压缩原理与图像编码技术。重点论述 JPEG-LS 国际图像压缩标准，对其上下文建模、常规模式编码、游程模式编码过程进行深入研究。根据本文编码器的需求，确定了各个参数的具体取值范围，为后续的编码器硬件设计与实现奠定基础。

2.1 图像压缩基本理论

2.1.1 图像压缩信息论基础

图像是信息的重要表现形式之一，具有直观、形象的特点。但是，图像中总是或多或少的含有冗余信息。当前，信息进入爆炸式增长时代，图像数据信息更是越来越丰富。图像压缩编码技术是一种减少图像信息冗余的办法，是数字图像处理的重要手段之一。图像经过压缩去除冗余信息之后，在保持原有信息的同时，仍然能够传递足够的信息量，这一点在航天军工应用中尤为重要。

关于数字图像信息中的冗余，一般可以分为：编码冗余、像素间冗余、视觉冗余等^[40]。编码是用来表示时间集合或者信息实体的符号系统（比特、符号、字母和数字等）^[41]。由于人眼的视觉特性，图像信息数据并不能被人类的眼睛全部捕捉到。人类的视觉系统对图形中的内部细节、色度的变换等相对不敏感，这样就形成了视觉上的冗余。如果当前图像中像素点的数据位宽比图像的香农熵值大，那么图像中存在着编码冗余。利用算术编码、Golomb 编码和 Huffman 编码可以消除编码冗余。像素间冗余是由于像素阵列中每个像素值类似或取决于相邻像素，信息没有必要的重复出现了，也称为空间冗余。

香农第一定理是通信基础理论之一，实现了利用概率对信息进行定量计算。在香农理论中，将信源 X 记为一个符号变量， $\{x_1, x_2, \dots, x_n\}$ 是它的所有取值范围。那么， x_i 的信息量 $I(x_i)$ 可以用以下公式求得：

$$I(x_i) = -\log_n p(x_i) \quad (2-1)$$

$p(x_i)$ 表示信源变量为 x_i 的概率。一般情况下，公式（2-1）中的 n 值取 $2^{[42]}$ 。如果符号变量出现的概率具有相互独立性，也就是信源 X 为无记忆信源时，那么

该信号变量总体的信息量可由以下公式求得：

$$H(X) = - \sum_{i=1}^n p(x_i) \log_n p(x_i) \quad (2-2)$$

$H(X)$ 表示信号的总信息量，也称为图像信源 X 的熵， $H(X)$ 的单位是 bit/pixel。

如果符号变量出现的概率具有某种统计相关性，也就是信源 X 为有记忆信源，假设一个符号出现的概率与前面 N 个符号有关，则称该信源为 N 阶马尔科夫信源^[43]。这种情况下，信号关于 X 的总体信息量可以由以下公式求得：

$$\begin{aligned} H(X) &= - \sum_{x^N} p(x_{i_1}, x_{i_2}, \dots, x_{i_N}) H(x | x_{i_1}, x_{i_2}, \dots, x_{i_N}) \\ &= - \sum_{x^N} \sum_j^M p(x_{i_1}, x_{i_2}, \dots, x_{i_N}) p(x_j | x_{i_1}, x_{i_2}, \dots, x_{i_N}) \log_n p(x_j | x_{i_1}, x_{i_2}, \dots, x_{i_N}) \end{aligned} \quad (2-3)$$

其中， $i_n(n=1,2,\dots,N)$, $j=1,2,\dots,M$ 。 $H(X)$ 为该 N 阶马尔科夫信源的条件熵。数据的冗余可以由以下公式来定量地计算：

$$R = H_0 - H(X) \quad (2-4)$$

公式 (2-4) 中， H_0 为决策量， $H(X)$ 为信源熵。

2.1.2 主要图像编码技术

图像压缩主要利用数据中的冗余性进行压缩，假设压缩过程中导致数字图像失真，那么该压缩编码属于有损压缩编码。压缩过程中图像不失真，那么该压缩编码属于无损压缩编码。编码的过程实际上是一种信息集到字符集合的映射过程。对于无损压缩编码来讲，该编码的映射是可逆的，即压缩编码的逆过程解压的结果将得到与原来一样的图像，不会出现图像信息丢失的问题。

熵编码属于无损压缩编码之一，其编码的平均码长最小值就是信源熵的值。算术编码 (AC)、行程编码 (RLC) 和哈夫曼 (Huffman) 编码都属于熵编码。算术编码的编码方式是统计 0 与 1 的间隔值，并用其表示信源编码。行程编码是针对有顺序排列的多次重复数据的一种压缩方法，使用具体的累计数值替换相同的数据值，如图 2-1 所示的数据，其经过行程编码之后得到的结果是：3150628325。

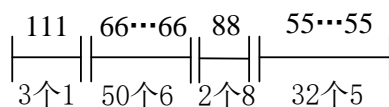


图 2-1 行程编码

预测编码是根据图像局部区域的像素值具有高度相关性的基础而产生的一种编码方式。扫描获取当前像素点周围的几个像素值，通过一定的逻辑运算过程推算出当前像素点的值即为预测值，利用实际值减去预测值，对得到的结果进行编码。由于一般情况下，预测值与实际值相差不大，因此二者的差值趋近于零，从而达到压缩的目的。

变换编码一般是由于空间域中的像素之间相关性很大，它们经过映射变换到正交矢量空间中之后，多数能量主要分布在低频区域。通过编码去除某些能量很小的高频区域，或者在量化时对方差较小的分量分配较少比特数达到压缩的目的。

2.2 JPEG-LS 图像压缩标准研究

JPEG-LS 是一种新的国际图像压缩标准，主要应用于静态图像的无损/近无损压缩。本文主要利用它的无损压缩模式进行编码器设计，它具有压缩率高、恢复重建后图像质量好、实现复杂度低的特点^[11]。JPEG-LS 没有采用传统的变换编码的方式达到图像压缩的目的，而是将当前待压缩像素周围的几个邻近像素作为上下文，从而预测误差，最终通过 Golomb 编码的方式得到最终的压缩码流。JPEG-LS 的图像压缩过程并不是单一的，它分为常规模式编码和游程模式编码两种方式。其中，常规模式编码过程涉及了预测编码方法、Golomb 编码；游程模式编码过程涉及了行程编码方法和 Golomb 编码。JPEG-LS 编码的基本结构图如图 2-2 所示。

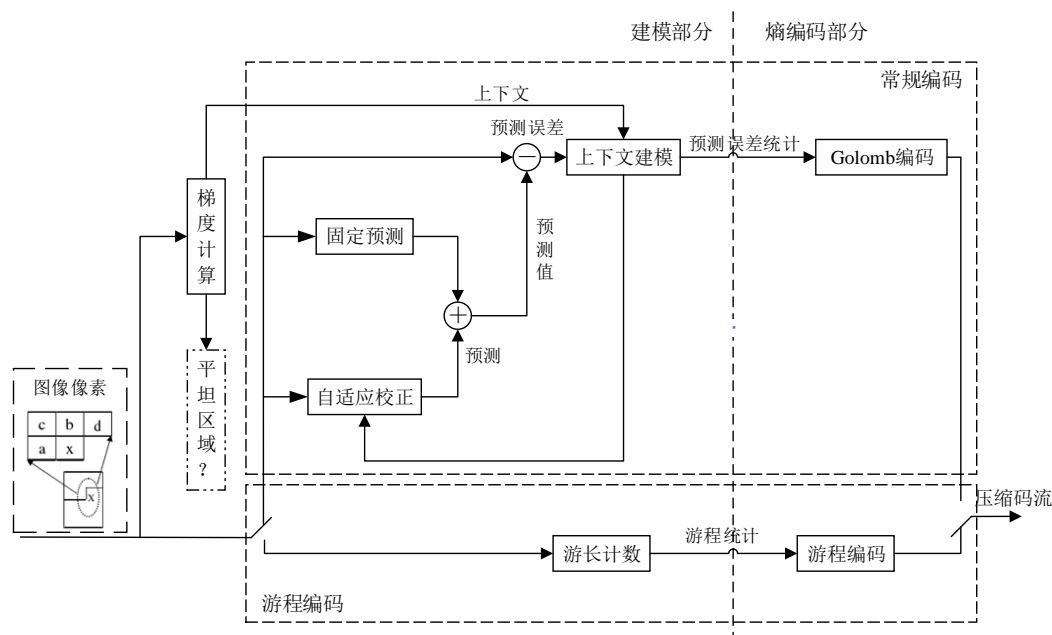


图 2-2 JPEG-LS 图像压缩标准结构图

JPEG-LS 首先判断待压缩图像是否位于平坦区域（像素点 A、B、C、D 的局部梯度值小于 *Near*），如果属于平坦区域，则选择游程模式进行编码，否则选择常规模式编码。接下来，将详细介绍 JPEG-LS 的编码过程。

2.2.1 上下文建模

图像是由像素矩阵组成的，根据 JPEG-LS 图像压缩标准的特点，JPEG-LS 将同时获取到 5 个像素点的像素值，因此引出了“上下文”的概念。5 个像素点位于像素矩阵中的相对位置如图 2-3 所示。其中，像素点 X 为当前待压缩像素点的位置，像素点 A、像素点 B、像素点 C、像素点 D 都是像素点 X 周围相邻的像素点。如果像素点 X 恰好位于像素矩阵的边缘，那么像素点 A、B、C、D 不位于像素矩阵中，需要进行数据补充处理。

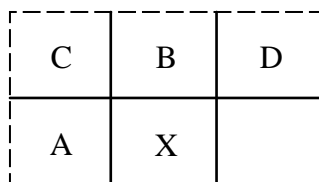


图 2-3 像素点相对位置图

上下文建模的目的是推算条件概率，过程包括确定上下文、推算条件概率、上下文量化三个部分^[43,44]。在 JPEG-LS 标准中，上下文建模主要分为两个部分：

(1) **梯度推导** 当获取到 5 个像素点的数据后，首先进行局部梯度的计算。*Ra*、*Rb*、*Rc*、*Rd* 分别为周围点的像素值，依据以下公式计算得到梯度值：

$$\begin{aligned}
 D_1 &= Rd - Rb \\
 D_2 &= Rb - Rc \\
 D_3 &= Rc - Ra
 \end{aligned} \tag{2-5}$$

计算得到局部梯度的值 D_1 、 D_2 、 D_3 ，可以反映出像素点 X 周围的图像的平滑程度、边界性问题。

(2) **模式选择** 在模式选择阶段，JPEG-LS 将根据计算得到的 D_1 、 D_2 、 D_3 的值，判断像素 A、B、C、D、X 将进入常规模式还是游程模式。*Near* 为失真度控制参数，如果 D_1 、 D_2 、 D_3 的绝对值不大于 *Near* 值，则选择游程模式编码，反之选择常规模式编码，伪代码如图 2-4 所示。

```

if ((abs(D1) <= Near) & (abs(D2) <= Near) & (abs(D3) <= Near))
    select run mode
else
    select regular mode
    
```

图 2-4 模式选择伪代码

若 $Near$ 值为 0, 则 JPEG-LS 为无损压缩编码模式; 若 $Near$ 大于 0, 则 JPEG-LS 为近无损压缩编码模式, 而且随着 $Near$ 值的增大, 失真程度将越大。本文中 $Near$ 值设为 0。

2.2.2 常规模式编码

进入常规模式编码后, 将对每个像素值分别进行预测编码与 Golomb 编码后输出压缩结果。

(1) 梯度的量化与合并 首先应该对上下文梯度推导得到的局部梯度值 D_1 、 D_2 、 D_3 进行梯度量化。如果图像矩阵在 X 轴与 Y 轴方向没有任何倾向性, 那么可以将 D_1 、 D_2 、 D_3 量化为近似等概率区间, 也就是说局部梯度 D 的概率与局部梯度 $-D$ 的概率是一样的, 具体的梯度量化过程根据以下公式完成:

$$Q_i = \begin{cases} -4, & D_i \leq -T_3 \\ -3, & -T_3 < D_i \leq -T_2 \\ -2, & -T_2 < D_i \leq -T_1 \\ -1, & -T_1 < D_i \leq -Near \\ 0, & -Near < D_i \leq Near \\ 1, & Near < D_i \leq T_1 \\ 2, & T_1 < D_i \leq T_2 \\ 3, & T_2 < D_i \leq T_3 \\ 4, & D_i > T_3 \end{cases} \quad (2-6)$$

其中, i 的取值范围为{1,2,3}, $Near$ 为失真度控制参数, T_1 、 T_2 、 T_3 为非负的阈值参数。如此, 根据 D_1 、 D_2 、 D_3 与阈值 T_1 、 T_2 、 T_3 及 $Near$ 值的关系就可以计算得到上下文矢量 (Q_1, Q_2, Q_3) 的值。可以看到, 局部梯度经过量化之后, 最终都会处于{-4,-3,-2,-1,0,1,2,3,4}的 9 个整数的范围中。

在梯度合并之前, 需要先对矢量 (Q_1, Q_2, Q_3) 进行符号校正, 校正伪代码如图 2-5 所示。校正规则为: 假设矢量的第一个非零值是小于零的数, 则该矢量的符号

位取反；否则，该矢量符号保持不变。如果矢量符号取反，符号变量 $SIGN$ 的值将同时被置为-1，否则，符号变量 $SIGN$ 的值置为 1^[45]。

```

if      ( $Q_1 < 0$ ) || (( $Q_1 == 0$ ) & (( $Q_2 < 0$ ) || (( $Q_2 == 0$ ) & ( $Q_3 < 0$ ))))
    ( $Q_1, Q_2, Q_3$ ) = ( $-Q_1, -Q_2, -Q_3$ )
     $SIGN = -1$ 
else
    ( $Q_1, Q_2, Q_3$ ) = ( $Q_1, Q_2, Q_3$ )
     $SIGN = 1$ 
    
```

图 2-5 符号校正伪代码

经过量化并符号校正后的梯度需要进行合并形成整数 Q ，目的是避免随着上下文数目的增加而带来的数据存储需求的增加，从而实现 JPEG-LS 低复杂度的目标。JPEG-LS 标准要求矢量 (Q_1 、 Q_2 、 Q_3) 到整数 Q 的映射必须为一一映射关系，且 Q 的范围为[0, 364]。本文，采用映射关系式为：

$$Q = 9 \times (9 \times Q_1 + Q_2) + Q_3 \quad (2-7)$$

(2)预测 预测编码是 JPEG-LS 算法的重要组成部分，预测分为四大部分：边界检测、预测值校正、预测误差计算和误差范围约束。

一般情况下，预测编码首先应该根据待编码像素点 X 周围的信息数据情况建立不同的模型，然后根据具体的情况进行自适应预测。但是在实际操作过程中，为了达到 JPEG-LS 低复杂度的需求，采用一个固定的预测模型完成边缘检测功能，该预测模型如下：

$$Px = \begin{cases} \max(Ra, Rb), & Rc \geq \min(Ra, Rb) \\ \min(Ra, Rb), & Rc \geq \max(Ra, Rb) \\ Ra + Rb - Rc, & \text{else} \end{cases} \quad (2-8)$$

其中， Ra 、 Rb 、 Rc 分别为像素点 A、B、C 处像素值（重建值）， Px 为通过预测模型预测得到的预测值。此预测模型功能：如果待测像素点 X 位于像素矩阵的垂直边界处，则预测值 Px 为像素点 A 处的值；如果待测像素点 X 位于像素矩阵的水平边界处，则预测值 Px 为像素点 B 处的值；如果待测像素点 X 位于像素矩阵的平面中，则预测值 Px 为 $Ra+Rb-Rc$ 的值。

预测难免存在着预测偏差，边界预测结束需要对预测结果进行修正，补偿预测偏差，从而使得预测误差满足双边几何分布。该过程需要用到梯度合并过程中生成的符号变量 $SIGN$ 和参数 $C[Q]$ ，过程如下：

$$P_x = \begin{cases} P_x + C[Q], & SIGN == +1 \\ P_x - C[Q], & SIGN == -1 \end{cases} \quad (2-9)$$

并且需要保证校正后的值 P_x 位于 $[0, MAXVAL]$ 的区间内, 过程如下:

$$P_x = \begin{cases} MAXVAL, & P_x > MAXVAL \\ 0, & P_x < 0 \end{cases} \quad (2-10)$$

$MAXVAL$ 的大小为 $2^P - 1$, P 为像素精度, 本文 P 值为 8。

得到待测像素点 X 的预测值 P_x , 再利用该像素点的真实像素值 I_x 和符号变量 $SIGN$, 通过以下公式计算得到预测误差。

$$Errval = \begin{cases} I_x - P_x, & SIGN == +1 \\ P_x - I_x, & SIGN == -1 \end{cases} \quad (2-11)$$

在本文中, 只考虑 JPEG-LS 算法的无损压缩模式, 因此只需要完成预测误差的范围约束, 具体的约束公式如下:

$$Errval = \begin{cases} Errval + RANGE, & Errval < (-RANGE/2) \\ Errval - RANGE, & Errval \geq ((1 + RANGE)/2) \end{cases} \quad (2-12)$$

公式 (2-12) 中的 $RANGE$ 为动态范围, $RANGE = MAXVAL + 1$ 。

(3) 误差编码 **Golomb** 编码在针对服从几何分布的非负整数的编码方面有很大的优势, 属于熵编码的范畴, 它是预测误差编码的主要部分。该编码方式的核心: 假设非负整数 N 为待编码数据, 同时有给定的参数 M , 那么 **Golomb** 编码结果将由两部分构成: 第一部分是 N/M 的二元码(相应个数的零表示的正整数值的大小)表示的比特位; 第二部分是 $N \bmod M$ (N/M 的余数)表示比特位, 并且在第一部分和第二部分之间需要加入一个比特位“1”。在 JPEG-LS 标准中, 采用的是 **Golomb-Rice** 编码^[46], 它的好处在于规定参数 M 必须为 2 的 K 次幂, 如此做将很容易得到余数及对余数进行编码, 达到 JPEG-LS 低复杂度的需求。

在预测误差编码之前需要利用变量 $A[Q]$ 、 $N[Q]$ 的值计算得到 **Golomb-Rice** 编码所需参数 K , 根据以下公式求得:

$$for(k = 0; (N[Q] < k) < A[Q]; k++) \quad (2-13)$$

再利用 K 值及 $B[Q]$ 、 $N[Q]$ 的值将预测误差 $Errval$ 映射为 $MErrval$, 得到的结果将是一个非负整数值, 伪代码如图 2-6 所示。

```

if(NEAR == 0)&&(k == 0)&&(2*B(q) <= -N(q))
    if(Errval >= 0)
        MErrval = 2*Errval + 1;
    else
        MErrval = -2*(Errval + 1);
    end
else
    if(Errval >= 0)
        MErrval = 2*Errval;
    else
        MErrval = -2*Errval - 1;
    end
end

```

图 2-6 误差校正伪代码

预测误差编码规则分为两种情况，具体如下：

假若 $MErrval/2k < Limit-qbpp-1$ ，则 Golomb 编码的高比特位为 $MErrval/2k$ 的二元码，即为 $MErrval/2k$ 整数位个“0”；低比特位为 $MErrval$ 的 k 个比特位的二进制表示；将一个二进制位“1”插入上述两部分之间，最终构成 Golomb 编码结果。

假若 $MErrval/2k \geq Limit-qbpp-1$ ，则 Golomb 编码的高比特位为 $Limit-qbpp-1$ 的二元码，即为 $Limit-qbpp-1$ 个“0”；低比特位为 $MErrval-1$ 的 $qbpp$ 个比特位的二进制表示；将一个二进制位“1”插入上述两部分之间，最终构成 Golomb 编码结果。

2.2.3 游程模式编码

当前像素点的值进入游程模式编码，意味着该点周围像素矩阵的梯度值在变量 $Near$ 的范围内。无损压缩模式时， $Near$ 等于零，也就是当前像素点周围像素值相等。游程模式编码即是对检测到相等的这些像素点进行编码，加快编码进程，解决直接 Golomb 编码造成的冗余问题。游程模式编码主要分为两大部分：游程扫描及编码、游程中止采样编码。

(1) **游程扫描及编码** 游程扫描的目的是获取连续相等的像素点的数目，过程较为简单。如果上一次像素点周围域内的像素值与此次像素点周围域内的像素值相等，那么游程扫描将游程长度加 1，如此循环接收下一个像素点的值，直至不满足此条件或者扫描到行末，计算得到最终的游程长度值。

游程扫描结束得到游程长度值即可对此值进行编码，JPEG-LS 用一个大小为

32 的数组 $J[0,0,0,0,1,1,1,1,2,2,2,2,3,3,3,3,4,5,5,6,6,7,7,8,9,10,11,12,13,14,15]$ 来存储编码的长度等级。游程长度编码过程伪代码如图 2-7 所示。

```
while(RUNcnt >= (1 << J[RUNindex])){
    output = (output<<1)+1;
    RUNcnt -= (1 << J[RUNindex]);
    if(RUNindex < 31)
        RUNindex++
}
if(!EOLine){
    output=(output<<1);
    output=(output<<(J[RUNindex]))+RUNcnt;
    if(RUNindex > 0)
        RUNindex--
}else
    if(RUNcnt > 0)
        output=(output<<1)+1;
```

图 2-7 游程长度编码过程伪代码

(2) 游程中止采样编码 当游程扫描过程由于遇到不等像素点或者扫描到行末而结束时,将对新采样进行单独编码,该编码过程与之前的 Golomb 编码类似,只不过是一种变长编码方式,即编码过程用到的 $Limit$ 不是确定值。除此外,该过程与常规模式编码相似,不再赘述。

2.3 本章小结

在本章中,介绍了图像压缩理论的基本知识,说明了图像压缩的原理和方法,并对主流的压缩方法进行了简单的说明。着重研究了 JPEG-LS 图像压缩标准,给出了其实现的结构框图,并对 JPEG-LS 算法的主要过程:上下文建模、常规模式编码、游程模式编码等做了深入的研究,分析了其压缩流程,确定了编码器需求的重要参数的取值范围。

第3章 JPEG-LS 压缩编码器设计与实现

本章主要内容是采用 Verilog HDL 实现基于 JPEG-LS 算法的硬件设计，将整个编码器划分为五大部分：模式选择器负责选择编码模式的工作；常规和游程模式编码器属于核心编码器，负责完成常规与游程编码；参数更新模块完成编码器内所有参数的更新功能；输出码流控制器负责码流的拼接、整合与输出。为了配合该编码器的正常运行，将对图像进行预处理操作，并设计像素输入控制器将数据正确输送到编码器内。

3.1 图像预处理模块设计

本文设计的 JPEG-LS 图像压缩编码器的数据来源要求是二进制格式数据，因此需要对 bmp 格式的图像信息进行转化。MATLAB 在图像处理上具有操作简单、转化速度快的特点，故采用 MATLAB 软件对已有的自然图像进行处理。处理过程如图 3-1 所示。

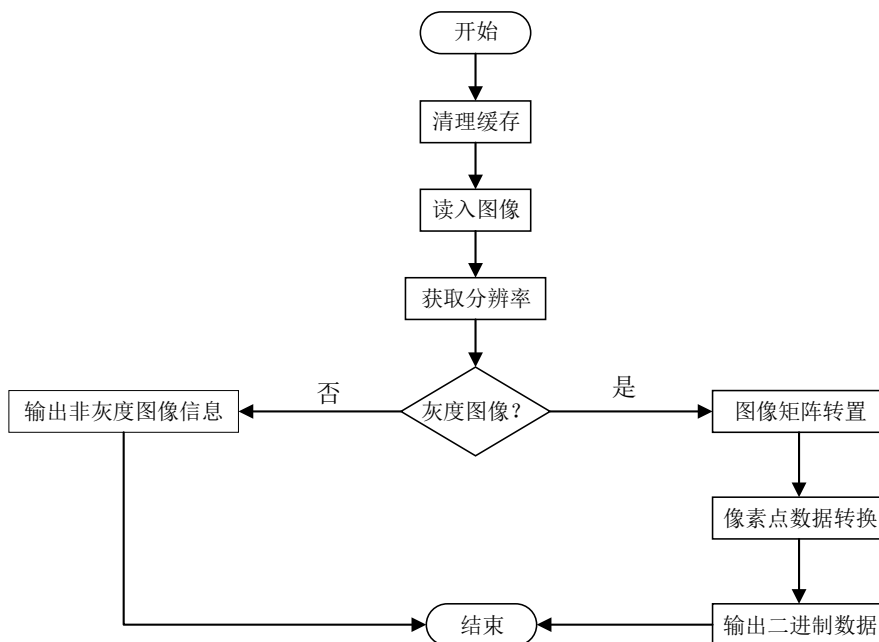


图 3-1 图像预处理流程图

首先，清理上一次图像处理过程结束后的缓存数据，避免缓存数据对本次处理过程造成影响。读入图像数据，先获取图像的分辨率，即横竖两个方向的像素点数目，判断是否为 512×512 的图像，再根据图像像素点的分布情况判断是否为

本文需要的灰度图像。在处理对象为 512×512 的灰度图像的前提下，进行图像数据矩阵转置和像素点数据转换等操作，最终输出二进制格式的图像数据，输出数据名将以原图像名加分辨率的格式命名。如果不是灰度图像，则输出该图像信息后退出程序。

图像信息处理前后的数据对比如图 3-2 所示，(a)为未经过处理的图像信息，(b)为经过处理转换后的部分二进制数据信息，此数据才可输入到图像压缩编码器中。利用 MATLAB 本身具有的像素值分析能力，可得出原来的十进制像素值。经过对比，转换后得到的二进制像素值与原十进制像素值是相同的。

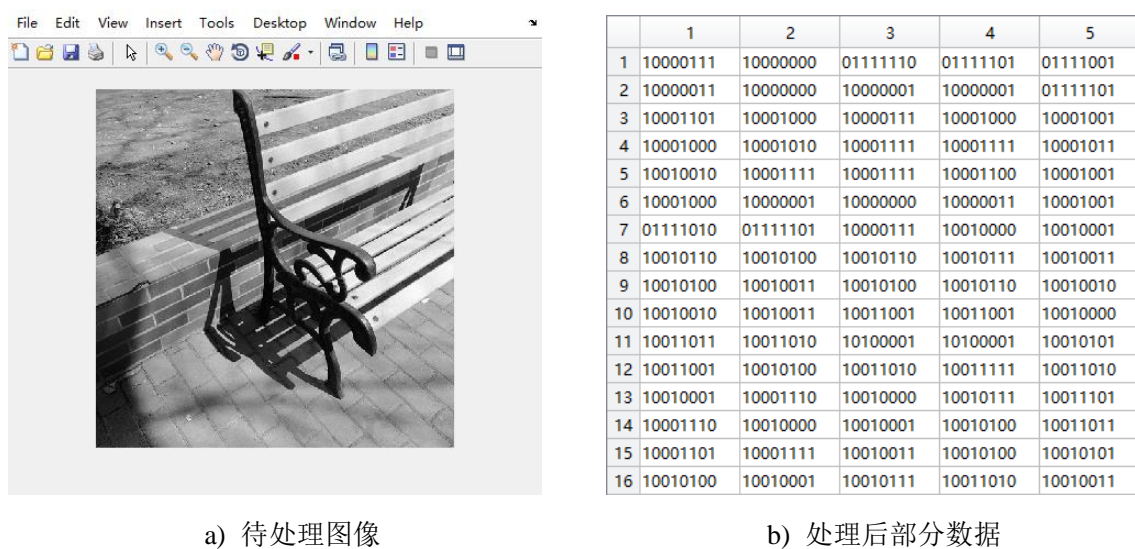


图 3-2 待处理图像与处理后部分数据

3.2 像素输入控制器模块设计

JPEG-LS 算法与传统的 JPEG2000 等图像压缩算法不同点在于：该算法按照栅扫描的方式读入数据，即读取图像像素值时，先读取像素矩阵左上角的像素值，向右扫一条水平线依次读取水平线上的像素值，每当读取到行末时，再回扫到下一行的最左边处像素点，继续向右扫这一行水平线，按照此固定顺序不停地扫下去，当读取完成最后一条水平线上的全部像素值时结束。而且，每次读取像素点 X 处的像素值时，需要同时读入该点周围 4 处像素点 A、B、C、D 的像素值，如图 3-3 所示，即一次性需要读取 5 个 8bit 的像素值。此外，输入像素值进入 JPEG-LS 编码器的时钟周期不确定，需要编码器发送请求信号才能发送数据。因此，JPEG-LS 算法这种获取像素值的方式要求必须设计与之相匹配的像素输入控制器，从而保证像素值输入的高效性与准确性。

已读 像素值	已读 像素值	已读 像素值	已读 像素值	已读 像素值
已读 像素值	C像素值	B像素值	D像素值	待读 像素值
已读 像素值	A像素值	X像素值	待读 像素值	待读 像素值
待读 像素值	待读 像素值	待读 像素值	待读 像素值	待读 像素值

图 3-3 像素值输入示意图

考虑到需要同时输入 5 个 8bit 的像素值，对于 512×512 的像素矩阵，向编码器内输入第一行、第一列的像素值 X 时，X 周围的像素值 A、C、B、D 并不存在，所以需要像素输入控制器补足像素矩阵第一行、第一列、最后一列的像素值，在本文设计中，将其补成像素值 0，由此像素矩阵变为 $513 \text{ 行} \times 514 \text{ 列}$ ，如图 3-4 所示。

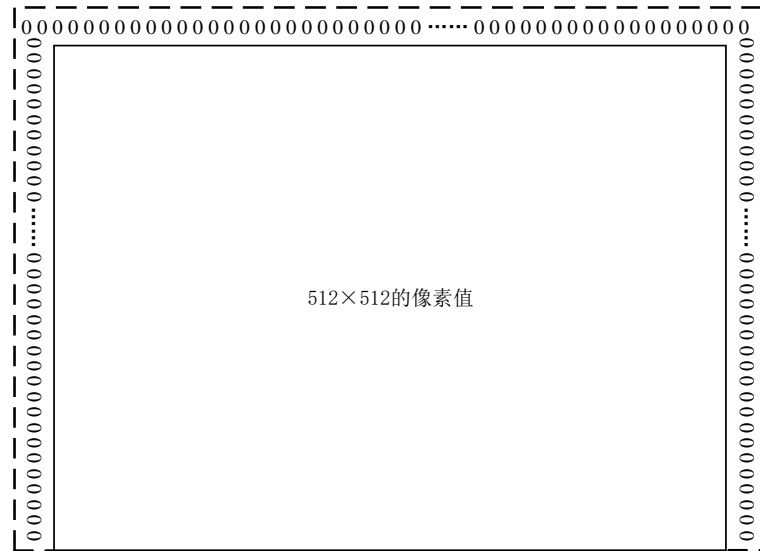


图 3-4 513×514 的像素矩阵示意图

本文中，像素输入控制器开始工作时，每次接收到编码器的像素请求信号，便从存储器中读取像素矩阵的一个像素值并存储在 2×512 的像素缓存器的第一行的相应位置中，直至读到像素矩阵的行末。下一次读取像素矩阵的第二行存储到 2×512 的像素缓存器的第二行的相应位置的同时，控制器读取缓存器的第一行像素数据输入给图像压缩编码器。由此顺序，缓存器的第一行存储位将得到奇数行的像素矩阵值，缓存器的第二行存储位将得到偶数行的像素矩阵值，并且完成同时

输入 5 个像素值的任务，如图 3-5 所示。



图 3-5 像素矩阵存储器与缓存器关系图

图 3-6 为像素输入控制器的仿真结果，可以看到，该控制器可以正确地将 5 个像素值输出。

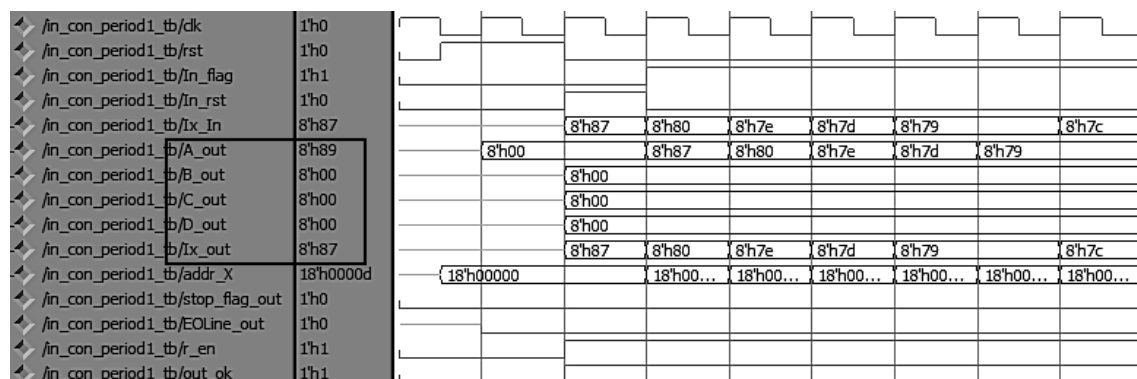


图 3-6 输入控制器仿真结果

3.3 基于 JPEG-LS 的无损压缩编码器设计

基于 JPEG-LS 算法的压缩编码器设计可分为五大部分：常规模式编码器设计、游程模式编码器设计、参数更新模块设计、模式选择器设计和输出码流控制器设计。采用层次化编程的方式，将整个系统的结构层次划分为 4 层，如图 3-7 所示。

第一层为顶层模块 top，包含所有模块的组织 and 调用，以及对所有的 I/O 信号的定义；第二层包含参数更新模块、常规模式编码模块、游程模式编码模块、模式选择模块和输出码流控制器模块，其中参数更新模块负责控制整个系统的参数更新操作，常规模式编码模块将实现 JPEG-LS 算法常规编码部分的功能，游程模

式编码模块将实现该算法游程编码部分的功能，模式选择模块将判断被输入的图片二进制码流是否位于平坦区域，从而控制像素值输入常规模式模块或游程模式模块，输出码流控制器模块将获得上述两个编码模块的结果，从而整合成压缩码流输出到已有的存储器中；第三层结构是两个编码模块的子模块，各个子模块间的数据通信与使能控制皆是经过上一级模块完成；第四层是哥伦布编码模块的子模块。

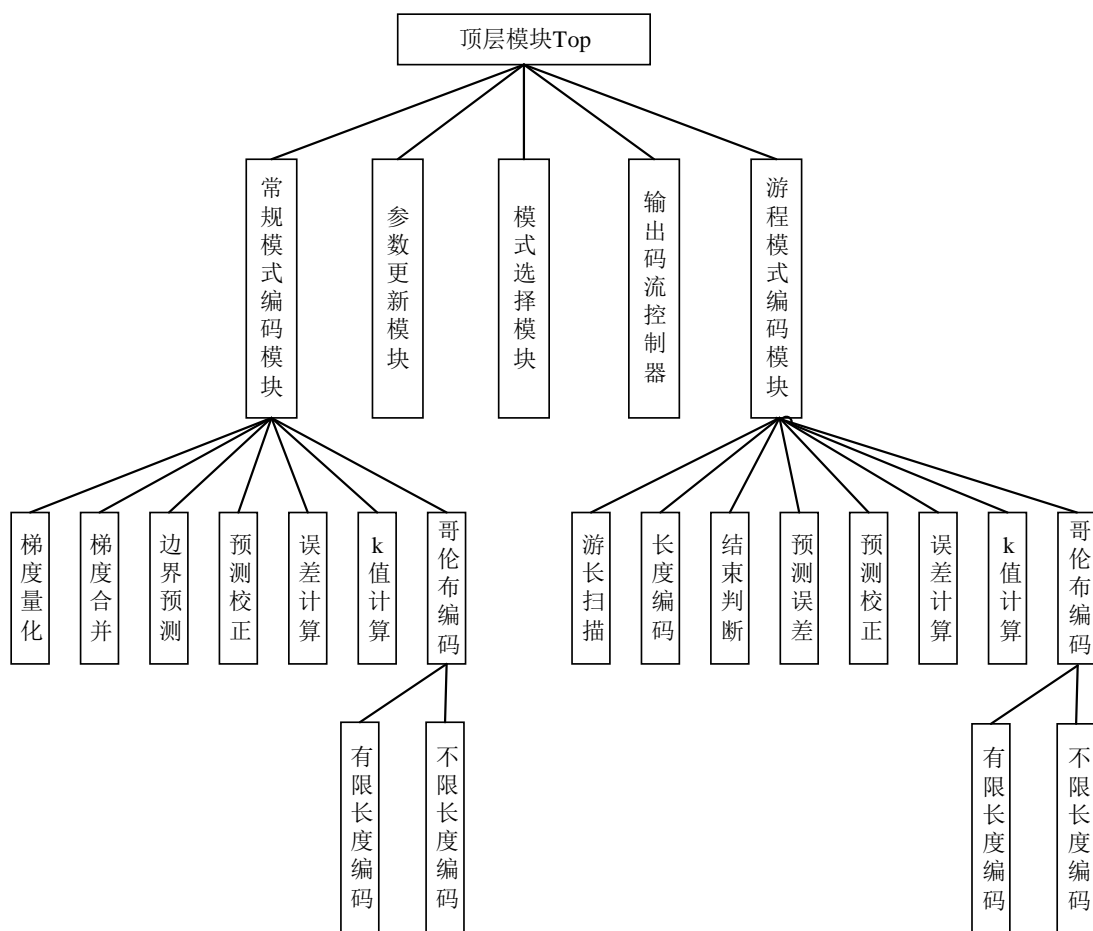


图 3-7 层次化编程示意图

JPEG-LS 无损压缩编码器硬件设计整体结构如图 3-8 所示。在像素输入控制器的调控下，待压缩的图像二进制数据进入到模式选择器内；模式选择器将对其进行判断，符合游程模式编码器的条件，则将其输送到游程模式编码器内，否则输入到常规模式编码器中；常规模式编码器和游程模式编码属于核心编码器，所有的像素数据都必须经过它们才能输出编码结果，编码过程中涉及到参数的使用与更新，由参数更新模块完成；最终，输出码流控制器接收核心编码器的编码结果，并将其拼接成完整的二进制码流后输出。

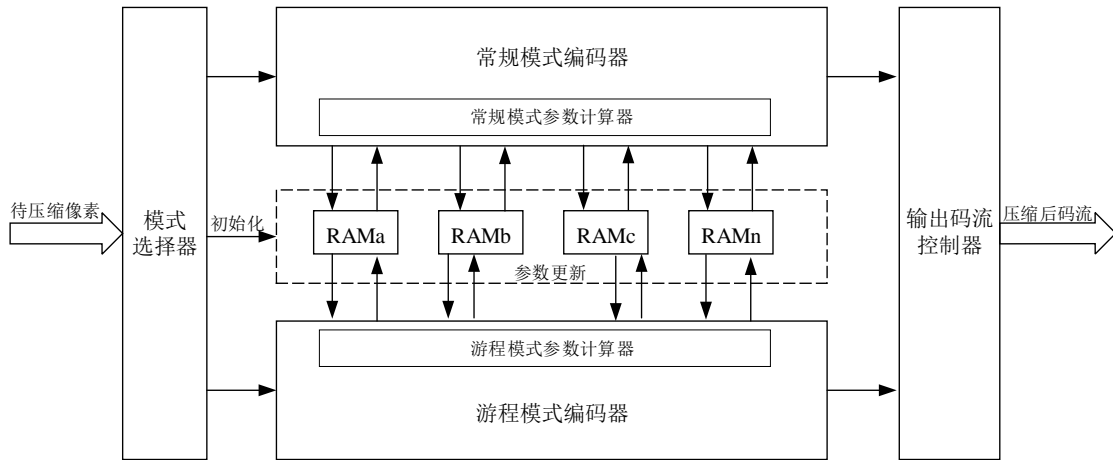


图 3-8 JPEG-LS 编码器整体架构

3.3.1 模式选择器设计

在 JPEG-LS 标准中，模式选择阶段负责完成判断当前像素点局部区域的像素值的是否具有相关性，即当前采样区域周围值是否相等。因此，根据此需求，模式选择器的设计分为两步，首先进行梯度推导，然后根据梯度推导的结果决定编码器选择哪一种工作模式。

梯度推导过程如图 3-9 所示，将输入控制器输入得到的像素值 R_a 、 R_b 、 R_c 和 R_d 暂存到寄存器中，再按照梯度推导公式进行模减运算，通过比较器得到梯度值是否完全相等，从而为后面的模块提供使能信号。

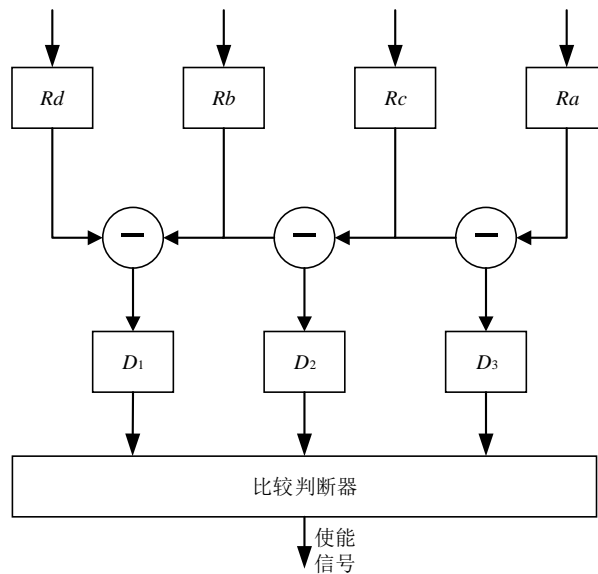


图 3-9 梯度推导运算示意图

在梯度推导执行完成后，模式选择模块开始工作。在本文中，此部分设计采用 5 个状态的状态机完成，如图 3-10 所示。

req_flag 为下一个模块的请求信号，当前一个像素值完成压缩过程时，此信号有效为 1，否则为 0；IDLE 为初始状态，等待图像压缩的开始信号 $start$ ，当 $start$ 有效时，状态机从 IDLE 转入 S_0 态，否则一直等待； S_0 态为初次进入模式选择阶段的一个状态，在这个状态下，满足不同的条件状态机跳转到不同的状态中去：当像素值 Ra 、 Rb 、 Rc 、 Rd 不完全相等且 req_flag 有效时，进入常规模式编码 S_1 态，当 Ra 、 Rb 、 Rc 、 Rd 完全相等、 Ra 等于 Ix 且 req_flag 有效时，进入游程模式编码 S_2 态，当 Ra 、 Rb 、 Rc 、 Rd 完全相等、 Ra 不等于 Ix 且 req_flag 有效时，进入游程模式编码 S_3 态； S_1 态、 S_2 态和 S_3 态之间转换的条件如图 3-10 所示，特别注意的是，为了满足最终两个模式编码结果不冲突，在游程与常规模式的切换过程中，一定要等待当前模式空闲状态才能进行切换。

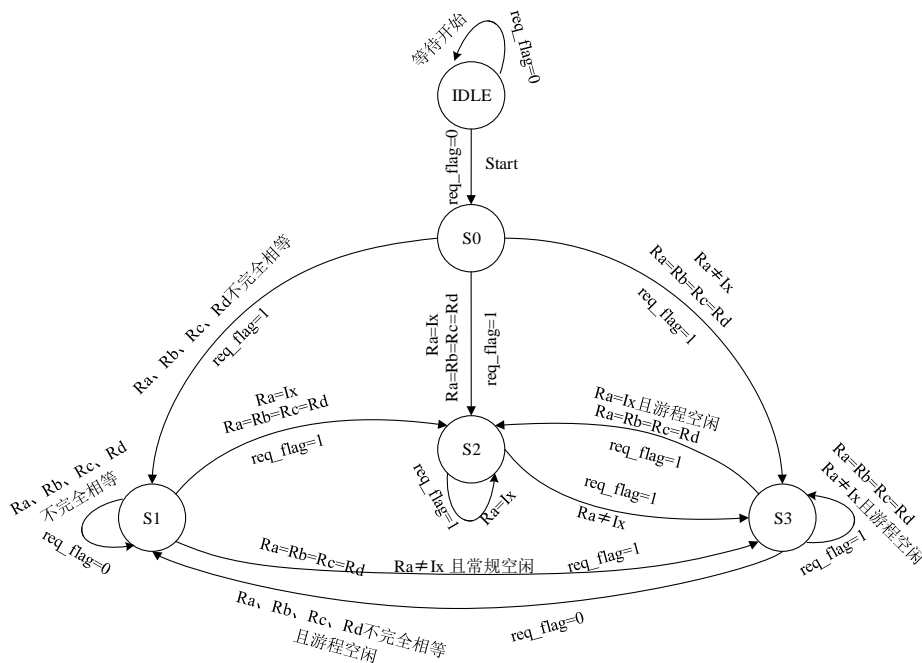


图 3-10 模式选择状态机

3.3.2 常规模式编码器设计

常规模式编码器的设计中包含有梯度计算、边界预测、预测校正、误差计算、 K 值计算和 Golomb 编码等模块，下面将分开阐述各个子模块实现方式。

在梯度计算之前，图像像素点值进入模式选择阶段得到了相应的像素点值的

差 D_1 、 D_2 和 D_3 ，如果 D_1 、 D_2 和 D_3 的值不完全相等，则该差值将被输入梯度计算模块。

梯度计算包含两个部分：局部梯度量化和量化后的梯度合并。在常规编码模式中，局部梯度量化是建立上下文模型的重要一步。梯度量化即对差值 D_1 、 D_2 和 D_3 进行量化，梯度量化的原理见式 (2-6)。根据 D_1 、 D_2 、 D_3 与阈值 T_1 、 T_2 、 T_3 的关系，得到一个区域号 Q_1 、 Q_2 和 Q_3 ，此为量化结果。

本文涉及的图像为 8bit 的灰度图像，即每个像素点的由 8bit 的二进制数据表示，但是其经过梯度推导之后，数值有可能出现负数。因此，需要第 9 位即最高位表示符号位。在量化过程中差值 D_1 、 D_2 和 D_3 与阈值 T_1 、 T_2 和 T_3 的比较涉及负数的比较。所以在此模块设计中采用先比较符号位再比较数值位的比较方式，得到最终的量化结果，如图 3-11 所示。

当梯度推导差值 D_1 、 D_2 和 D_3 进入梯度量化模块后，该模块开始工作，首先通过符号检测器检测梯度差值的符号，当符号标志位相同时，启动数值比较器比较数值大小，否则直接得出差值与阈值的大小关系。数值比较器采用移位寄存器的方式实现，这样可以节省寄存器硬件资源。工作方式如下：先比较最高位（第 7bit），如果数值相同，则启动移位寄存器左移一位，再比较最高位，直至比较出两位不相等，从而可以得出梯度推导差值与阈值的大小关系。由于存在三个梯度推导差值和三个阈值，因此整个模块需要三套这样的结构即可。

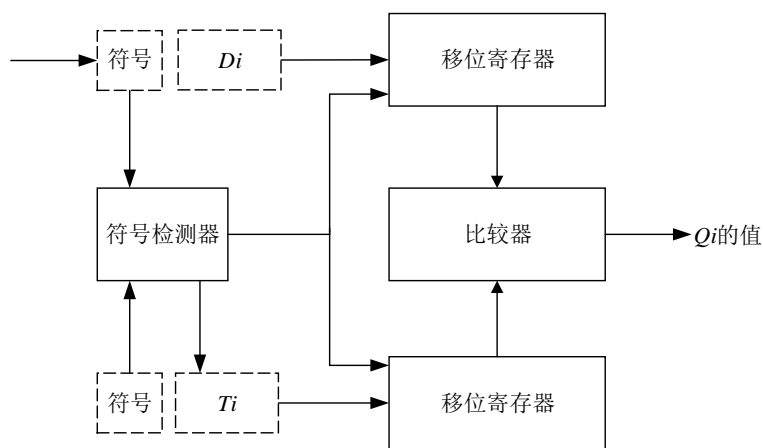


图 3-11 比较器实现结构图

梯度计算的另外一部分是量化后的梯度合并。在此部分中，首先完成的是对上述过程得到的每个量化梯度 Q_i 的符号进行修正，目的是保证合并过程中用到的 Q_i 是正整数，校正规则如 2.2.2 节所述。校正后的量化梯度矢量 (Q_1, Q_2, Q_3) 被一一映射成一个整数值 Q ，即表示当前像素点处的上下文。本文选用的映射函数如

下所示：

$$Q = 9 \times (9 \times Q_1 + Q_2) + Q_3 \quad (3-1)$$

观察公式（3-1）可知，在此表达式中涉及两次乘 9 运算，两次加法运算。对于硬件实现来说，直接乘 9 运算将耗费巨大的硬件资源，且需要更多的时钟周期，进而影响整个电路运行的性能与速度。因此，在本文设计中，将式（3-1）进行改造转化如下：

$$\begin{aligned} Q_{Temp} &= 8 \times Q_1 + Q_1 + Q_2 \\ Q &= 8 \times Q_{Temp} + Q_{Temp} + Q_3 \end{aligned} \quad (3-2)$$

这样，表达式中原本的乘 9 运算转换成乘 8 运算和加法运算的组合。对于硬件实现，乘 8 运算只需要将被乘数左移 3 位即可实现，而且移位运算仅仅需要一个时钟周期即可完成。采用此方式，将大大缩小硬件资源耗费的同时还有有效的提高了硬件执行速率。具体实现方式如图 3-12 所示。

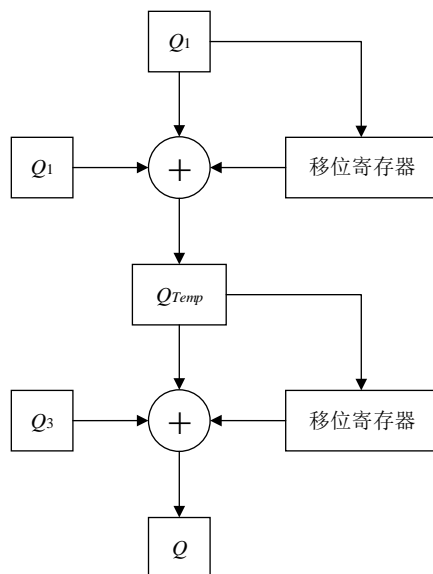


图 3-12 映射函数实现结构

边界预测是 JPEG-LS 算法中的重要一步，边界预测的目的是根据像素点 X 周围的像素值估计当前像素点的像素值 P_x 。预测处理的过程如 2.2.2 节所述，在本文设计中，其实现方式如图 3-13 所示。

像素点 A、像素点 B 的值通过比较逻辑得到 R_a 、 R_b 中的最大值与最小值；再通过比较逻辑得到像素点 C 处的值与 $\text{Max}(R_a, R_b)$ 及 $\text{Min}(R_a, R_b)$ 的关系，最后通过

选择器选择最终的结果得到 P_x 值。边界预测得到当前像素点的预测像素值 P_x 后，校正预测值的过程除了需要 P_x 外，还需利用四个值：偏差 $C[Q]$ 的值、变量 $SIGN$ 、零和常量 $MAXVAL$ 。在本文的设计实现中，采用如图 3-13 所示的结构实现预测校正的功能。 P_x 与 $C[Q]$ 的进行加法运算及减法运算，通过 $SIGN$ 的值筛选得到经过处理的 P_{x_t} ， P_{x_t} 的值有可能为负值甚至会大于 $MAXVAL$ ，因此通过比较器将 P_{x_t} 控制在 $[0 \dots MAXVAL]$ 范围内，最终得到校正后的 P_x 值。

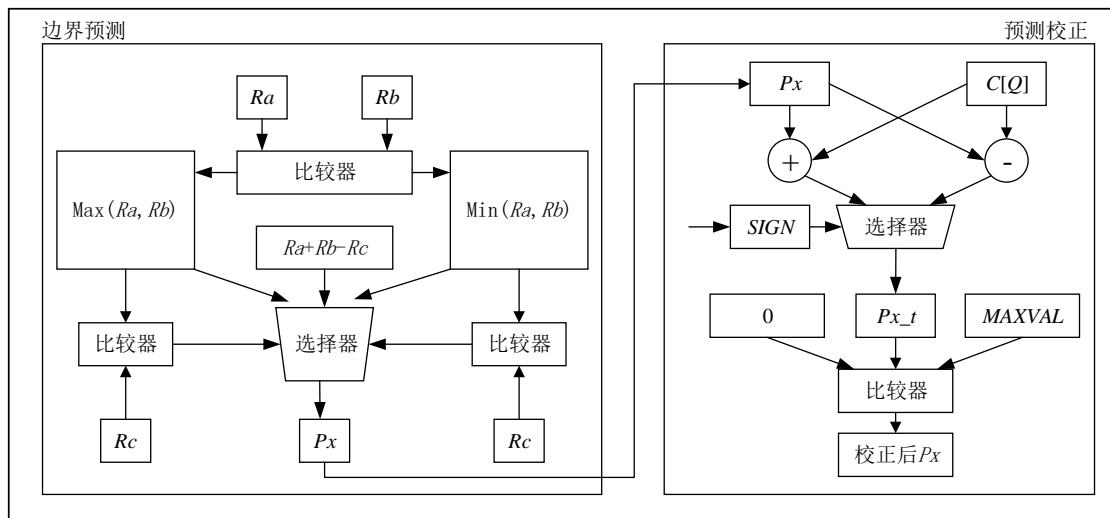


图 3-13 边界预测与预测校正结构图

K 值的获取依赖于 $A[Q]$ 和 $N[Q]$ 值的大小关系。硬件实现结构如图 3-14 所示，

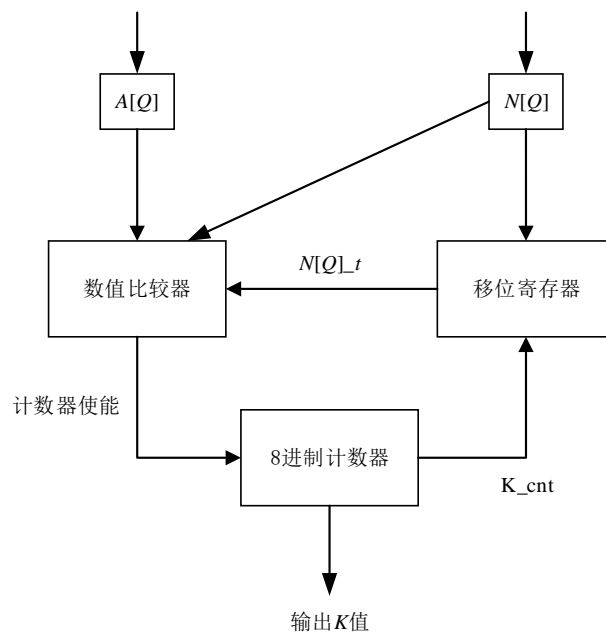


图 3-14 K 值计算结构图

$A[Q]$ 和 $N[Q]$ 的值通过对 RAM 存储器读取来获取,将得到的 $A[Q]$ 值和 $N[Q]$ 值进行比较,满足 $N[Q]$ 值大于等于 $A[Q]$,输出 K 值为 0; 否则,将 $N[Q]$ 值送入移位寄存器, $N[Q]$ 的值左移 K_cnt 位(计数器输出,初始值为 0),并且 8 进制计数器使能端有效,计数器开始计数。再次将移位后的 $N[Q]$ 值送入比较器与 $A[Q]$ 值进行比较,直至 $N[Q]$ 值大于等于 $A[Q]$,此时计数器输出的值 K_cnt 即为所求 K 值。由于 $A[Q]$ 值与 $N[Q]$ 值都是正数,所以不存在负数比较的问题,不必考虑符号位。

误差计算模块包含预测误差和误差修正两个部分。预测误差阶段利用校正后的预测值 P_x 和当前像素点的真实值 I_x , 计算预测误差 $Errval$ 。同时,该阶段将根据梯度计算模块得到的符号变量 $SIGN$, 计算预测误差的 $Errval$ 的符号。对得到的 $Errval$ 进一步处理,从而降低对预测误差编码的码字长度。根据 JPEG-LS 标准中的映射函数,误差校正阶段将对得到的 $Errval$ 进行映射,目的是将 $Errval$ 映射得到非负整数。简单来讲, $Errval$ 经过各类逻辑运算之后,通过零值、 K 值、 $B[Q]$ 与 $N[Q]$ 的关系结果对选择器进行使能,选择出相应的 $MErrval$ 值。整个模块的硬件实现结构如图 3-15 所示。

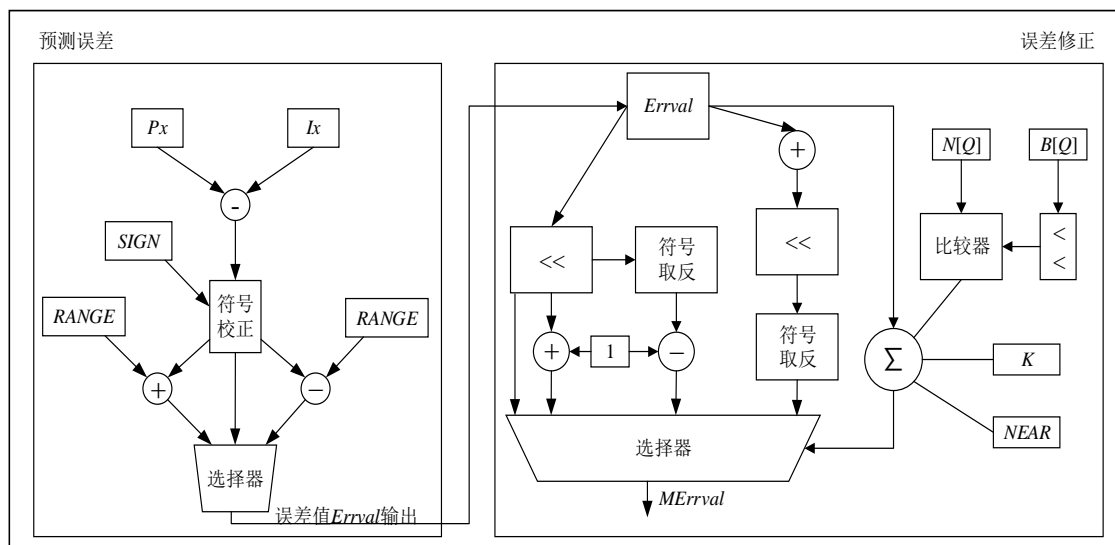


图 3-15 误差计算结构图

Golomb 编码模块是常规编码器中较为复杂的一个子模块,而且也是最终决定编码是否正确的决定性模块。该模块分为限长编码形式和非限长编码形式,在本文设计中将对其一同进行设计。根据 JPEG-LS 标准规则设计 Golomb 编码器结构如图 3-16 所示。

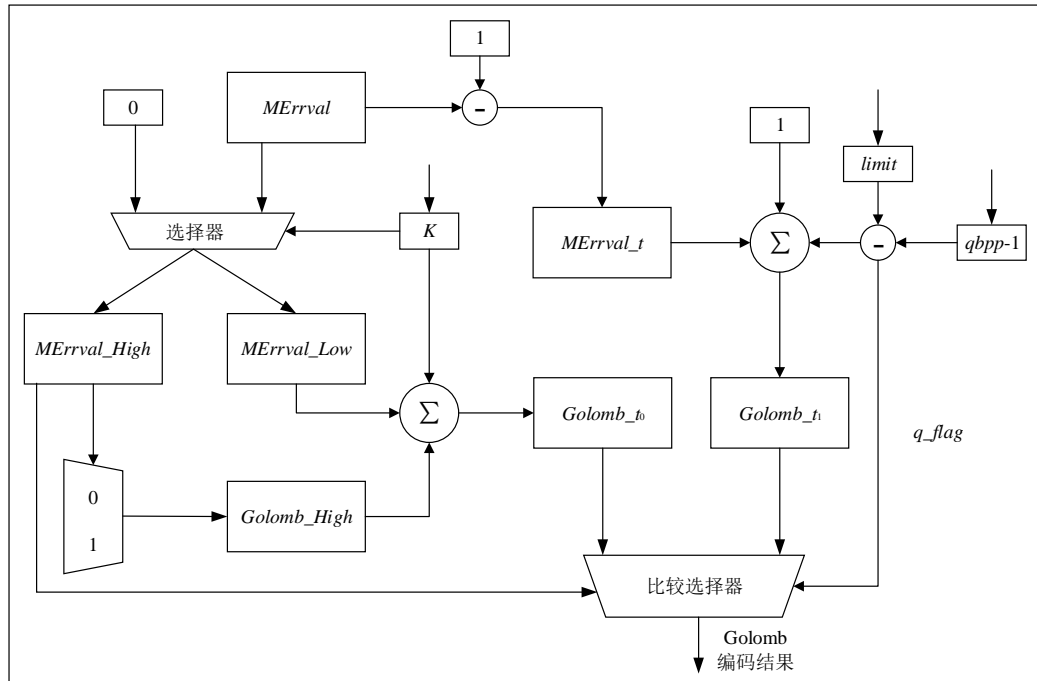


图 3-16 Golomb 编码结构图

Golomb 编码器接收到上一模块计算得到的 K 值与非负整数 $MErrval$ ，通过 K 值的大小对 $MErrval$ 进行分割，分出 k 个低比特位 $MErrval_Low$ ，剩下的值作为高比特位 $MErrval_High$ 。编码器根据 $MErrval_High$ 的值为编码结果的高比特位 $Golomb_High$ 分配相应的比特位的 0 和 1 值， $Golomb_High$ 与 $MErrval_Low$ 通过组合逻辑运算得到临时结果 $Golomb_t0$ ；同时，编码器根据输入的 $MErrval$ 、 $limit$ 和 $qbpp$ 的值计算得到临时结果 $Golomb_t1$ 值；最终通过选择比较器比较 $MErrval_High$ 与参数 q_flag 的值判断输出为限长编码形式还是非限长编码形式，从而输出 Golomb 编码结果。

综上所述，完成了常规模式编码器中各个子模块的设计，单独对每个模块进行了仿真验证，证明各个子模块功能正确。分析子模块的内变量的逻辑关系，得到常规模式编码器执行过程的流程图，如图 3-17 所示。

显然，在该编码器执行具有顺序执行的特点，故分析了每个子模块所需的时钟周期，并将可以并行执行的模块进行并行处理，尽可能地缩小压缩一帧图像所需的时钟数目，得到如图 3-18 所示的流水线结构。其中，Q 为梯度量化模块，M 为梯度合并模块，P 为边界预测模块，K 为 K 值计算模块，Pc 为预测校正模块，E 为预测误差模块，Me 为误差校正，G 为 Golomb 编码，up 为变量参数更新模块。

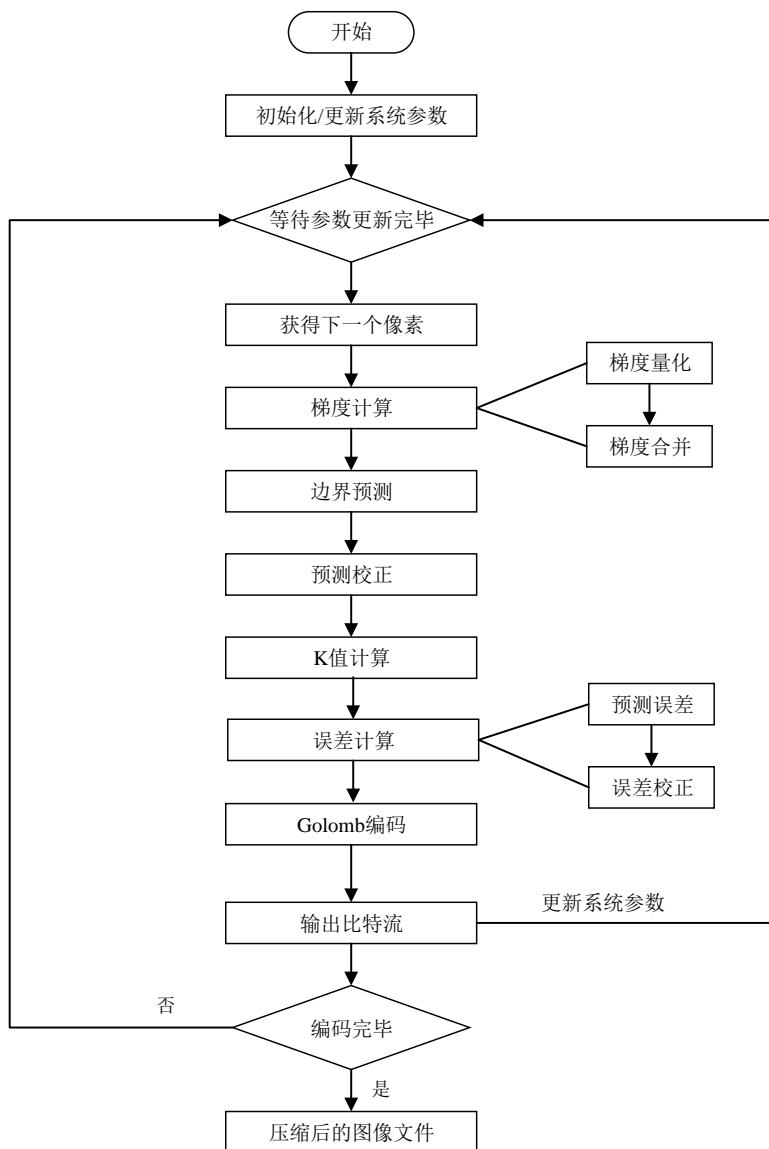


图 3-17 常规模式子模块执行流程图

在 JPEG-LS 算法中存在一种反馈逻辑环路，即上下文参数的更新环路，这使得前后像素具有较强的依赖关系^[47]。这种反馈逻辑环路为流水线结构实现带来了麻烦，因为反馈环路要求只有得到上个像素点参数变量更新后的才能进行下一个像素点值的压缩，而且下一个像素点值压缩未完成的情况下，不能再进行新的像素点值的压缩。为了解决这一矛盾，在本文的流水线设计中加了一级寄存器，用于存储上一像素点参数变量值，并且该寄存器的存取与预测误差模块并行执行，这样可以保证参数正确更新的同时，提高了整个系统的速率。利用 ModelSim 单独进行了仿真，仿真结果符合预期要求。

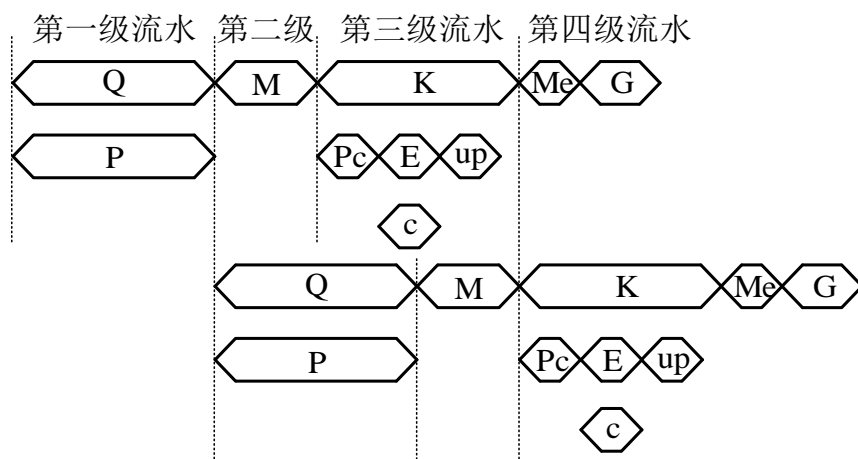


图 3-18 常规模式编码器流水线结构

3.3.3 游程模式编码器设计

在一些情况下，图像在局部区域内的像素点具有高度的相关性，也就是上文所叙述的平坦区域，即相邻几个像素点的值相等。这样的图像区域内，进行游程模式编码可以极大提高压缩速率。在 JPEG-LS 无损压缩标准中，如果采样位置 A、B、C 和 D 点的像素值完全相同，编码过程就通过模式选择模块进入游程模式。本文在游程模式编码器硬件实现上分为四大模块：游程扫描模块、游程长度编码模块、编码参数生成模块和采样编码模块，其中采样编码模块包含若干子模块，后面将详细叙述。

图像像素值进入游程模式编码器中，首先经过游程扫描模块，以当前像素点 X 位置为起点，向像素矩阵的 x 方向进行扫描，检验与其具有相同像素值的连续采样序列，直至像素值与之不相等位置或者扫描到像素矩阵的行末，则退出扫描并输出扫描长度，并且输出退出扫描的类型（像素不等退出或扫描到行末退出），如图 3-19 中游程扫描部分所示。首先将像素点 A 处的像素值 Ra 存入寄存器中，然后接收下一个像素点的像素值 Tx ，并与 Ra 进行比较，如果相等则计数器加 1，并且继续接收下一个像素点的像素值；如果不相等，则直接输出计数器的值，此值即为所求的游程长度，并输出游程扫描结束类型 end_type ；在上述过程中，如果接收到行末标志位 $EOLine$ 有效，则直接输出计数器的值和扫描结束类型。

在上一个模块中计算出了游程长度 $length$ ，游程长度编码模块的功能是对其进行编码。编码过程涉及对数组 J 的读取，根据读取的值调整参数 2^k ，数组 J 内数值如 2.2.3 节所述。设计游程长度编码结构如图 3-19 中游程长度编码部分所示。首次进入游程长度编码模块时，使用数组 J 的第一个数字，满足相应的逻辑运算后，

则将一个二进制位“1”插入到编码比特流中；后续每次得到一个游程长度时，指向数组 J 的地址变量加 1，持续增加到最大值 31。如果游程扫描是由于扫描行的末尾结束，并且该游程长度经过逻辑运算后剩余长度仍然大于零，则另外加入一个“1”比特到比特流中。如果游程扫描由于像素值不同而结束，则对剩余游程长度进行编码，码字长度为 $k+1$ 位，并且在其前面加入一个二进制位“0”。同时，数组 J 的地址变量减去 1，但不能小于 0。

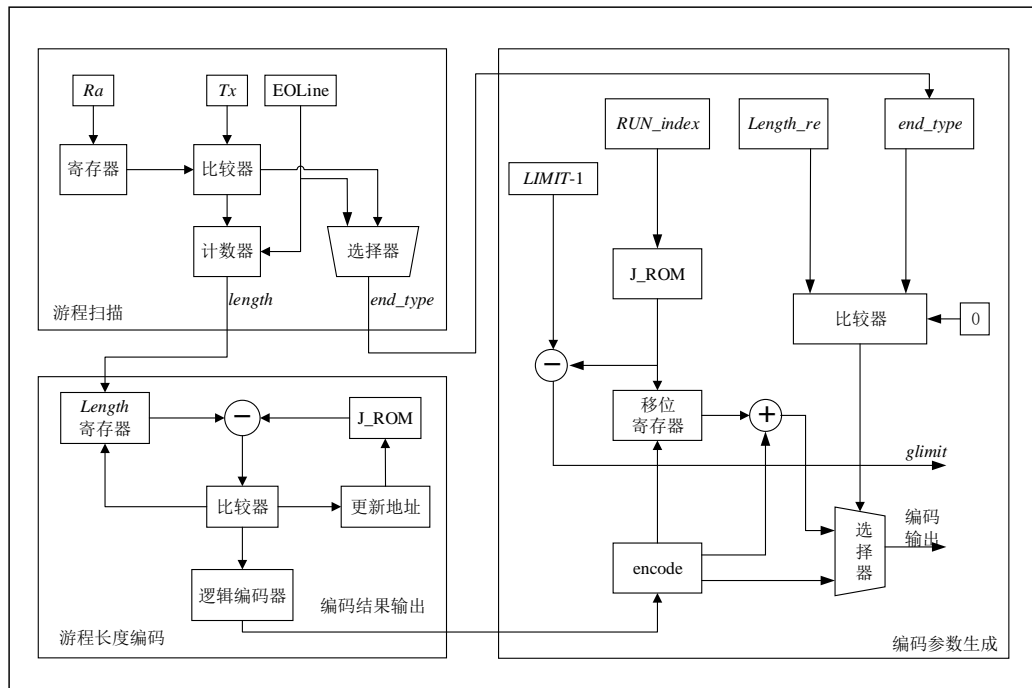


图 3-19 游程扫描及编码结构图

在游程长度编码模块中，数组 J 地址变量 RUN_index 得到了不断更新， RUN_index 的值同样与 Golomb 编码模块的参数 $glimit$ 相关，参数生成模块将根据 RUN_index 值生成 $glimit$ ；同时，如果因为扫描到像素矩阵行末而结束的类型，本模块还负责输出编码结果，此时的编码结果即为整个游程模式编码器的编码结果。设计本模块硬件实现结构如图 3-19 中编码参数生成部分所示，从上个模块获得 RUN_index 值，根据此值到 J_ROM 存储器中读取相应的数据，将此数据与 $LIMIT-1$ 进行模减运算即可得到参数 $glimit$ ；获得上个模块的结果剩余游程长度 $Length_re$ 及扫描结束类型 end_type ，通过逻辑比较器的输出结果控制编码输出结果。

对于由于像素值不等而结束游程扫描模块的情况，中断采样编码模块将对其进行编码。在编码过程中，将完成预测误差计算、预测误差校正、 K 值计算、Golomb 编码和参数更新。以上各个过程与常规模式的功能作用相同，在硬件实现上仅对

已完成的模块做部分修改就可用于游程模式中。中断采样编码模块结构如图 3-20 所示。

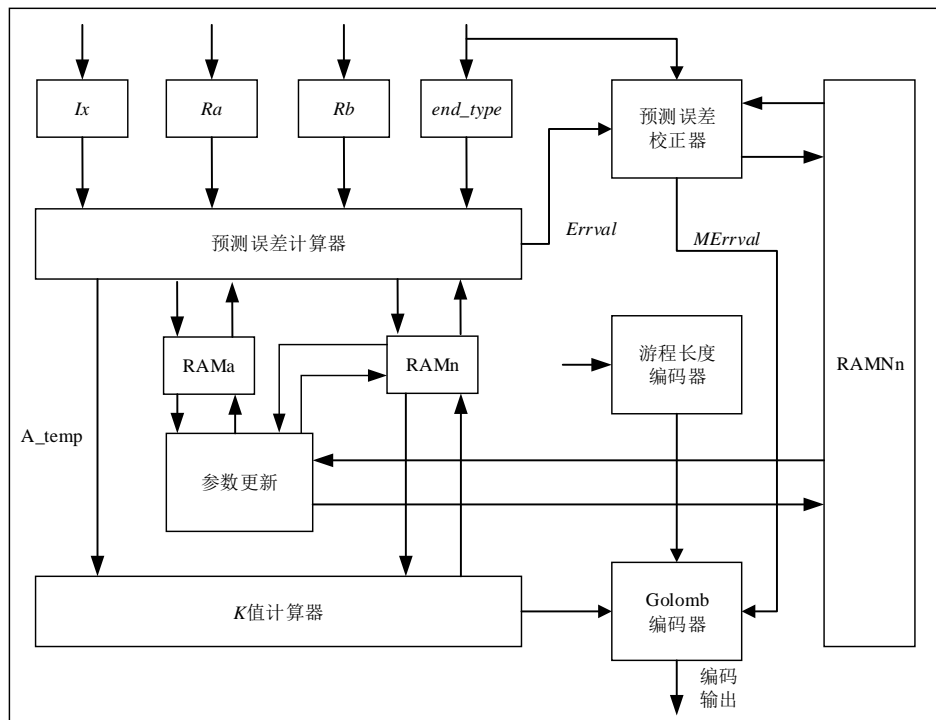


图 3-20 中断采样编码模块框架图

将游程扫描结束类型信号 end_type 连同像素点 A、B、X 的像素值输入预测误差计算器中计算得到预测误差 $Errval$ 值，并且通过逻辑运算得到 $A[Q]$ 的映射值 A_temp ；在 end_type 值的选取下， $Errval$ 通过预测误差校正器得到校正后的值 $MErrval$ ； K 值计算器从预测误差计算器中获取到 $A[Q]$ 的映射值 A_temp ，从存储器 $RAMn$ 中得到 $N[Q]$ 值，从而计算出 K 值；Golomb 编码器接收以上的信号值，利用编码规则输出编码结果；参数更新模块最终完成对参数 $A[Q]$ 、 $N[Q]$ 和 $Nn[Q]$ 的更新任务。

由此，设计完成整个游程模式编码器，由于游长扫描模块的存在，游长长度的不确定性，导致游长扫描模块、游长长度编码模块时钟周期不确定，因此无法采用流水线的处理方式提高编码速率。

3.3.4 参数更新模块设计

变量参数更新模块的目的是对常规模式和游程模式用到的变量 $A[Q]$ ， $B[Q]$ ， $C[Q]$ ， $N[Q]$ 进行更新，更新逻辑按照 JPEG-LS 标准进行设计。在硬件实现上采用

单端口 RAM 读写实现, RAMa 宽度为 13, 深度为 366; RAMb 宽度为 9, 深度为 366; RAMc 宽度为 9, 深度为 366; RAMn 宽度为 8, 深度为 366。地址位即为上述梯度合并模块计算得到的 Q 值, 通过 Q 寻址对各个 RAM 进行读写操作。其中, 地址[0:364]为常规模式所使用, 地址[365:366]为游程模式所使用, 如图 3-21 所示。

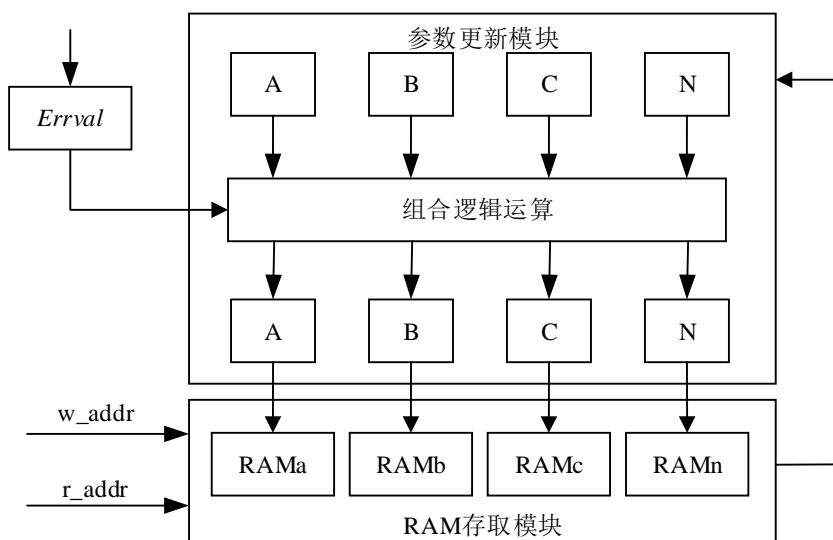


图 3-21 参数更新模块结构

3.3.5 输出码流控制器设计

由于 Golomb 编码输出的编码结果位数不确定, 输出位数在 1-32 位之间, 且常规模式输出码流与游程模式输出码流需要进行有顺序的拼接。因此, 需要单独设计输出码流控制器以便高效、正确的输出压缩码流。

两种编码模式输出结果不确定性给硬件实现带来了一定的困难, 在本文中, 设计了如图 3-22 所示的输出控制器实现码流输出。

常规模式编码器与游程模式编码器将分别输出码流与码流对应的比特数目到输出码流控制器中。两种编码器输出的码流将分时地进入该模块的码流拼接器中, 码流拼接器负责将本次接收到的码流与之前的码流进行拼接。寄存器地址生成器接收到比特数目, 继而根据此数据计算生成拼接码流所需的 64 位移位寄存器的首地址, 一方面将该首地址输送给 64 位移位寄存器, 控制其移位; 另一方面, 将该地址输送给输出检测器。输出检测器判断首地址是否超过 31, 如果地址位超过 31, 则输出控制器使能端有效, 否则, 使能端无效。最终, 输出控制器从 64 位移位寄存器中截取[0:31]位数据输出。由于该模块接收数据的时间具有不确定性, 因此设计输入信号 In_flag 作为控制信号, 确保码流数据与比特数据同时被接收。利用简

答移位寄存器与拼接器即实现了码流控制输出功能，节省了硬件资源。

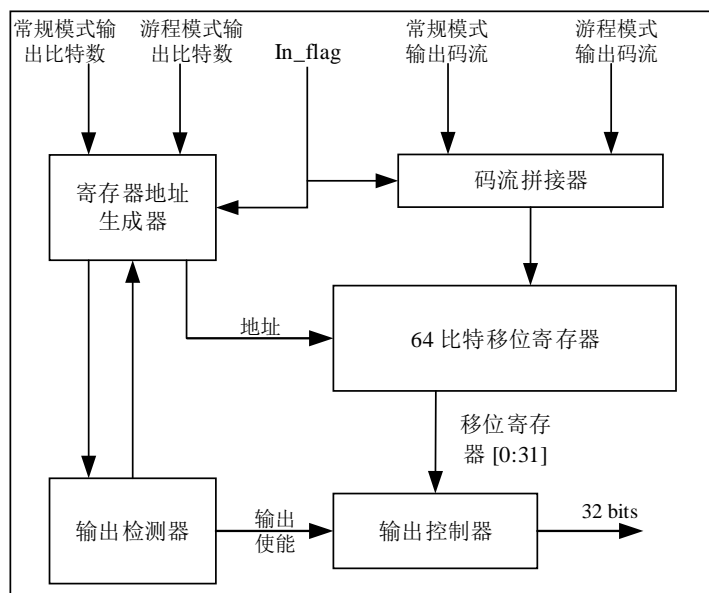


图 3-22 控制器结构图

3.4 本章小结

本章设计并实现了图像预处理模块，该模块能够将 bmp 格式的图像转换为本文所需要的二进制文件；设计并实现像素输入控制器模块，能够不间断地、正确无误地向 JPEG-LS 编码器内输入像素值；设计并实现了 JPEG-LS 图像压缩编码器，给出了优化方案：利用 FSM 完成模式选择的功能，保证了模式的快速切换；优化了常规模式编码的流水线设计方案；利用简单移位寄存器完成码流输出控制功能，节省硬件资源。本章设计的 JPEG-LS 无损图像压缩编码器实现正常的图像压缩功能，能够对 512×512 的灰度图像进行压缩并生成二进制码流文件，为后面的编码器抗 SEU 加固设计提供了基础。

第4章 编码器抗 SEU 加固设计

第 3 章详细介绍了 JPEG-LS 无损压缩编码器的设计过程，实现了对 512×512 8bit 灰度图像的正确压缩功能。本章将在第 3 章已完成编码器的基础上，分析编码器各个模块及图像压缩过程的特点，对编码器的各个部分设计实现具体的抗 SEU 加固方案。

4.1 模式选择器的抗 SEU 加固设计

在本文的设计中，模式选择器是不可或缺的重要一环。模式选择器的工作主要由五态的状态机完成，而在实际工作过程中，单粒子效应极有可能造成状态机工作状态的跳转从而发生错误，因此对状态机进行加固是提高模式选择器抗 SEU 能力的重要措施。

4.1.1 有限状态机编码

二进制编码是有限状态机最为常用的编码方式之一。但是，其编码方式具有一定的可靠性缺陷。当图像压缩编码器中的状态机发生单粒子翻转效应时，其状态将会跳转到错误状态，该错误不易被发现，最终将造成编码器的输出码流错误。此外，模式选择器虽然不负责像素的压缩编码功能，但是它负责向核心编码器传输数据，一旦模式选择器发生单粒子翻转故障，将导致整个 JPEG-LS 编码器的无法正常工作。因此，需要着重对模式选择器进行加固设计。

独热码（One-Hot）是只有一个比特为 1，其他全为 0 的一种码制，也就是说有多少个状态就有多少个比特位。本文中，模式选择器内的状态机为 5 个状态，则独热码编码为：00001，00010，00100，01000，10000。独热码编码与二进制编码、格雷码编码相比，虽然在一定程度上使用了较多的触发器，但是在状态比较时仅仅需要比较 1 个 bit，这在一定程度上简化了译码逻辑。在抗 SEU 方面，使用独热码编码的状态机码距要比二进制编码、格雷码的码距长，即使受到单粒子翻转效应的影响，也仅仅导致状态机状态跳转到未定义状态，而不会导致状态机跳转到错误状态^[48]。跳转到未定义状态的状态机，可以通过复位刷新的方式，使得状态机恢复初始状态。

4.1.2 有限状态机加固

三模冗余处理是一种对触发器应用最广泛的、同样也是最经典的一种加固方式，它的主要思路是把待加固的部分复制三份，再通过表决电路选择出未经单粒子翻转效应影响的正确结果。如果独热码编码结合三模冗余方式加固的有限状态机受到 2bit SEU 的影响，分析结果如图 4-1 所示。本文的状态机有 5 个状态，因此列出了 15 种 2bit SEU 对状态寄存器的影响情况，经过分析可知，对于前 5 种情况，虽然表决输出结果不正确，但是由于状态机跳转到了未定义状态，对最终的输出结果没有影响；对于后 10 种情况，表决输出结果正确。

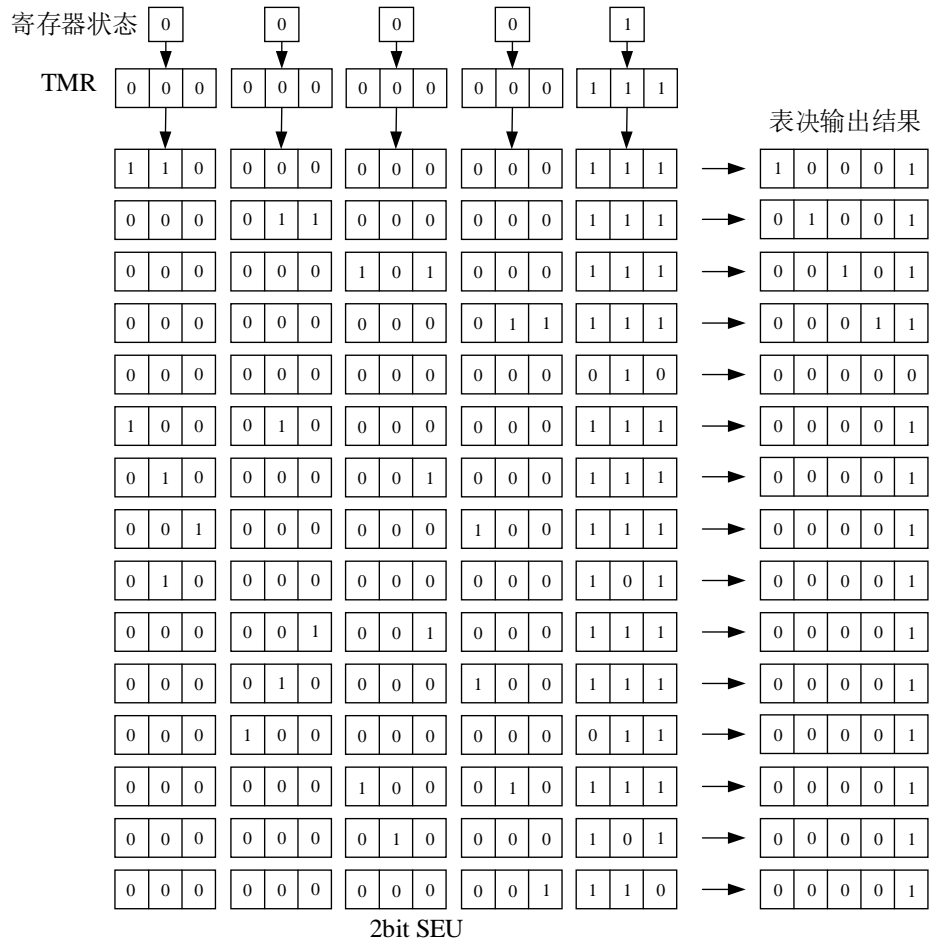


图 4-1 独热码结合 TMR 加固发生 2bitSEU 的情况

在本文的设计中，采用独热码结合三模冗余的方式对模式选择器进行抗单粒子翻转的加固设计。如图 4-2 所示，利用独热码对模式选择器中涉及的状态机进行编码，同时将编码后的现态寄存器进行三模冗余处理，最终利用多路表决器选择出正确的当前状态。

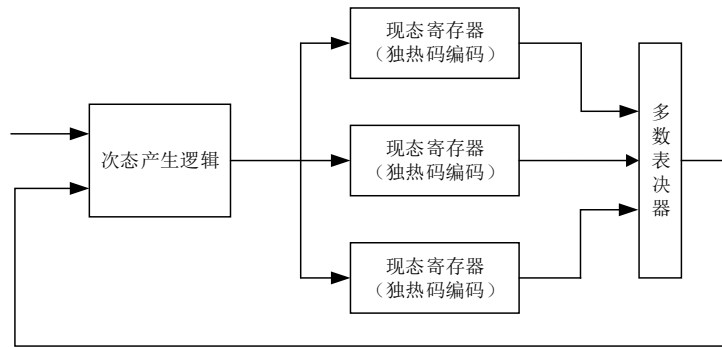


图 4-2 独热码结合三模冗余加固结构示意图

4.2 核心编码器的抗 SEU 加固设计

常规模式编码器和游程模式编码器是 JPEG-LS 无损图像压缩编码器的核心，对其进行抗 SEU 加固设计是重中之重。本文采用冗余设计和超时检测的方式对以上两个编码器进行加固。

4.2.1 三模冗余技术

三模冗余是一种常用的系统容错技术。虽然它会带来资源上的耗费，但是由于其编码电路和译码电路都十分简单易操作，因此被广泛应用于各类抗辐照电路的设计当中。

图 4-3 为最基本的三模冗余结构图，输入端数据 d_{in} 分别输入到三个 D 触发器中，如果一路触发器因受到 SEU 的影响而发生翻转，表决器依然可以根据另外两个 D 触发器输出的值 d_{out} 判断出正确值，并选择输出。表决器逻辑如图 4-4 所示。

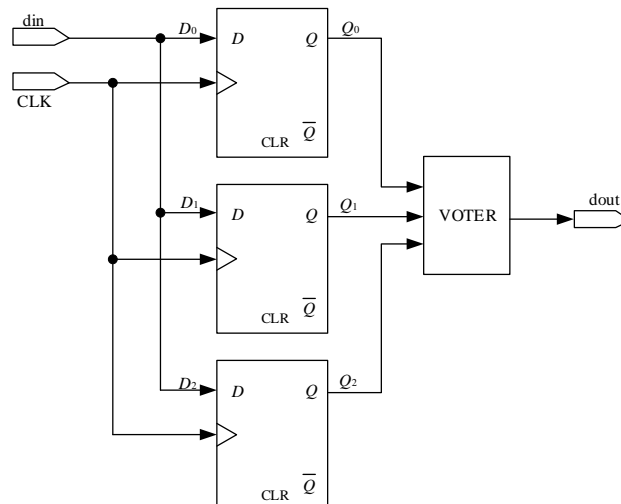


图 4-3 基本三模冗余

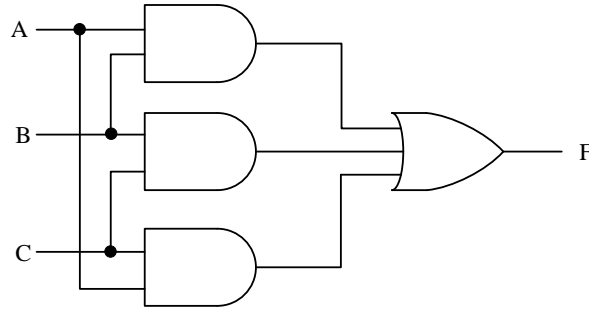


图 4-4 表决器逻辑

经过上述电路，当一路发生单粒子翻转效应时，通过表决器的三选二功能，可以选择出正确结果并输出。

4.2.2 超时检测技术

在本文的 JPEG-LS 的常规模式编码器的设计中，采用流水线技术加快常规模式编码速度的同时，也限制了内部各个子模块必须在特定的时间内完成各自功能。正常情况下，各个子模块将按部就班地执行自身功能，一旦受到单粒子翻转效应的影响，子模块的结束信号（关键信号）可能丢失，最终导致整个压缩编码器的输出结果错误。为此本文设计了超时检测器，用以检测由于单粒子翻转效应而导致的模块结束信号丢失的问题。

超时检测器的设计原理与“看门狗”电路类似，首先计算出常规模式编码器下各个子模块的运行时钟周期数，然后设计相应大小的计数器。当子模块开始工作时，计数器置为 0 且开始计数，计数到相应大小的值时，检测当前子模块结束信号是否有效，设计结构图如图 4-5 所示。

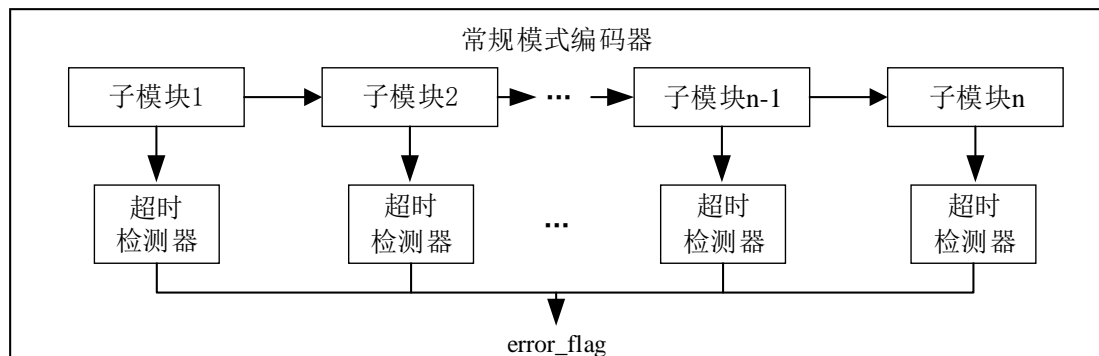


图 4-5 超时检测器工作示意图

正常情况下，计数器计数到相应值时，能够检测到子模块的结束信号。此时，超时检测器对图像压缩编码器无影响；当受到单粒子翻转效应时，计数器计数到

相应的值，检测器可能不会检测到子模块的结束信号。此时，超时检测器发送出标志位 `error_flag`，表示发生了故障，等待处理完错误恢复正常状态。

4.2.3 编码器加固

对于常规模式编码器，一方面采用三模冗余的方式对整个常规模式编码器进行加固设计，这样可以避免一路受到单粒子效应的影响而导致的系统故障；另一方面采用超时检测技术来监督内部各个子模块是否能够在特定的时间内输出结束信号，如果不能，则输出错误信号 `error_flag`。最终在三模冗余的前提下，如果有两路或以上都输出错误信号 `error_flag`，则表决电路无法正常输出结果，需要复位刷新后再进行正常编码，具体实现方式如图 4-6 所示。

对于游程模式编码器，进行整体的三模冗余设计，这样即使其中一份游程编码器受到单粒子翻转效应的影响而出现错误，另外两份游程编码器的结果也可以正常输出。

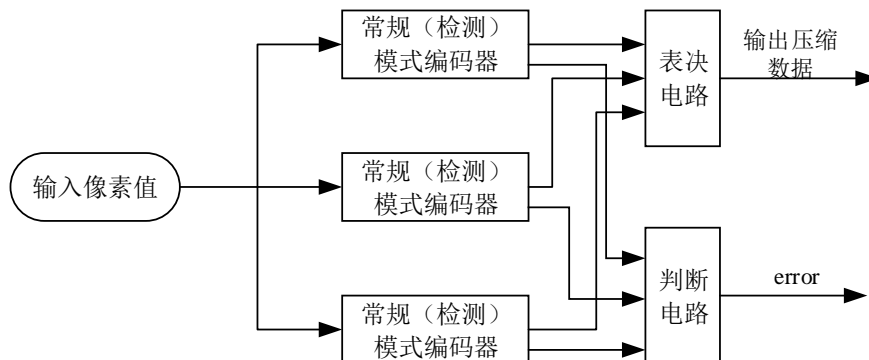


图 4-6 常规（检测）模式编码器加固

4.3 参数更新模块的抗 SEU 加固设计

在 JPEG-LS 编码器的设计过程中，使用的重要参数 $A[Q]$ 、 $B[Q]$ 、 $C[Q]$ 和 $N[Q]$ 都来自于存储器 RAM，参数更新模块的完成即是对 RAM 的读写操作。在每个像素点的压缩过程中， $A[Q]$ 、 $B[Q]$ 、 $C[Q]$ 和 $N[Q]$ 值都会被使用，并且用于压缩过程关键数据处理。参数更新模块将对其进行更新处理，因此上一个像素点的变量参数值将对下一个像素点的压缩过程产生影响。一旦存储器 RAM 中的参数数据受到单粒子翻转效应影响而发生变化，将导致后续图像像素点的压缩结果全部出错，进而无法恢复解压缩原图像。因此，有必要对 JPEG-LS 压缩编码器中涉及到的 RAM 存储器进行加固。

4.3.1 汉明码检纠错技术

本文采用汉明码（Hamming Code）实现对 RAM 存储器内数据的检纠错设计。汉明码不仅可以检测到 1 位错误，而且可以根据校正子与错误位置的关系进行错误纠正。在原来的 n 位数据的基础上，插入 k 位数据作为校验位，把原来的 n 位数据变为 m ($m=n+k$) 位编码是汉明码的实现方法。其中汉明码编码时要满足以下原则：

$$2^k - 1 \geq m \quad (\text{其中 } m=n+k) \quad (4-1)$$

式 (4-1) 为 Hamming 不等式。代码长度和检测位数的关系如表 4-1 所示。

表 4-1 代码长度与检测位数之间关系

N	K (最小值)
1	2
2~4	3
5~11	4
12~26	5
27~57	6

通过上表可知，数据位宽度为 8bit、9bit、13bit 时，对应的校验位数分别为 4bit、4bit、5bit。得到数据位与校验位，便可以根据具体检错纠错要求确定相应的位置与取值。

4.3.2 参数更新模块的 EDAC 设计

在本文的研究当中，RAM 的宽度有 8bit、9bit、13bit 三种类型，因此设计了三种汉明码检错纠错方式。

8bit 汉明码检错纠错方式如下：假定信息位是 $a_0 \sim a_7$ ，监督位是 $b_0 \sim b_7$ ，校正子为 $S_0 \sim S_3$ ，在确定校验位数为 4bit 的情况下，设计生成校验矩阵如图 4-7 所示。

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

图 4-7 校验矩阵

其中，矩阵的第 1~8 列对应信息位 $a_0 \sim a_7$ ，矩阵的第 9~12 列对应监督位 $b_0 \sim b_7$ 。根据校验矩阵得到监督位 $b_0 \sim b_7$ 与信息位的关系如表 4-2 所示。

表 4-2 监督位与信息位之间关系

监督位	信息位
b_3	$a_7 \wedge a_6 \wedge a_5 \wedge a_4$
b_2	$a_7 \wedge a_3 \wedge a_2 \wedge a_1$
b_1	$a_6 \wedge a_5 \wedge a_3 \wedge a_2 \wedge a_0$
b_0	$a_6 \wedge a_4 \wedge a_3 \wedge a_1 \wedge a_0$

根据监督位与信息位之间的关系，推算出校正子与数据位的关系如表 4-3 所示。

表 4-3 校正子与数据位之间的关系

校正子	数据位
S_3	$b_3 \wedge a_7 \wedge a_6 \wedge a_5 \wedge a_4$
S_2	$b_2 \wedge a_7 \wedge a_3 \wedge a_2 \wedge a_1$
S_1	$b_1 \wedge a_6 \wedge a_5 \wedge a_3 \wedge a_2 \wedge a_0$
S_0	$b_0 \wedge a_6 \wedge a_4 \wedge a_3 \wedge a_1 \wedge a_0$

最终通过校正子可知错码位置，其关系如表 4-4 所示。

表 4-4 校正子与错码位置之间关系

$S_3 S_2 S_1 S_0$	错码位置
1100	a_7
1011	a_6
1010	a_5
1001	a_4
0111	a_3
0110	a_2
0101	a_1
0011	a_0
1000	b_3
0100	b_2
0010	b_1
0001	b_0

根据以上关系设计结构如图 4-8 所示。信息码在写使能有效时，通过数据总线传输到 EDAC 编码模块，完成监督位生成、信息位与监督位整合后存储到存储单元中；在读使能有效时，存储单元中数据经过数据总线传输到 EDAC 解码模块，完成校正子生成、检错与纠错后输出。9bit、13bit 的汉明码检错纠错方式与之类似，不再详细表述。

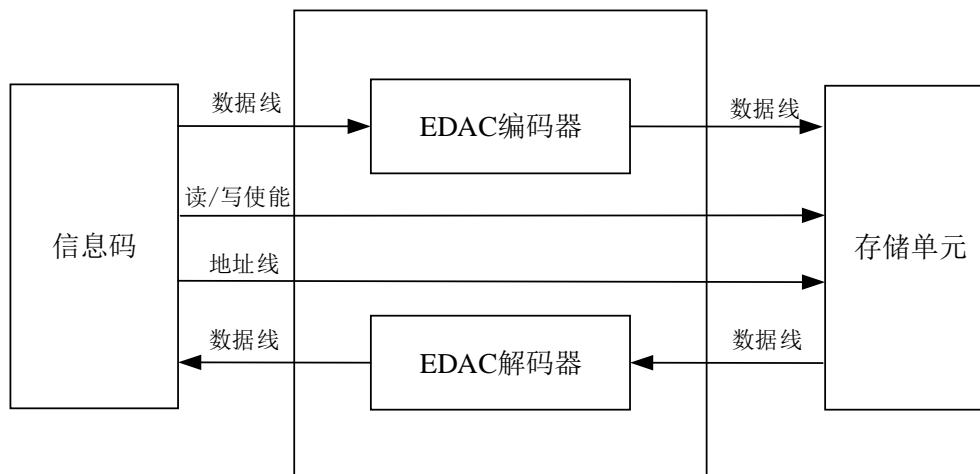


图 4-8 EDAC 设计框图

4.4 帧间图像独立处理

4.4.1 独立处理分析

在图像处理过程中，图像压缩编码器一帧一帧地进行处理，即上一帧图像的压缩结果对下一帧的图像压缩内容没有影响。利用图像处理的这一特点，对每帧图像进行独立处理，即在对每一帧图像进行压缩之前，编码器将对自身系统内相关的参数及触发器进行复位处理。

由于 JPEG-LS 标准本身的特殊性，即待压缩像素点与周围像素点数据相关性大。因此，将一帧的图像压缩过程作为一个复位处理的基本单元。如果在一帧图像压缩过程中受到 SEU 的影响，那么在本帧图像压缩过程结束后进行复位操作，从而避免辐照效应的错误累积。

4.4.2 复位信号设计

异步复位设计有着独特的优势：复位信号直接连接到触发器的复位端口，复位路径上没有延时，做到了复位的实时性。但是，异步复位也有缺点，组合逻辑

电路将产生毛刺，毛刺的存在有可能导致误触发复位逻辑。另外，复位逻辑的撤离（即复位信号无效）的瞬间可能与时钟有效沿很接近，从而导致亚稳态的产生。

针对以上两个问题做了具体的分析。如果数据输入在时钟边沿来临之前或之后的有限时间间隔内发生了变化（即输入信号不满足建立时间或保持时间），则触发器将进入亚稳态。此时，器件将输出一个 0 和 1 之间的输出，并且无法进行译码。如图 4-9 所示，当触发器处于亚稳态，输出会在高低电平之间进行波动，一段时间后恢复到一个不确定（0 或 1）的稳定值。

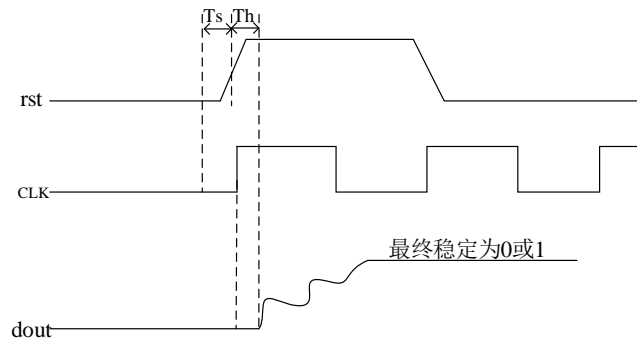


图 4-9 亚稳态问题

电路在某种输入的作用下，输出在不应当发生跳变的时候发生了跳变，称之为发生了静态冒险。一旦静态冒险产生的毛刺作用于异步复位信号上，结果将导致系统不必要复位，造成系统紊乱，导致输出结果错误。在太空复杂的辐照环境中，这种问题变得尤为突出。如图 4-10（a）所示，复位逻辑产生的毛刺不能被屏蔽掉，而是直接对高电平有效的复位端使能，结果将造成不必要的复位。解决方法是将复位逻辑放置在触发器的同步复位端或等价的 D 端逻辑，如图 4-10（b）所示。由于触发器在时钟边沿采样，因此可屏蔽掉不必要的毛刺，并且可以利用静态时序分析工具完成验证，从而保证了设计的可靠性。

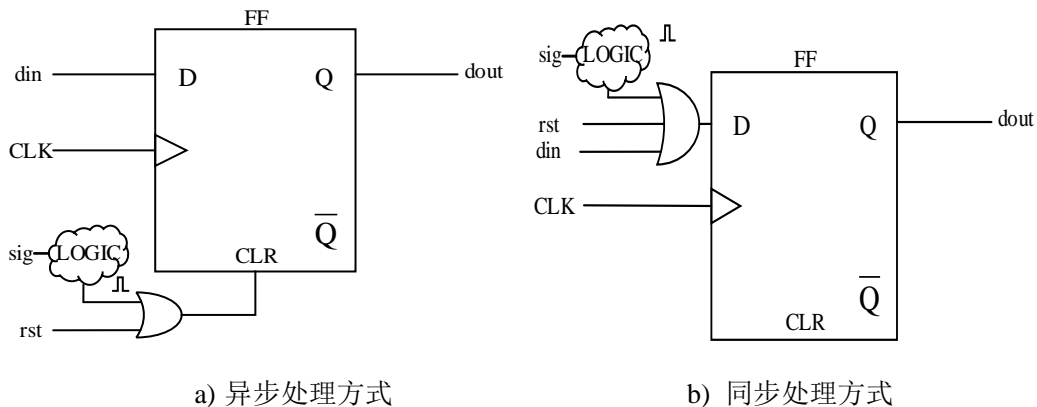


图 4-10 异步处理与同步处理方式

同步处理的问题是复位逻辑信号需从 D 端输入，而每个触发器都进行复位，将会增大硬件资源的开销。如果选择使用异步复位端直接复位，则采用异步复位、同步释放的方式进行复位操作，如图 4-11 所示。复位逻辑信号与全局复位信号 *rst* 经过触发器的采样后，过滤掉毛刺。输入到异步复位端口，经过双缓冲器（双寄存器）后，亚稳态问题得到解决。

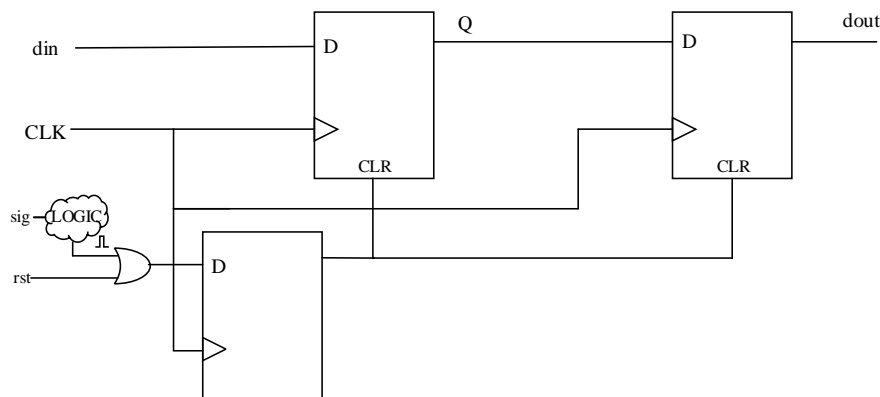


图 4-11 异步复位、同步释放处理方式

4.5 本章小结

本章针对 JPEG-LS 图像压缩编码器中各个易受到单粒子效应的模块的特点，分别设计了不同的抗 SEU 加固方案。模式选择器采用独热码和三模冗余的方式器状态机进行加固；核心编码器采用超时检测和三模冗余的方式对其进行加固；由于参数更新模块使用到 RAM，因此利用汉明码的纠检错能力对存储单元进行了加固；最后，根据 JPEG-LS 标准的特点，每帧图像压缩结果互不影响，采用帧间图像独立处理技术进行处理，减少了 SEU 的累积。

第5章 功能验证与加固效果分析

本章对 JPEG-LS 无损压缩编码器进行功能验证及抗 SEU 加固验证。对未受到 SEU 影响常态工作模式的编码器进行功能验证，设计验证平台结构及验证方案，验证功能仿真结果的正确性。利用故障注入技术，模拟 SEU 对编码器造成的影响，验证加固方案的有效性，并对整体加固效果做了评估。最后，利用 Vivado 软件对加固前后的编码器进行综合、布局布线后，在 FPAG VC707 上进行了验证。

5.1 常态工作模式功能仿真验证

5.1.1 仿真验证平台设计

为了验证本文设计的 JPEG-LS 无损压缩编码器的功能的正确性，设计了专门的仿真验证平台，如图 5-1 所示。

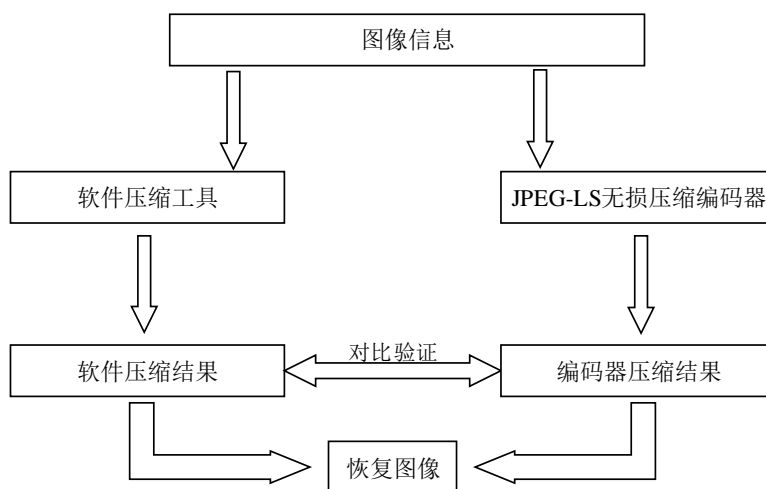
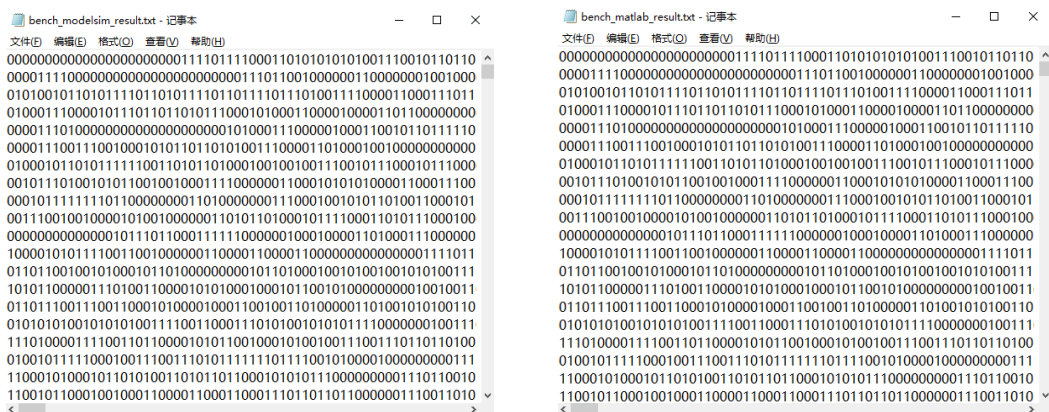


图 5-1 仿真验证平台

首先，同一幅图像信息分别送入已有的 JPEG-LS 软件压缩工具和本文设计的无损压缩编码器，图像信息在编码器内经过图像预处理与压缩编码等过程后输出码流结果。同时，软件压缩工具输出该图像的压缩结果。利用工具比较两个压缩码流结果是否一致，如果结果一致，说明本文设计的无损压缩编码器功能正确，否则不正确。一致的情况下，进一步利用软件的解压缩功能将软件压缩结果与编码器压缩结果分别解压缩恢复原图像，观察其恢复图像是否一致。

5.1.2 功能验证结果

为了使整个 JPEG-LS 编码器可以正常运行，对其中的各个模块进行了单独的仿真验证。最终结果表明，图像预处理模块、像素输入控制器模块、模式选择器模块、常规模式编码器、游程模式编码器和输出码流控制器模块都能独立输出正确结果。随后，对它们进行例化并联合仿真，最终得到了自然图像的压缩码流文件，如图 5-2（a）所示。使用软件压缩工具得到同一幅图像的压缩码流文件，如图 5-2（b）所示。

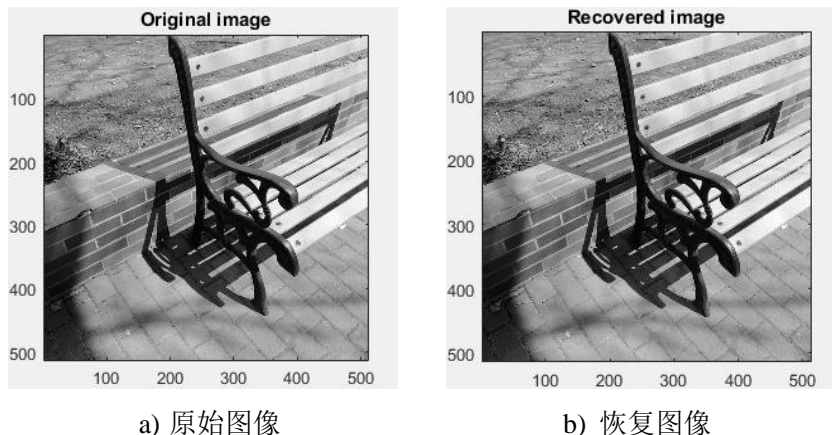


a) 编码器仿真结果

b) 软件工具压缩结果

图 5-2 两种方式得到的压缩码流文件

利用 perl 语言对两个文本进行了比较，比较结果表明两个文本内容一致。对大量的国际标准测试图像和自然图像进行了验证，都获得了二者压缩结果一致性的结论，表明本文设计的 JPEG-LS 编码器功能正确。再次将 JPEG-LS 编码器处理的结果放入软件工具中进行解压缩，并与原图像进行对比，取其中 1 幅图像的对比图像，如图 5-3 所示，表明本文设计的压缩码流结果可以恢复出原图像。



a) 原始图像

b) 恢复图像

图 5-3 原始与恢复图像对比

利用 ModelSim 仿真，对 10 幅图像进行了测试，通过在输出码流控制器模块加入检测计数器的方式，对一幅图像用到的常规模式编码器和游程模式编码器使用次数进行了统计分析，并在时钟周期为 10ns 的情况下，统计其压缩时间，如表 5-1 所示。

表 5-1 10 幅图像统计分析

图像名称	常规次数	游程次数	游程频次	压缩时间
Aerial	260830	806	0.31%	0.0265s
Airplane	256909	2643	1.02%	0.0270s
Babbon	262116	25	0.01%	0.0262s
Fingerprint	262022	102	0.04%	0.0263s
Lena	261429	491	0.19%	0.0264s
Yacht	260164	1259	0.48%	0.0266s
Tree	261925	122	0.05%	0.0263s
Peppers	261556	406	0.15%	0.0264s
Xiaoxi	262076	30	0.01%	0.0262s
Bench	261489	369	0.14%	0.0263s

结果表明，尽管游程模式编码器没有采用流水线处理方式，由于其在实际压缩过程中使用率很低，所以不会影响整体的压缩速率，编码器平均压缩时间为 0.0264 秒/帧。

5.2 抗 SEU 加固测试

5.2.1 加固仿真测试

编码器抗 SEU 加固的措施将通过在 ModelSim 仿真过程中利用命令行的方式来进行测试。观察相应的加固措施是否有效。

(1) **模式选择器加固有效性测试** 模式选择器采用的加固方式是独热码和三模冗余，分别测试发生 1 位 SEU 和 2 位 SEU 时，模式选择器内状态机结果。

如图 5-4 所示，对寄存器随机注入 1 位 SEU，state_1 由 5'b00100 变为 5'b00101，但是最终输出 state 状态并没有因此发生改变，表明该加固方式可以容忍 1 位 SEU。

/JPEG_LS_top_tb/jpeg_ls/sc/state	5'b00100	5'b00100				
/JPEG_LS_top_tb/jpeg_ls/sc/state_1	5'b00100	5'b00100		5'b00101		5'b00100
/JPEG_LS_top_tb/jpeg_ls/sc/state_2	5'b00100	5'b00100				
/JPEG_LS_top_tb/jpeg_ls/sc/state_3	5'b00100	5'b00100				
/JPEG_LS_top_tb/jpeg_ls/sc/sc_error_flag	1'h0					

图 5-4 模式选择器有限状态机发生 1 位 SEU 情况

对寄存器随机注入 2 位 SEU，如图 5-5 所示，a) 为情况 1，state_1 由 5'b00100 变为 5'b01100，同时 state_2 由 5'b00100 变为 5'b00101。可见，最终的输出 state 状态依然保持不变。b) 为情况 2，state_1 由 5'b00100 变为 5'b01100，同时 state_2 由 5'b00100 变为 5'b01100。可见，最终的输出 state 状态跳转到未定义态 01100，此时错误标志位 sc_error_flag 有效。

/JPEG_LS_top_tb/jpeg_ls/sc/state	5'b00100	5'b00100				
/JPEG_LS_top_tb/jpeg_ls/sc/state_1	5'b00100	5'b00100		5'b01100		5'b00100
/JPEG_LS_top_tb/jpeg_ls/sc/state_2	5'b00100	5'b00100		5'b00101		5'b00100
/JPEG_LS_top_tb/jpeg_ls/sc/state_3	5'b00100	5'b00100				
/JPEG_LS_top_tb/jpeg_ls/sc/sc_error_flag	1'h0					

a) 有限状态机发生 2 位 SEU 情况 1

/JPEG_LS_top_tb/jpeg_ls/sc/state	5'b01100	5'b01100		5'b01100		
/JPEG_LS_top_tb/jpeg_ls/sc/state_1	5'b01100	5'b01100		5'b01100		
/JPEG_LS_top_tb/jpeg_ls/sc/state_2	5'b01100	5'b01100		5'b01100		
/JPEG_LS_top_tb/jpeg_ls/sc/state_3	5'b00100	5'b00100				
/JPEG_LS_top_tb/jpeg_ls/sc/sc_error_flag	1'h1					

b) 有限状态机发生 2 位 SEU 情况 2

图 5-5 模式选择器有限状态机发生 2 位 SEU 情况

结果表明，本文模式选择器的加固方式可以容忍 1 位 SEU 错误，发生 2 位 SEU 错误可跳转到未定义态而不产生后续错误。

(2) 三模冗余有效性测试 选取常规模式编码器中的预测模块信号 Px 进行观察。首先，模拟一路信号 Px_tmr1 受到单粒子翻转效应的影响，信号值由 8'b10001100 转变为 8'b10101100，而其他两路未受到单粒子翻转效应的影响，如图 5-6 所示。

/JPEG_LS_top_tb/jpeg_ls/com/Px_tmr1	8'b10101100	8'b10101100		8'b10001100	8'b10101100
/JPEG_LS_top_tb/jpeg_ls/com/Px_tmr2	8'b10100001	8'b10101100		8'b10001100	
/JPEG_LS_top_tb/jpeg_ls/com/Px_tmr3	8'b10100001	8'b10101100		8'b10001100	
/JPEG_LS_top_tb/jpeg_ls/com/Px	8'b10100001	8'b10101100		8'b10001100	

图 5-6 Px 信号三模冗余仿真图

可以看到，虽然信号 Px_tmr1 发生了错误，但是最终的输出信号 Px 并未产生错误，依然保持原有的输出结果。可以证明，三模冗余方式可以提高编码器的抗 SEU 的能力。

(3) 超时检测技术有效性测试 选择常规模式编码器中的 K 值计算模块进

行观察。该模块中的 K_cnt 信号受到 SEU 的影响发生翻转将导致不能按时输出。利用 ModelSim 模拟其中一路 (tmr1) 的 K_cnt 受到 SEU 的影响由 4'b1001 变为 4'b1000, 如图 5-7 所示。

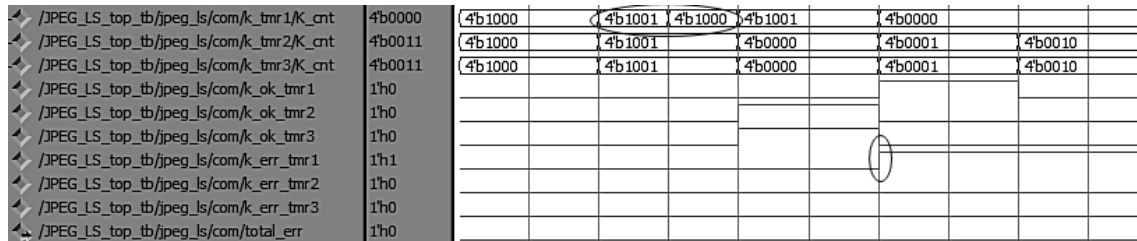


图 5-7 超时检测到 1 路错误

结果发现, 该路输出信号标志位 k_ok_tmr1 未按时置为 1。此时, 错误标志位 k_err_tmr1 有效, 但是并没有触发错误警报, total_err 为 0, 编码器可以继续编码。

当另外一路 (tmr2) 内的 K_cnt 也受到模拟单粒子翻转效应影响, 由 4'b1000 变为 4'b1010 时, 如图 5-8 所示。结果发现, 该路输出信号标志位 k_ok_tmr2 同样未按时置为 1。此时, 错位标志位 k_err_tmr2 有效, 同时触发错误警报, total_err 置为 1, 编码器内的三模结构出现两路及两路以上错误, 需要停止工作。

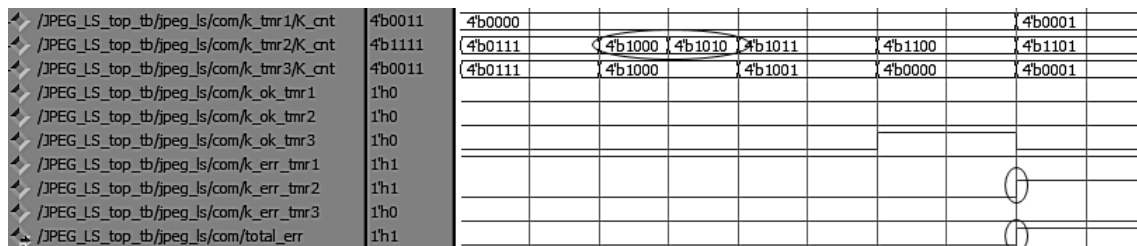


图 5-8 超时检测到 2 路错误

(4) 参数更新模块加固有效效应测试 对参数更新模块的 RAM 进行模拟 SEU 的单比特故障注入仿真, 验证该模块加固设计的正确性, 如图 5-9 所示。

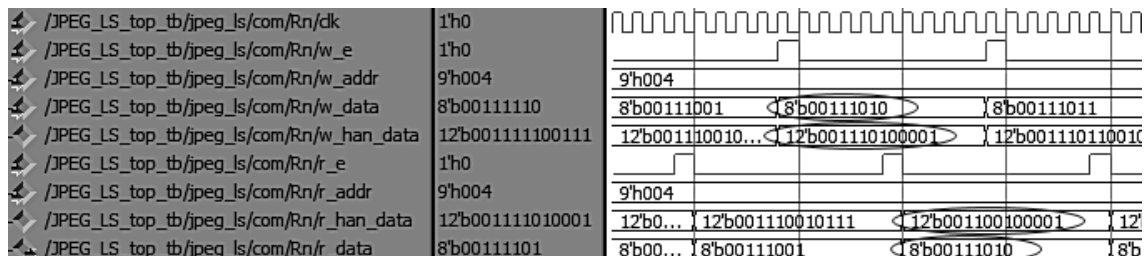


图 5-9 汉明码纠检错仿真模拟波形图

对其中 8bit 位宽的 RAM 的 1 地址数据改变一个比特位, 由原来的 8'b00111010 改变为 8'b00110010, 模拟发生了单粒子翻转。可以看到, 经过编码后 w_han_data 为 12'b00111010001, 译码之前的数据位 r_han_data 为 12'b001100100001, 说明成

功地模拟了单比特翻转现象。经过译码后，输出端信号 r_data 值为 $8'b00111010$ ，与原写入数据 w_data 一致，表示经过汉明码的纠错译码后，正确地将原数据输出。

5.2.2 加固效果评估

基于已完成的 JPEG-LS 编码器，对其进行模拟单粒子翻转的注入。分别对加固前和加固后的 JPEG-LS 编码器注入故障，并利用同一软件解压缩工具，分别将二者输出的结果恢复成图像。加固前与加固后编码器恢复图像对比如图 5-10 所示。a) 为加固前编码器发生单粒子翻转效应时输出的压缩码流经解码恢复的图像，b) 为加固后编码器发生单粒子翻转效应时输出的压缩码流经解码恢复的图像。可以看到，JPEG-LS 无损压缩编码器受到辐照影响之后，后续码流无法解码恢复原图像，证明了本文加固方式的有效性。

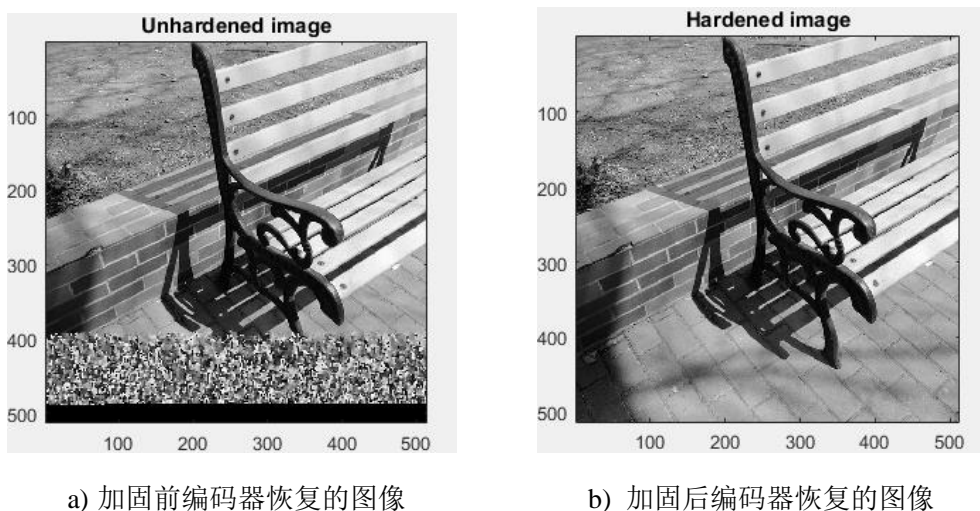


图 5-10 加固前与加固后编码器恢复图像

为了评估该图像压缩编码器的整体加固设计效果，采用文献[49]的方法对其进行了故障注入。故障注入类型为单比特 SEU，注入目标为该编码器的所有寄存器，每位故障注入次数为 100 次，观察点个数为 100 个，监视时长（注入故障后，黄金结果与注入故障后的结果对比时间）为注入故障后的 3000 个时钟周期。可以得到具体的分析统计结果如图 5-11 所示。

从图中可以看到，各个模块在加固后平均错误率具有明显的降低，由于常规模式编码器采用子模块三模冗余，因此对整个模块进行故障注入后，结果仍然具有一定的错误率。模式选择器未对有限状态机外的部分进行加固，因此仍然有 1.83% 的错误率。

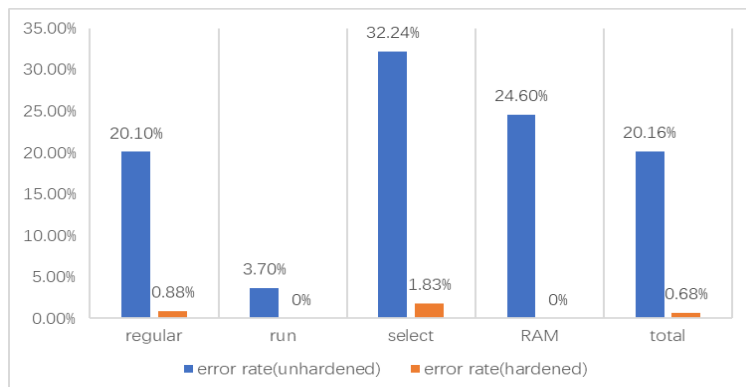


图 5-11 加固前后平均错误率对比

通过以下公式可以计算出加固前与加固后该编码器的整体错误率：

$$\sigma_{\text{unharden}} = \frac{\sum_{i=1}^i \sigma_{\text{mod}}}{N} = 20.16\% \quad (5-1)$$

其中， σ_{unharden} 为加固前编码器整体错误率， σ_{mod} 为编码器子模块的错误率。N 为模块数量，此处取值为 4。

$$\sigma_{\text{harden}} = \frac{\sum_{i=1}^i \sigma_{\text{mod}}}{N} = 0.68\% \quad (5-2)$$

其中， σ_{harden} 为加固后编码器整体错误率。对各个子模块的错误率取得平均值得到编码器整体错误率。由公式（5-1）、公式（5-2）可得到，当对该编码器的各个部分有针对性的进行加固后，编码器的整体错误率降低了 19.48%。

5.3 FPGA 验证

5.3.1 FPGA 硬件平台搭建

完成 JPEG-LS 编码器功能验证之后，利用 Xilinx 的 Virtex7 VC707 型号的 FPGA 开发板进行硬件实现。设计 FPGA 硬件平台如图 5-12 所示。

在整个平台搭建的过程中，除了原有的编码器模块，还加入了比较器、4 个 IP 核，包含 2 个单端口 ROM、1 个伪双口 RAM 和一个时钟分频 IP。由于 VC707 开发板提供的晶振时钟为 200MHz 的差分时钟，因此需要调用时钟 IP 对其进行处理，一方面进行降频到编码器所需的时钟频率，另一方面将差分时钟转为单端时钟。通过 Vivado 调用 Clocking Wizard 中的 IP 核来定制。ROM1 用于存储图像压缩前的二进制码流数据，经过 JPEG-LS 无损压缩后存储到 RAM 中。ROM2 用于存储

软件压缩工具压缩后的二进制码流数据，最终比较器的功能是将 RAM 与 ROM2 中的数据逐位比较，并给出比较结果。ROM 和 RAM 均通过 Vivado 调用 BLOCK MEMEORY 中的 IP 核来定制。

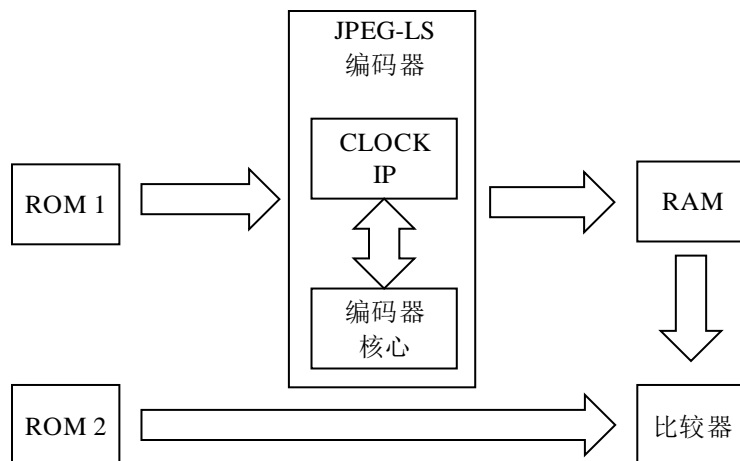


图 5-12 硬件平台结构

5.3.2 结果验证与性能分析

选用 Xilinx 的 Virtex7 VC707 FPGA，搭建好硬件平台后，将相应的设计文件导入 Vivado 中。编写约束文件，设定综合选项，绑定管脚，将定制 IP 核与设计文件一起进行综合编译，经过布局布线后生成比特流。连接好 FPGA 开发板后，将比特流下载。如图 5-13 所示为 VC707 FPGA 开发板整体图。



图 5-13 VC707 FPGA 开发板整体图

其中，SW7 为初始化复位键，按下后编码器内部所有逻辑将被复位；SW5 为压缩开始键，模拟图像像素数据进入压缩编码器的开始信号，按下后编码器开始

读取 ROM1 内的数据并对其进行无损压缩；SW4 为比较开始键，当编码器压缩工作结束，按下此键后系统分别从 ROM2 和 RAM 中逐地址读取数据并进行比较。

图 5-14 为验证结果，三幅图都是 FPGA 的局部 LED 部分。首先，将编码器进行全局复位操作，此时如图 5-14 a) 所示，所有 LED5~LED7 灯熄灭，表示复位成功；按下 SW5 键，编码器开始进行无损压缩工作，LED7 灯亮后代表编码工作结束，如图 5-14 b) 所示；按下 SW4 键，系统比较压缩结果正确性，LED6 灯亮后代表比较工作结束，LED5 灯亮后代表比较结果表明二者数据一致，FPGA 验证编码器功能结果正确，如图 5-14 c) 所示。



图 5-14 验证结果

使用 Vivado 综合后，得到了加固前后包括验证部分在内设计的资源利用情况，如表 5-2 所示。

表 5-2 编码器加固前后资源利用对比

逻辑器件	加固前		加固后		总数量
	使用数量	资源占比	使用数量	资源占比	
Slice LUTs	3481	1.15	5962	1.96	303600
Slice Registers	1316	0.22	2705	0.45	607200
Block RAM	155.5	15.1	155.5	15.1	1030
Bonded IOB	10	1.43	10	1.43	700
BUFG	2	15.1	2	15.1	32
MMCM	1	7.14	1	7.14	14

从表 5-2 可以看出，加固后的硬件资源占比都较加固前有了大幅度的增加，说明冗余加固设计带来了资源耗费的增加。同时利用工具综合出该设计的最高时钟频率为 112MHz。在实际 FPGA 验证过程中，由于 VC707 FPGA 开发板提供 200MHz 的差分时钟，所以利用 CLOCK IP 将其转化为 100MHz 的单端时钟再进行使用。此时，压缩一幅 512×512 8bit 的灰度图像需要时间大约为 0.0264s，即利用该图像压缩编码器 1s 可以压缩 38 帧图像，符合图像压缩的速度要求。

5.4 本章小结

本章对常态工作模式的编码器进行了功能验证，设计实现了仿真验证平台，结果证明编码器能够正确地对 512×512 8bit 的灰度图像进行无损压缩，平均压缩时间为 38 帧/s。利用故障注入技术，模拟 SEU 对编码器造成的影响，对加固后的编码器进行了测试，验证了本文加固方案的有效性，并对加固效果做了评估，编码器错误率可降低 19.48%。利用 Vivado 对加固前后的编码器进行综合，分析了综合结果，并在 Xilinx 的 Virtex7 VC707 FPGA 开发板上搭建了硬件平台，对编码器功能进行了验证。

结 论

本文设计并实现了 JPEG-LS 无损压缩编码器,并对其进行抗 SEU 加固设计。验证了硬件编码器功能的正确性,测试了加固方案的有效性,而且对加固效果做了评估。利用 Xilinx 的 Virtex-7 VC707 FPGA 开发板,验证了加固前后编码器的设计。论文取得的主要研究成果如下:

(1) 深入研究了 JPEG-LS 无损压缩算法的上下文建模、常规模式编码和游程模式编码流程,确定了基于 JPEG-LS 的压缩编码器所需重要参数的取值范围。并且设计了像素输入控制器,调控像素矩阵输送到编码器内,实现了准确、及时地为编码器提供码流的功能,并通过仿真验证了其功能正确性。

(2) 采用 Verilog HDL 语言实现了基于 JPEG-LS 的无损压缩编码器。利用有限状态机的方式调控选择编码模式,实现常规模式与游程模式的快速切换。采用流水线技术对常规模式编码器进行处理,提高了编码速率。利用简单的 64 比特移位寄存器与拼接器即实现了输出码流控制器,完成了最终输出码流的整合,节省了硬件资源。

(3) 针对 JPEG-LS 编码器的特点,对不同部分采用了不同的加固措施。对模式选择器内的 FSM 使用独热码结合三模冗余方式加固,可以容忍 1 位 SEU 错误,2 位 SEU 错误可跳转到未定义态而不产生后续错误;核心编码器采用三模冗余和超时检测方式进行加固;利用 EDAC 技术对基于 RAM 的参数更新模块加固;根据帧间图像压缩时,数据互不影响的特点,采用帧间图像独立处理的方式减少 SEU 的累积。通过仿真测试,验证了加固方案的有效性。

(4) 评估了编码器整体加固效果,并利用 Xilinx 的 Virtex7 VC707 FPGA 对编码器进行了硬件实现。利用故障注入工具向硬件编码器注入故障,评估加固前后编码器发生 SEU 时编码错误率,结果表明本文采取的加固措施将 JPEG-LS 编码器错误率降低 19.48%。对加固前后的编码器分别进行综合,利用 FPGA 搭建了硬件测试系统,验证了编码器设计的正确性。

综上,本文设计并实现了一款 JPEG-LS 无损压缩编码器,可对 512×512 8 bit 灰度图像正确压缩。编码器进行了抗 SEU 加固设计,编码器错误率降低 19.48%。

参考文献

- [1] 吴乐南. 静止图象压缩标准新进展[J]. 电脑应用技术, 1998(42): 8-11.
- [2] 刘春香, 郭永飞, 李宁. 星上多通道遥感图像的实时合成压缩[J]. 光学精密工程, 2013, 21(2): 445-453.
- [3] 陈佳怡, 韩庆龙, 祝名. 航天器用抗辐照 FPGA 器件需求型谱设计方法研究[J]. 电子产品可靠性与环境试验, 2018(1): 80-85.
- [4] ISO/IEC, Information technology-Lossless and near-lossless compression of continuous-tone still images, International Standard ISO/IEC 14495-1 and ITU Recommendation T.87, 1999.
- [5] Weinberger M J. LOCO-I: A low complexity, context-based, lossless image compression algorithm[C]// IEEE proc. on Data Compression Conference, 1996. IEEE, 1996: 140-149.
- [6] 陈益刚, 邓家先, 谢凯明. 基于 LOCO-I 算法的图像压缩比控制[J]. 电视技术, 2015, 39(02): 57-60.
- [7] Jakhetiya v, AUOC, JAISWAL S, et al. Lossless/near Lossless Compression Using Bilateral Filter and Symmetrical Based Bias Cancellation [C]//6th International Symposium on Communication, Control and Signal Processing. Piscataway. IEEE, 2014: 77-80.
- [8] CHENJ, CHENTS, LINC, et al. A simple JPEG-LS Compressed Technique for 2DGE Image with ROI Emphasis[J]. The Imaging Science Journal, 2015, 63(2): 76-80.
- [9] Song Hongmei, Xu Xueqing, Mu Haiwei, et al. Realization and Performance of Lossless Image-compress JPEG-LS Algorithm[J]. Optical Instruments, 2014(4): 315-318.
- [10] Chen Yigang, Deng Jiaxian, Xie Kaiming. Image Compression Encryption Algorithm Based on Compression Ratio Control of JPEG-LS[J]. Journal of Computer Applications, 2015, 35(1): 93-98.
- [11] Weinberger M . The LOCO-I lossless image compression algorithm : Principles and standardization into JPEG-LS[J]. IEEE Trans. Image Process. 2000, 9(8):

1309-1324.

- [12] Z Wang, A Michael, S Wahl, P Werner, S Simon. A memory efficient parallel lossless image compression engine for high performance embedded systems[C]// Image and Signal Processing and Analysis. Busan, 2011: 390-395.
- [13] Bedi S , Edirisinghe E A , Grecos G . Improvements to the JPEG-LS prediction scheme[J]. Image and Vision Computing, 2004, 22(1):9-14.
- [14] Kau L J , Lin S W . High performance architecture for the encoder of JPEG-LS on SOPC platform[C]// Signal Processing Systems. IEEE, 2013:141-146.
- [15] Mert Y M . Low complexity scheme with JPEG-LS for near-lossless, multi-component and selective compression[C]// Spie Commercial + Scientific Sensing & Imaging. 2017: 1-11.
- [16] 程子敬. 遥感图像高保真高速实时压缩技术研究[D]. 北京: 北京航空航天大学, 2001: 10-15.
- [17] 刘荣科, 张晓林, 聂振钢. 无损图像压缩国际标准 JPEG-LS 的抗差错性分析 [C]// 中国航空学会信号与信息处理专业学术会议. 2001: 178-182.
- [18] Niu H , Shang Y , Yang X , et al. Design and research on the JPEG-LS image compression algorithm[C]// 2010 Second International Conference on Communication Systems, Networks and Applications. IEEE, 2010: 232-234.
- [19] 戴鑫. JPEG-LS 图像无损压缩 IP 核的 FPGA 设计[D]. 太原理工大学, 2010: 41-52.
- [20] 寇计萍. 基于 JPEG-LS 的多重加密与单重加密算法研究 [D]. 海南大学, 2016: 24-28.
- [21] 阙恒, 王渊峰, 武凤霞. 一种新的 JPEG-LS 压缩算法的低延迟实现方案[J]. 集成电路应用, 2017, 34(06): 23-27.
- [22] 林金茂. SDR 平台抗 SEU 性能评价关键技术研究[D]. 国防科学技术大学, 2009: 25-36.
- [23] CHEN H, BINDLEY G, LU P. Use of solder mask as a protective coating for radiation detector [M]. US. 2009.
- [24] ARCHER J S. Hardening of spacecraft structures against momentary high level radiation exposure using a radiation shield [M]. US. 1989.
- [25] 王红举. 星载 ATP 控制系统容错设计及单粒子效应加固技术研究[D]. 中国科学院大学. 2016: 13-15.
- [26] Garg R, Khatri S P. A novel, highly SEU tolerant digital circuit design

- approach[C]// Computer Design, 2008. ICCD 2008. IEEE International Conference on. IEEE, 2008: 14-20.
- [27] Shayan M, Singh V, Singh A D, et al. SEU tolerant robust memory cell design[C]// On-Line Testing Symposium (IOLTS), 2012 IEEE 18th International. IEEE, 2012: 13-18.
- [28] A. Lopes Filho, R. d'Amore. A tolerant JPEG-LS image compressor foreseeing COTS FPGA implementation[J]. Microprocessors and Microsystems 49 (2017), 2017: 54–63.
- [29] J. Anwer, M. Platzner, S. Meisner. FPGA Redundancy Configurations: An Automated Design Space Exploration[C]// Parallel & Distributed Processing Symposium Workshops (IPDPSW), 2014: 275-280.
- [30] V. M. Goncalves Martins, J. G Reis, H. C. C, et.al. FPGA redundancy recovery based on partial bitstreams for multiple partitions[C]// 2015 16th Latin-American Test Symposium (LATS), 2015: 1-4.
- [31] M. Watanabe. Triple modular redundancy on parallel-operation-oriented FPGA architectures for optical communications[C]// 2015 IEEE International Conference on Space Optical Systems and Applications (ICSOS), 2015: 1-4.
- [32] 谭新宇. 航天图像压缩芯片的抗辐照设计[D]. 西安: 西安电子科技大学. 2014: 19-35.
- [33] 王波. 星载图像压缩编码器可靠性技术研究是实现[D]. 西安: 西安电子科技大学. 2015: 37-57.
- [34] 董珊. 星载遥感图像实时处理专用芯片的抗辐照电路设计[D]. 北京: 北京理工大学. 2016: 47-67.
- [35] 赵静. 深空探测载荷图像压缩系统的设计与实现[D]. 西安: 西安电子科技大学. 2017: 31-46.
- [36] M. Rebaudengo, M. Sonza Reorda and M. Violante. A New Functional Fault Model for FPGA Application-Oriented Testing[C]// 17th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, 2002: 372-380.
- [37] S. Baloch, T. Arslan and A. Stoica. Design of a Novel Soft Error Mitigation Technique for Reconfigurable Architectures[C]// IEEE Aerospace Conference, 2006: 1-9.

- [38] 牛仁杰. 星载 ATP 图像处理平台关键技术研究[D]. 北京: 中国科学院大学. 2018: 17-25.
- [39] Gregory Allen. Virtex-4VQ Dynamic and Mitigated Single Event Upset Characterization Summary. JPL Publication 09-4. 2009: 1-1.
- [40] 翁银. 基于 JPEG2000 的医学图像压缩的研究与应用[D]. 西安: 西安电子科技大学. 2008: 10-15.
- [41] Rafael C. Gonzalez, Richard E. Woods. Digital Image Processing[M], 3rd Edition. USA: Prentice Hall, 2008: 10-15.
- [42] 任杰. 基于深度学习的图像压缩方法研究[D]. 哈尔滨: 哈尔滨工业大学. 2017: 10-11.
- [43] 陈益刚. 基于改进 JPEG-LS 的图像压缩加密算法研究[D]. 海南大学. 2015: 8-18.
- [44] 王海荣. 一种易于硬件实现的 JPEG-LS 无损图像压缩算法[J]. 科技视界, 2014(17): 22-24.
- [45] 范文晶,王召利,王惠娟等. 基于 FPGA 的无损图像压缩算法实现[J]. 电子科技, 2016,29(11):126-128.
- [46] Ding JJ, Chen HH, Wei WY. Adaptive Golomb Code for Joint Geometrically Distributed Data and Its Application in Image Coding[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2013, 23(4): 661-670.
- [47] 李长兴. 基于 FPGA 实现 JPEG-LS 无损压缩算法的研究[D]. 沈阳: 沈阳航空航天大学, 2018: 13-17.
- [48] L. Macchiarulo and M. Marek-Sadowska, Wave-steering one-hot encoded FSMs[C]// Proceedings 37th Design Automation Conference USA, 2000: 357-360.
- [49] 李安龙. RTL 级故障注入工具开发与应用[D]. 哈尔滨: 哈尔滨工业大学. 2017: 8-14.

攻读硕士学位期间发表的论文及其它成果

（一）发表的学术论文

- [1] Chunhua Qi, Jianning Ma, Mingxue Huo, et al. Design of Radiation-Hardened Image Compressor Based on Lossless JPEG-LS[C]// 2019 IEEE 26th International Symposium on the Physical and Failure Analysis of Integrated Circuits. (已收录)

致 谢

两年的研究生生活转瞬即逝，入学的情形仿佛就在昨日。时间虽然短暂，但却让我学到了很多专业技能，懂得了更多人生的道理。

霍明学教授是一位专业能力强、责任心强、对待学生亲切友善的老师。机缘巧合，有幸能够师从霍老师。两年的学习生活中，感谢霍老师提供的一系列科研交流机会，让我了解专业内最新科研动态的同时，更加深了我对本专业的理解和认识。感谢您在我学习过程中，提供的建议和意见，指引着我找到正确的方向。生活中的您幽默有趣，给课题组带来不少欢笑，言传身教地教会了我们许多待人处事的道理。经济上、生活上对我们无微不至地关心和照顾，细心体贴地帮助我们解决各种困难，已经远超一名硕士导师应负之责，对此我谨记在心，复多无言不能尽吾之心！

王天琦老师对我的学习和生活提供了很大的帮助。感谢您为我们创造了许多了解学习专业内前沿知识和技能的机会，提供了一系列实践交流的平台，让我增长见识、提升了专业技能。科研工作中，您提出了许多宝贵的意见，使我的工作更加严谨，让我受益匪浅。日常生活中，遇到的诸多难题，您操心帮我们解决。两年时间不长，您却让我学到了许多工作与生活的知识。

十分感谢齐春华老师，您教会了许多专业技能。每次向您请教专业知识，您都不厌其烦地为我讲解，更是耐心细致地教会我使用多个专业软件工具。感谢老师生活中提供的帮助，为我提供了不少的便利。您亲切热情，润物无声，让我受益良多。同时感谢课题组的马国亮老师、张延清老师、刘超铭老师对我学习过程中提供的支持和帮助。您们学识渊博、视野广阔，指导我学习，让我不断进步。

同时感谢微电子中心的老师们，包括肖立伊老师、王进祥老师、来逢昌老师、付方发老师和王永生老师等微电子中心所有老师在硕士期间给予的指导和帮助。

感谢李杰师兄在我学习过程中遇到困难和疑惑时，提供的协助。感谢六年同窗朴英琿同学给予的帮助和支持。感谢课题组的师弟师妹，一起学习与进步，活跃了实验室气氛。感谢室友刚泽辉、田畅以及张警文，为我的生活带来的欢乐。

特别感谢我的家人多年对我的培养，你们是我前进道路上的动力，有了你们的支持和鼓励，我才能不断进步，谢谢你们多年的心血和付出。

最后对所有参加论文评审的各位专家、教授、老师表示由衷的感谢。