报告正文

前向纠错编码的寄存器传输级形式化验证和自动对偶综合

1. 立项依据与研究内容：

1.1. 项目的立项依据

1.1.1. 研究意义：

通讯和多媒体应用是半导体工业的主要推动力。这两个领域日新月异的发展，带来了对传输带宽永无止境的追求。这也导致每一代新的传输标准都会采用全新的编码方案，以克服在高频信号衰减方面的挑战。因此，在通讯和多媒体芯片设计项目中，最为关键且困难的工作之一是设计和验证特定的物理层编码器和解码器。

为此，在本项目负责人沈胜宇的上一个自然科学基金项目”面向通讯应用的自动对偶综合方法研究”中，我们首创了对偶综合的概念并出提出相关算法，以在寄存器传输级(RTL)，从任意编码器的源代码自动产生其对应的解码器代码[1] [2] [3] [4] [5][6]。这些工作有效的解决了 10G以太网[40]和PCI Express[41]等常见传输标准的解码器自动产生问题。

然而近两年来，随着 100G以太网[35]、128G光纤通道[36]和InfiniBand EDR[37]的出现，单通道传输带宽达到 25~32Gbps，从而导致高频衰减在标准的背板传输距离上超过了 30dB，并使其无法达到以太网标准要求的 10^{-12}误码率[43]。而工业界最新的实验性 56Gbps 串行传输技术仅能在 11 英寸以内的距离上保证 10^{-12}误码率[39]。

为了克服上述误码率问题，基于有限域(Galois field)[21]的前向纠错编码(FEC)[53]被广泛采用于 100G以太网[35]、128G光纤通道[36]和InfiniBand EDR[37]等全新的传输标准中。该纠错机制的特点及其对目前的对偶综合算法的挑战如下：

1. 前向纠错编码设计者和集成电路工程师之间在知识背景和抽象层次上的差异，导致无法很好的协作完成纠错码的集成电路实现。一方面，前向纠错编码设计者专注于有限域等抽象数学领域，使用诸如singular[28]等数学工具，在抽象数学的层面上对FEC进行推理。然而，将上述抽象的数学对象映射到集成电路的寄存器传输级描述的工作，需要由集成电路工程师完成。而后者关注的是流水线分级、布尔逻辑功能和物理时序等工程细节。这种知识背景和抽象层次上的差异，有可能在前向纠错编码(FEC)的集成电路实现上产生潜在的缺陷。因此就带来了在寄存器传输级上，对前向纠错编码(FEC)进行形式化验证和对偶综合的强烈需求。
2. 前向纠错编码(FEC)中的有限域算术操作无法使用布尔逻辑推理引擎进行高效推理。包括对偶综合在内的绝大多数形式化方法依赖于高效的布尔逻辑推理引擎，包括命题逻辑可满足求解器(SAT)[15]和二叉判决图(BDD)[14]。而在将有限域算术操作映射到布尔逻辑的过程中，会产生大量的异或操作。这极大的削弱了SAT和BDD的效率。近年来致力于验证纠错编码的多篇论文均指出了这一点[3] [7] [8] [10]。
3. 前向纠错编码(FEC)中的长帧将导致对偶综合的巨大运算开销。现有的对偶综合算法[1][2][3][4][5][6]通过逐步的扩大迁移关系的展开长度，以找到一个特定大小的移动窗口，使得该窗口内的输出序列能够唯一决定当前的输入字符。在我们使用的多个工业界标准编码器中，该窗口大小均不超过 5。然而在FEC中，为了尽量减小校验码所占用的带宽，通常会选择很长的FEC帧尺寸。比如在IEEE 802.3bj定义的 100G以太网中[42]，每个FEC帧包含 5280 个比特。在典型的 250~260 位数据路径宽度上，这将导致移动窗口的尺寸至少为 20。这超出了目前为止所有对偶综合算法的处理能力。
4. 前向纠错编码(FEC)的非对称结构和阻塞式的解码算法，导致现有的对偶综合算法无法产生规则而高效的解码器结构。正如我们将在下文中指出的，FEC 解码算法的复杂性远比编码高得多，而且并不存在线性流水线式的实现，必须在一个完整的 FEC 帧上经过多次迭代处理方能完成。这和我们现有对偶综合框架中，对解码器结构的线性流水线假设有很大区别。

应对并解决这些困难和挑战，将极大的推进 FEC 的形式化验证和对偶综合方面的研究，并进而提升面向通讯和多媒体的集成电路芯片设计质量，并缩短设计周期。同时，本项目所提出的算法，也将极大的推动与有限域相关的其他领域，如密码学的形式化方法研究。

有一个问题需要在这里特别指出并进行回答: 上述基于有限域算术的前向纠错机制(FEC)已经在学术界和工业界得到了数十年的广泛应用，为何今天才具备了对他们进行形式化验证和对偶综合的必要性？

1. 首先，以Reed-Solomon编码[22]为代表的前向纠错编码技术，从上世纪 60 年代早期提出，一直到 80 年代，仅被应用于航天和军事领域，如 1977 年发射的旅行者探测器[48]。此类应用并不需要设计大规模生产的集成电路芯片，而是使用通用计算机或者由分立元件搭建的专用电路。受限于这些物理条件，当时能够设计出来的FEC编码器和解码器并不具备很大的规模和很复杂的结构。因此并没有很强烈的进行形式化验证的需求。
2. 其次，上世纪 80 年代开始到本世纪初，消费类多媒体电子设备和存储应用的发展，如CD[49]和DVD[50]等，第一次催生了大规模生产的前向纠错编码 (FEC)芯片。不过此类应用的特点决定了其FEC的复杂性仍然不高。如CD采用的FEC结构[51]为两层嵌套的Reed-Solomon编码，有限域尺寸分别为(32,28)和(255,251)。这对应 32 个byte长度的帧尺寸和 2 个byte的纠错能力。全部的组合只有 $32*31*2^{16}$种情形，完全可以使用动态模拟完成验证。
3. 最后，本世纪初以来，随着Gbps级别高速数据通讯尤其是以太网的高速发展和广泛使用，真正复杂且需要大规模生产的前向纠错编码 (FEC)芯片开始出现。此时简单的采用动态模拟已经不足以完成验证任务[7][8]。这就对形式化验证技术产生了需求。而另一方面，相关的形式化验证技术也只有到了 2001 年以后，随着高性能的SAT求解器的出现[26]才日渐成熟。

综上所述，无论是从需求方面还是相关的使能技术方面来看，目前是研究前向纠错编码(FEC)的形式化验证和对偶综合的最佳时机。

1.1.2. 国内外研究现状及发展动态分析

本节将介绍我们在对偶综合领域的研究工作，以及国际学术界在纠错编码的形式化验证方面的相关研究工作。

对偶综合

如上所述，在通讯和多媒体芯片设计项目中，最为关键且困难的

工作之一是设计和验证特定的物理层编码器和解码器。

针对该需求，我们在发表于ICCAD’09 的论文[6]中首次提出了对偶综合算法，以在寄存器传输级上，从任意编码器的源代码自动产生其解码器。该工作成为我们后来所有其他研究成果的基础，并使得国立台湾大学的江介宏教授及其研究小组也在该领域做出了出色的研究工作[9][10][11]。两个研究小组之间产生了良性的互动和相互促进。

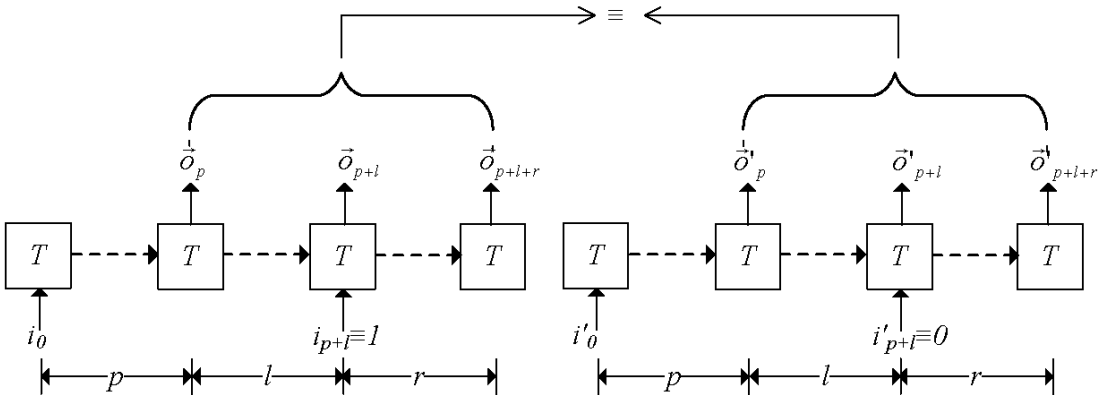

图 1 判定输入 i 能够被输出序列唯一决定

算法[6]的基本原理为：

1. 首先，该算法将编码器表示为一个有限状态机$M=<S,I,O,T>$。其中S,I和O分别是其状态变量，输入变量和输出变量集合。而$T:2^I x 2^S \rightarrow 2^O x 2^S$则是状态迁移关系，将输入字符$i \in 2^I$和当前状态$s \in 2^S$映射到输出字符$o \in 2^O$和下一状态$s' \in 2^S$。
2. 其次，如图 1 所示，该算法迭代的展开状态迁移关系T，以找到一个特定的移动窗口$\langle p,\ldots,p+l+r\rangle$，使得当前的输入字符$i_{p+l}$能够被输出序列$\langle o_p,\ldots,o_{p+l+r}\rangle$唯一决定。判定该条件的方法是将该展开的迁移关系序列、$\langle o_p,\ldots,o_{p+l+r}\rangle \equiv \langle o'_p,\ldots,o'_{p+l+r}\rangle$和$i_{p+l} \neq i'_{p+l}$三个条件的合取转换为CNF公式，并使用SAT求解器求解。当结果为不可满足时，代表i_{p+l}能够被$\langle o_p,\ldots,o_{p+l+r}\rangle$唯一决定。
3. 最后，对于i_{p+l}中的每一位，使用SAT求解器的解遍历算法，找到使其为 1 的输出序列$\langle o_p,\ldots,o_{p+l+r}\rangle$的赋值。该集合的析取即为该位的解码器函数。

在我们 2010 年发表于IEEE Transactions on CAD of Integrated Circuits and Systems的论文[3]中，针对通讯电路中大量异或操作导致论文[6]中解遍历算法效率较低的问题，提出了异或操作的推导和压缩算法，极大地提升了算法效率。

同时，国立台湾大学的Hsiou-Yuan Liu在发表于ICCAD’11 的论文[11]中，通过使用Craig插值[12]求解解遍历问题，进一步提升了算法效率。

在我们 2011 年发表于IEEE Transactions on CAD of Integrated Circuits and Systems的论文[2]中，针对算法[6]在解码器不存在的情形下不能停机的问题，通过在移动窗口中检测环形路径的方法，将解码器不存在的限界证明推广到非限界情形，从而首次得到了算法[6]的停机版本。

于此同时，国立台湾大学的Hsiou-Yuan Liu在发表于ICCAD'11的论文[11]中也提出了类似的停机算法。

在我们发表于ICCAD'11 的论文[4]中，针对用户需要手工给出断言以剔除非法输入模式的困难，我们提出了自动推导该断言的算法，极大地降低了用户的使用难度。

进一步的，在我们 2012 年发表于IEEE Transactions on CAD of Integrated Circuits and Systems的论文[1]中，我们首次发现在不同的输入模式断言下，存在完全不同的多个解码器。我们采用基于Craig插值[12]的函数依赖算法，为每个可能的解码器推导其前提条件。实验结果表明，该条件仅包含了少量的输入变量，用户可以非常容易的通过研究这些前提条件选择正确的解码器。

另外，国立台湾大学的Hsiou-Yuan Liu在 2012 年发表于IEEE Transactions on CAD of Integrated Circuits and Systems的论文[10]，首次讨论了纠错编码电路的对偶综合问题。然而由于大量异或操作导致SAT求解器运算效率降低，导致该算法只能处理一位错情形，而无法处理更复杂的多位错误情形。该缺陷也是本项目的研究动机之一。

国立台湾大学的Kuan-Hua Tu在发表于DAC'13 的论文[9]中，首次将传统对偶综合算法的限界历史依赖假设放松为非限界假设，使得对偶综合可以仅考虑编码器的可达状态集合。

上述工作有效的解决了 10G以太网[40]和PCI Express[41]等常见传输标准的解码器自动产生问题。

而本项目是上述工作的自然延伸，以解决新一代传输标准中前向纠错编码(FEC)的对偶综合问题。

纠错编码的形式化验证方法

在纠错编码中大量使用了有限域$GF(q^m)$的算术操作，其布尔逻辑实现包含大量的异或操作。而在形式化方法中常用的SAT[15]和BDD[14]推理引擎，无法高效处理此类电路[3] [7] [8] [10]。

国立台湾大学的Hsiou-Yuan Liu在 2012 年发表于IEEE Transactions on CAD of Integrated Circuits and Systems的论文[10]，首次讨论了纠错编码电路的对偶综合问题。然而由于大量异或操作导致SAT求解器运算效率降低，导致该算法只能处理最简单的一位错情形，而无法处理多位错误的情形。该缺陷也是本项目的研究动机之一。

另一个处理一位错误的工作来自于Eli Arbel发表于 2014 年ICCAD的论文[16]。该算法首先通过随机模拟找到所有可能的parity信号集合，然后使用SAT求解器从中筛选出确实满足parity要求(即任意输入翻转均能导致信号翻转)的信号子集。然后通过结构分析找到所有受到该信号保护的寄存器列表，并在这些寄存器上添加一位错误注入逻辑，最后使用传统的形式化工具验证在一位错误情形下，该信号确实能保护上述寄存器列表。

Alexey Lvov在 2012 年发表于 FMCAD的论文[8]和 2014 年发表于Formal Methods in System Design 的论文[7]首次提出了能够处理多位错的算法。该论文首先指出基于SAT和BDD的算法很难扩展到超过 24 位的纠错电路上。为了解决该问题，Alexey Lvov[8] [7]通过符号模拟，将错误校验电路的输出表示为其输入的多项式。然后将有待验证的有限域断言的反，转换为有限域多项式组的可满足问题。最后使用Buchberger[24]算法求解Gröbner basis[25]以确定该多项式组是否可解。相对于将有限域算术展开成布尔逻辑的传统方法，该算法极大的降低了运算复杂性。该方法的问题在于，当处理布尔算术和有限域算术的混合推理问题时，需要针对每一个布尔变量进行case splitting，即分别处理每个布尔变量的 0 和 1 两种情形。这带来了一个潜在的状态空间爆炸问题。而且该算法面对的问题规模也相对较小，帧长度只有 1024 位。而在 100G以太网FEC中采用的帧长度为 5280 位。

上述相关研究指出，面向新型传输标准的 FEC 形式化验证和对偶综合，是具有重大意义的新颖研究课题。

1.1.3. 参考文献

[1]. ShengYu Shen, Ying Qin, Kefei Wang, Zhengbin Pang, Jianmin Zhang, Sikun Li: Inferring Assertion for Complementary Synthesis. IEEE Trans. on CAD of Integrated Circuits and Systems 31(8): 1288-1292 (2012)

[2]. ShengYu Shen, Ying Qin, Liquan Xiao, Kefei Wang, Jianmin Zhang, Sikun Li: A Halting Algorithm to Determine the Existence of the Decoder. IEEE Trans. on CAD of Integrated Circuits and Systems 30(10): 1556-1563 (2011)

[3]. ShengYu Shen, Ying Qin, Kefei Wang, Liquan Xiao, Jianmin Zhang, Sikun Li: Synthesizing Complementary Circuits Automatically. IEEE Trans. on CAD of Integrated Circuits and Systems 29(8): 1191-1202 (2010)

[4]. ShengYu Shen, Ying Qin, Jianmin Zhang: Inferring assertion for complementary synthesis. ICCAD 2011: 404-411

[5]. ShengYu Shen, Ying Qin, Jianmin Zhang, Sikun Li: A halting algorithm to determine the existence of decoder. FMCAD 2010: 91-99
[6]. ShengYu Shen, Jianmin Zhang, Ying Qin, Sikun Li: Synthesizing complementary circuits automatically. ICCAD 2009: 381-388
[7]. Alexey Lvov, Luis Alfonso Lastras-Montaño, Barry M. Trager, Viresh Paruthi, Robert Shadowen, Ali El-Zein: Verification of Galois field based circuits by formal reasoning based on computational algebraic geometry. Formal Methods in System Design 45(2): 189-212 (2014)
[8]. Alexey Lvov, Luis Alfonso Lastras-Montaño, Viresh Paruthi, Robert Shadowen, Ali El-Zein: Formal verification of error correcting circuits using computational algebraic geometry. FMCAD 2012: 141-148
[9]. Kuan-Hua Tu, Jie-Hong R. Jiang: Synthesis of feedback decoders for initialized encoders. DAC 2013: 49
[10]. Hsiou-Yuan Liu, Yen-Cheng Chou, Chen-Hsuan Lin, Jie-Hong R. Jiang: Automatic Decoder Synthesis: Methods and Case Studies. IEEE Trans. on CAD of Integrated Circuits and Systems 31(9): 1319-1331 (2012)
[11]. Hsiou-Yuan Liu, Yen-Cheng Chou, Chen-Hsuan Lin, Jie-Hong R. Jiang: Towards completely automatic decoder synthesis. ICCAD 2011: 389-395
[12]. Kenneth L. McMillan: Interpolation and SAT-Based Model Checking. CAV 2003: 1-13
[13]. Ron Roth. Introduction to Coding Theory. Cambridge University Press,2007.
[14]. Randal E. Bryant: Graph-Based Algorithms for Boolean Function Manipulation. IEEE Trans. Computers 35(8): 677-691 (1986)
[15]. Matthew W. Moskewicz, Conor F. Madigan, Ying Zhao, Lintao Zhang, Sharad Malik: Chaff: Engineering an Efficient SAT Solver. 530-535
[16]. Eli Arbel, Shlomit Koyfman, Prabhakar Kudva, Shiri Moran: Automated detection and verification of parity-protected memory elements. ICCAD 2014: 1-8
[17]. Alexander Zeh, Christian Senger: A link between Guruswami-Sudan's list-decoding and decoding of interleaved Reed-Solomon codes. ISIT 2010: 1198-1202
[18]. V. Guruswami and M. Sudan, “Improved Decoding of Reed–Solomon Codes and Algebraic Geometry Codes,” IEEE Trans. Inform. Theory, vol. 45, no. 6, pp. 1757–1767, September 1999.
[19]. R. Kötter, On Algebraic Decoding of Algebraic-Geometric and Cyclic Codes, Linkoping Studies in Science and Technology, no. 419 (Ph.D. Dissertation, Department of Electrical Engineering),

Linkoping U., 1996.
[20]. R. Roth and G. Ruckenstein, “Efficient Decoding of Reed–Solomon Codes be- yond Half the Minimum Distance,” IEEE Trans. Inform. Theory, vol. 46, no. 1, pp. 246–257, January 2000.
[21]. Mullen, Gary L.; Panario, Daniel (2013), Handbook of Finite Fields, CRC Press, ISBN 978-1-4398-7378-6
[22]. Reed, Irving S.; Solomon, Gustave (1960), "Polynomial Codes over Certain Finite Fields", Journal of the Society for Industrial and Applied Mathematics (SIAM) 8 (2): 300–304, doi:10.1137/0108018
[23]. Bose, R. C.; Ray-Chaudhuri, D. K. (March 1960), "On A Class of Error Correcting Binary Group Codes", Information and Control 3 (1): 68–79, doi:10.1016/s0019-9958(60)90287-4, ISSN 0890-5401
[24]. Buchberger, B. (August 1976). "Theoretical Basis for the Reduction of Polynomials to Canonical Forms". ACM SIGSAM Bull. (ACM) 10 (3): 19–29. doi:10.1145/1088216.1088219. MR 0463136.
[25]. William W. Adams, Philippe Loustaunau (1994). An Introduction to Gröbner Bases. American Mathematical Society, Graduate Studies in Mathematics, Volume 3. ISBN 0-8218-3804-0
[26]. Matthew W. Moskewicz, Conor F. Madigan, Ying Zhao, Lintao Zhang, Sharad Malik: Chaff: Engineering an Efficient SAT Solver. DAC 2001: 530-535
[27]. Dilip V. Sarwate, Zhiyuan Yan: Modified Euclidean algorithms for decoding Reed-Solomon codes. ISIT 2009: 1398-1402
[28]. Christoph Lossen: Singular: A Computer Algebra System CISE, 5 (4), 45-55, 2003.
[29]. Jeong-In Park, Hanho Lee, Seongsoo Lee: An area-efficient truncated inversionless Berlekamp-Massey architecture for Reed-Solomon decoders. ISCAS 2011: 2693-2696
[30]. Venkatesan Guruswami, Chaoping Xing:Optimal rate algebraic list decoding using narrow ray class fields. J. Comb. Theory, Ser. A 129: 160-183 (2015)
[31]. Niklas Eén, Niklas Sörensson:Temporal induction by incremental SAT solving. Electr. Notes Theor. Comput. Sci. 89(4): 543-560 (2003)
[32]. Ruben Martins, Saurabh Joshi, Vasco M. Manquinho, Inês Lynce:Incremental Cardinality Constraints for MaxSAT. CP 2014: 531-548
[33]. Shamim Ripon, Alice Miller:Verification of Symmetry Detection using PVS. ECEASST 35 (2010)
[34]. Forney, Jr., G. (October 1965), "On Decoding BCH Codes", IEEE Transactions on Information Theory 11 (4): 549–557, doi:10.1109/TIT.1965.1053825, ISSN 0018-9448
[35]. www.ieee802.org/3/bj

[36]. http://en.wikipedia.org/wiki/Fibre_Channel
[37]. http://en.wikipedia.org/wiki/InfiniBand
[38]. “Baseline Proposal for 100G Backplane Specification using PAM2”.http://www.ieee802.org/3/bj/public/mar12/dudek_01a_0312.pdf
[39]. http://media.wix.com/ugd/720847_e2386823f3ab4d09aba4a4c2d144a6e7.pdf
[40]. IEEE. 2012. IEEE Standard for Ethernet SECTION FOURTH. (2012). Retrieved January 25, 2013 from http://standards.ieee.org/getieee802/download/802.3-2012_section4.pdf
[41]. PCI-SIG. 2009. PCI Express Base 2.1 Specification. (2009). Retrieved January 25, 2013 from http://www.pcisig.com/members/downloads/specifications/pciexpress/PCI Express Base r2 1 04Mar09.pdf
[42]. “Backplane NRZ FEC Baseline Proposal ”. http://www.ieee802.org/3/bj/public/mar12/gustlin_01_0312.pdf
[43]. “Should the FEC be Optional for the NRZ PHY?”. http://www.ieee802.org/3/bj/public/mar12/meghelli_01a_0312.pdf
[44]. “Alignment Marker Lock State Machine for NRZ 100G-KR”. http://www.ieee802.org/3/bj/public/may12/wang_01_0512.pdf
[45]. “256b/257b Transcoding for 100 Gb/s Backplane and Copper Cable “ . http://www.ieee802.org/3/bj/public/mar12/cideciyan_01a_0312.pdf
[46]. “ Transmitter and Receiver Architecture Without Self-Synchronizing Rx Scrambler ” . http://www.ieee802.org/3/bj/public/may12/cideciyan_02_0512.pdf
[47]. “Scrambling scheme and FEC bits performance”. http://www.ieee802.org/3/bj/public/jul12/anslow_01a_0712.pdf
[48]. http://en.wikipedia.org/wiki/Voyager_program
[49]. http://en.wikipedia.org/wiki/Compact_disc
[50]. http://en.wikipedia.org/wiki/DVD
[51]. http://en.wikipedia.org/wiki/Reed–Solomon_error_correction
[52]. http://en.wikipedia.org/wiki/Buchberger%27s_algorithm
[53]. http://en.wikipedia.org/wiki/Forward_error_correction

1.2. 项目的研究内容、研究目标，以及拟解决的关键科学问题

1.2.1. 研究目标

本项目的目标为，以我们在对偶综合领域的工作为基础，研究全新的理论，算法和工具，以在寄存器传输级上，解决前向纠错机制的形式化验证和自动对偶综合问题。

1.2.2. 背景知识

有限域代数、Gröbner basis 和 Buchberger 算法

群$(G, \cdot)$是指集合G及其上的二元算子$\cdot$，满足条件**(1)**闭包：对于任意$a,b \in G$，有$a \cdot b \in G$；**(2)**结合律：对于任意$a,b,c \in G$，有$a \cdot (b \cdot c) \equiv (a \cdot b) \cdot c$；**(3)**幺元：存在$1 \in G$，使得对任意$a \in G$，有$1 \cdot a \equiv a \cdot 1 \equiv a$；**(4)**反元：对于任意$a \in G$，存在$a^{-1} \in G$，使得$a \cdot a^{-1} \equiv a^{-1} \cdot a \equiv 1$。

交换群是指对于任意$a,b \in G$，有$a \cdot b \equiv b \cdot a$。记$a^n = a \cdot a \ldots \cdot a$，即$n$个$a$。**循环群**$G$是指存在$a \in G$，使得$G$的任意元素均能表示$a^m$，其中$m$为非负整数。对于最小的$n$，使得$a^n \equiv 1$，称$n$为$a$的order。

环$(R,*,+)$是指在非空集合R，及其上的两个二元算子$*$和$+$，满足条件**(1)**$(R,+)$是交换群；**(2)**$*$满足结合律；**(3)**分配律：对任意$a,b,c \in R$，$a*(b+c) \equiv (a*b)+(a*c)$且$(b+c)*a \equiv b*a+c*a$。

其中$+$在群$(R,+)$上的幺元也称为$(R,*,+)$的**零元**，记为0。任意$x \in R$在$(R,+)$上的反元记为$-x$。若一个环$(R,*,+)$使得$*$满足交换律，且有幺元1，则称为**整环**。

域：当一个整环$(R,*,+)$使得对任意$a \in R$在$*$上存在反元a^{-1}，则称为域。**有限域**$GF(p)$为$(P,*_p,+_p)$，其中p为质数，$P=\{0,1,\ldots,p-1\}$，$a+_p b=(a+_Z b) \bmod p$，$a*_p b=(a*_Z b) \bmod p$。其中$+_Z$和$*_Z$分别是正整数集合Z上的加法和乘法。通过特定的操作可以将$GF(p)$扩展为**扩展域**$GF(p^m)$，为简明起见，不再赘述。

多项式：对于有限域F，多项式如$a(x)=\sum_{i=0}^{n} a_i x^i$所示，其中$a_i \in F$，则$deg\ a=n$。而每个$a_i x^i$称为一个单项式。$F$上的所有此类多项式集合记为$F[x]$。定义$(F[x],*_{F[x]},+_{F[x]})$，其中$a(x)+_{F[x]} b(x)=\sum_{i=0}^{n}(a_i +_F b_b)x^i$，而$a(x)*_{F[x]} b(x)=\sum_{i=0}^{2n} c_i x^i$，其中$c_i=\sum_{j=0}^{i} a_j b_{i-j}$。很明显$(F[x],*_{F[x]},+_{F[x]})$是整环。

上述在一个变量x上的多项式可以扩展为在多个变量$\{x_1,\ldots,x_n\}$上的多项式。此时所有此类多项式的集合记为$F[x_1,\ldots,x_n]$。

Ideal $I\subseteq F[x_1,\ldots,x_n]$满足**(1)**$0\in I$；**(2)**对任意$f,g\in I$，有$f+g\in I$；**(3)**面对任意$f\in I$和$h\in F[x_1,\ldots,x_n]$，有$hf\in I$。对于$f_1,\ldots,f_s\in F[x_1,\ldots,x_n]$，很明显$\langle f_1,\ldots,f_s\rangle=\left\{\sum_{i=1}^{s}h_i f_i : h_1,\ldots,h_s\in F[x_1,\ldots,x_n]\right\}$是一个ideal。

Hilbert Basis定理：$F[x_1,\ldots,x_n]$的任何ideal均可以表示为有限个$f_1,\ldots,f_s\in F[x_1,\ldots,x_n]$组成的$\langle f_1,\ldots,f_s\rangle$。$f_1,\ldots,f_s$称为该ideal的基(Basis)。

然而，对于特定的$f\in F[x_1,\ldots,x_n]$，有可能可以分解成为$f_1,\ldots,f_s$上的两个不同组合：$f\equiv\sum_{i=1}^{s}a_i f_i\equiv\sum_{i=1}^{s}b_i f_i$。而Gröbner basis则能保证这种分解是唯一的。而从任意$f_1,\ldots,f_s$中产生Gröbner basis的算法称为**Buchberger算法**[24]。该算法的粗略介绍如下：

首先给出一个单项式的排序规则，如按照$x_1,\ldots,x_n$顺序的字典序规则，以及按照degree排序的规则。以$F[x]$为例，任意多项式可以表示为$f\equiv\sum_{i=1}^{s}a_i x^i$。则$LT(f)=a_s x^s$，$LM(f)=x^s$，$LE(f)=s$。

对于两个多项式f和g，定义$LM(f)$和$LM(f)$的最小公倍数为$\gamma=max(LE(f),LE(g))$。而$S(f,g)=\frac{x^\gamma}{LT(f)}f-\frac{x^\gamma}{LT(g)}g$。很明显，$S(f,g)$将消去$f$和$g$的最高阶单项式。而对于基$G$，$\overline{S(f,g)}^{G}$是$S(f,g)$在$G$上的余数。则**Buchberger算法**[24]的流程如下所示。

Input: $F=(f_1,\ldots,f_s)$
Output: a Groebner basis $G=(g_1,\ldots,g_t)$ for I, with $F\subset G$

$G:=F$
REPEAT
  $G':=G$
  FOR each pair $\{p,q\}$, $p\neq q$ in G' DO
    $S:=\overline{S(p,q)}^{G'}$
    IF $S\neq 0$ THEN $G:=G\cup\{S\}$
UNTIL $G=G'$

图 2 Buchberger 算法

典型的 FEC：Reed-Solomon 编码

在讨论 FEC 的概念时，我们需要以下的系统架构模型：

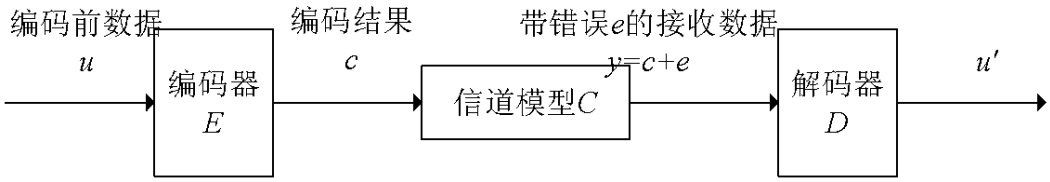

图 3 系统架构模型

Reed-Solomon编码是广泛应用于数据存储和通讯领域的前向纠错编码。如上图所示，一个纠错编码*[n,k,d]*的原理是，在编码器*E*编码过程中，将每个长度为*k*的输入$u=(u_0,\ldots,u_{k-1})$，映射至长度为*n*的输出$c=(c_0,\ldots,c_{n-1})$，使得任意两个*c*和*c'*的汉密尔顿距离至少为$d=n-k+1$。在传输和存储过程中，假设在信道模型*C*上有不多于 $(d-1)/2$ 个符号被损坏，同时假设加性噪声为$e=(e_0,\ldots,e_{n-1})$，则使得$|e_i|\neq 0$ 的*i*的个数不超过 $(d-1)/2$。此时通过信道得到的接收数据$y=c+e$。则在解码器*D*上解码时，可以通过多项式插值从*y*中恢复出正确的原始数据*u*。

其中，*n* 通常由用户愿意付出的编码空间和时间复杂性决定。其中空间复杂性是指硬件实现的电路面积或者软件实现需要的存储开销。而时间复杂性是指用于解码的时间延迟，通常是串行传输延迟的2~3 倍。*n-k+1* 或者 *d* 通常由通讯信道与存储介质的原始误码率 *p* 和用户期望达到的误码率 *p'*决定。该值越大则能得到的误码率提升 *p'/p* 就越高。

为了兼容尽量多的Reed-Solomon编码的变种，我们使用文献[13]的一般性记法(Generalized Reed-Solomon，简称为GRS)。

对于有限域*F*，假设$a_1,\ldots,a_n$为*F*中不同的非 0 元素，而$v_1,\ldots,v_n$和$v'_1,\ldots,v'_n$为*F*中非 0 元素(不要求不同)。在*F*上的GRS码是一个线性*[n,k,d]*码C_{GRS}。其产生矩阵为：

$$G_{\mathrm{GRS}} = \begin{pmatrix} 1 & 1 & \dots & 1 \\ \alpha_1 & \alpha_2 & \dots & \alpha_n \\ \alpha_1^2 & \alpha_2^2 & \dots & \alpha_n^2 \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_1^{k-1} & \alpha_2^{k-1} & \dots & \alpha_n^{k-1} \end{pmatrix} \begin{pmatrix} v'_1 & & & \\ & v'_2 & & 0 \\ & & \ddots & \\ 0 & & & v'_n \end{pmatrix}$$	(1)

对于输入$u=(u_0,\ldots,u_{k-1})$，其编码结果$c=(c_0,\ldots,c_{n-1})$为

$$c=uG_{GRS}$$	(2)

假设*F*上的*k*阶多项式记为$F_k[x]$。则可以将*u*视为一个*F*上的*k*阶多项式$u(x)=u_0+u_ix+\ldots+u_{k-1}x^{k-1}\in F_k[x]$。则$C_{GRS}$的编码可记为：

$$c=uG_{GRS}=(v_1'u(a_1),v_2'u(a_2),\ldots,v_n'u(a_n))$$	(3)

对于上述C_{GRS}，其对应的校验矩阵H_{GRS}记为：

$$H_{\mathrm{GRS}} = \begin{pmatrix} 1 & 1 & \dots & 1 \\ \alpha_1 & \alpha_2 & \dots & \alpha_n \\ \alpha_1^2 & \alpha_2^2 & \dots & \alpha_n^2 \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_1^{d-2} & \alpha_2^{d-2} & \dots & \alpha_n^{d-2} \end{pmatrix} \begin{pmatrix} v_1 & & & \\ & v_2 & & 0 \\ & & \ddots & \\ 0 & & & v_n \end{pmatrix} \tag{4}$$

(1)-(4)满足如下关系：

$$H_{GRS}G_{GRS}^{T} = 0 \tag{5}$$

由该关系，在已知G_{GRS}，即$a_1,\dots,a_n$和$v'_1,\dots,v'_n$的前提下，可以求出未知数$v_1,\dots,v_n$，从而得到校验矩阵H_{GRS}。反之也可以从H_{GRS}得到产生矩阵G_{GRS}。

假设c经过噪声信道后，受到加性噪声$e=(e_1,\dots,e_n)$的干扰，则解码器的输入$y=(y_1,\dots,y_n)$为：

$$y=c+e \tag{6}$$

由(5)和(6)可知

$$H_{GRS}y^{T} = H_{GRS}e^{T} \tag{7}$$

假设 J 为错误位置集合，我们有：

$$e_k \neq 0 \quad \Leftrightarrow \quad k \in J \tag{8}$$

很明显$|J| \leq (d-1)/2$。

C_{GRS}的解码过程要比其编码过程复杂得多，一般分为以下几步：

第一步，计算 Syndrome 如下：

$$\begin{pmatrix} S_0 \\ S_1 \\ \vdots \\ S_{d-2} \end{pmatrix} = H_{\mathrm{GRS}}\mathbf{y}^{T} . \tag{9}$$

若将其视为 F 上的多项式，则为：

$$S(x) = \sum_{\ell=0}^{d-2} S_\ell x^\ell . \tag{10}$$

由(7)和(10)可知

$$S_\ell = \sum_{j \in J} e_j v_j \alpha_j^\ell , \quad \ell = 0, 1, \dots, d-2 \tag{11}$$

将(11)代入(10)有:

$$S(x)=\sum_{\ell=0}^{d-2}x^{\ell}\sum_{j\in J}e_jv_j\alpha_j^{\ell}=\sum_{j\in J}e_jv_j\sum_{\ell=0}^{d-2}(\alpha_jx)^{\ell}$$	(12)

考虑环$F[x]/x^{d-1}$，即F上多项式针对x^{d-1}的模组成的环。在该环中所有不能被x整除的元素在该环的乘性算子下组成一个群。多项式$1-a_jx$在该群中的反元为$\sum_{\ell=0}^{d-2}(\alpha_jx)^{\ell}$，因为:

$$(1-\alpha_jx)\sum_{\ell=0}^{d-2}(\alpha_jx)^{\ell}=1-(\alpha_jx)^{d-1}\equiv 1 \pmod{x^{d-1}}$$	(13)

因此(12)可以改写为:

$$S(x)\equiv\sum_{j\in J}\frac{e_jv_j}{1-\alpha_jx} \pmod{x^{d-1}}$$	(14)

收集(14)的所有分母有:

$$\Lambda(x)=\prod_{j\in J}(1-\alpha_jx)$$	(15)

将(15)的右侧乘上(14)的右侧，得到:

$$\Gamma(x)=\sum_{j\in J}e_jv_j\prod_{m\in J\setminus\{j\}}(1-\alpha_mx)$$	(16)

因此我们有以下三个约束:

$gcd(\Lambda(x),\Gamma(x))=1$	(17)
$deg\,\Gamma < deg\,\Lambda \le (d-1)/2$	(18)
$\Lambda(x)S(x)\equiv\Gamma(x) \mod x^{d-1}$	(19)

第二步，(17)-(19)可以使用参数$a(x)=x^{d-1}$和$b(x)=S(x)$调用以下经典的Euclid算法[27]进行求解:

$$
\begin{aligned}
&r_{-1}(x)\leftarrow a(x);\ r_0(x)\leftarrow b(x);\\
&s_{-1}(x)\leftarrow 1;\ s_0(x)\leftarrow 0;\\
&t_{-1}(x)\leftarrow 0;\ t_0(x)\leftarrow 1;\\
&\text{for } (i\leftarrow 1;\ r_{i-1}(x)\neq 0;\ i{+}{+})\ \{\\
&\quad q_i(x)\leftarrow r_{i-2}(x)\ \text{div}\ r_{i-1}(x);\\
&\quad r_i(x)\leftarrow r_{i-2}(x)-q_i(x)r_{i-1}(x);\\
&\quad s_i(x)\leftarrow s_{i-2}(x)-q_i(x)s_{i-1}(x);\\
&\quad t_i(x)\leftarrow t_{i-2}(x)-q_i(x)t_{i-1}(x);\\
&\}
\end{aligned}
$$

正如[52]所指出的，该Euclid算法是上述Buchberger算法在一元多项式环上的特例。最终的解为

$\Lambda(x)=t_h(x)$	(20)
$\Gamma(x)=r_h(x)$	(21)

其中h是最小的i使得$r_i < (d-1)/2$。

第三步，错误位置和错误值使用Forney算法[34]得出：

$e_j = \begin{cases} -\frac{\alpha_j}{v_j} \cdot \frac{\Gamma(\alpha_j^{-1})}{\Lambda'(\alpha_j^{-1})} & \text{if } \Lambda(\alpha_j^{-1}) = 0 \\ 0 & \text{otherwise} \end{cases}, \quad j = 1, 2, \ldots, n\ .$	(22)

从上述的第一步到第三步可知，Reed-Solomon码的解码是阻塞式的，即需要在每一步保存完整的$y=(y_1,\ldots,y_n)$，并在其上进行多轮迭代。假设y在信道上的传输时间是T，则相对于非纠错的情形，上述三步需要额外的 $2T$时间。如果还要标记错误个数超出 $(d-1)/2$ 的情形的话，还需要另一个额外的T。根据我们在新一代天河超级计算机项目中的经验，这将带来大约 150 纳秒的额外延时。

Reed-Solomon 及其变种是一个非常广大的研究领域，我们不可能在这里给出完整的描述。上述算法仅仅是为了给本申请书提供一个简单的入门式介绍。该介绍无论是在一般性还是在特殊性方面都是不完备的。为此，我们在下面对该领域内的其他相关工作给出简单介绍。

首先，在实际应用中存在Reed-Solomon编码的多种变种，每一种都与上面的"标准"描述有或多或少的差别。比如，在公式(1)中的a_i和v_i有多种可能的选择，如当v_i全部为 1 时，称为normalized RS码，当$a_i=v_i$时称为狭义RS码。当u和c不像(2)(3)那样被视为多项式参数，而是被视为特定多项式在F上的取值时，则能得到systematic RS码，使得u成为c的一部分。

其次，还存在其他的高效求解(17)-(19)的方法，如Berlekamp-Massey算法[13]。另外，Guruswami-Sudan算法[18]则将纠错能力扩展到了半径为$(d-1)/2$ 的圆之外，从而提升了纠错效率。Kötter[19]和 Roth-Ruckenstein[20]则进一步改善了该算法的复杂性。

1.2.3. FEC 的形式化验证和对偶综合算法框架

在介绍具体的研究内容之前，我们首先需要介绍用于形式化验证和对偶综合的模型和流程。

假设我们的工具将使用如下所示的系统结构：

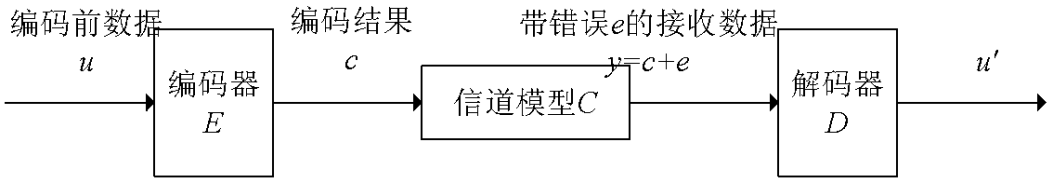

图 4 系统架构模型

其中编码器E使用常见的寄存器传输级硬件描述语言Verilog或VHDL描述。我们使用现有的EDA工具，如Synopsys的Design Compiler将其展开为门级网表形式，以便使用现有的语法分析器[6]得到所需的状态迁移关系T_E。T_E将被用于图 1 中的对偶综合。和现有对偶综合不同的是，我们需要在该过程中保留所有的有限域算术操作，以便使用singular[28]进行相应的有限域运算。同时，假设编码器需要n周期才能完成一个FEC帧的处理，因此我们还需要借助类似[8][7]的符号模拟算法，将T_E展开n次，以得到如公式(2)所示的从编码前输入u到编码结果c的映射$c=uG_{GRS}$。

对于解码器D，在进行形式化验证的情形下，我们假设已经得到了D的Verilog或者VHDL描述，我们将采用类似于E的处理方法，以得到如公式(4)所示的H_{GRS}，以及如公式(22)所示的错误修正。这些将被用于验证最后经过错误修正的输出u'是否等于u。而在进行对偶综合的情形下，T_D将由对偶综合算法产生，并转换成为Verilog或者VHDL描述。

信道模型 C 将负责产生所有可能的出错组合，包括在长度为 n 的向量 c 上产生最多 $(d-1)/2$ 个错误，以及每个错误位置上所有可能的错误值。

对于**形式化验证流程**，其算法流程如下:

1. 通过符号模拟，将恢复的原始数据 u’表示为原始数据 u 和所有可能的错误组合 e 的多项式。
2. 将 $u\neq u$’表示展开为多项式组。
3. 使用Singular工具[28]提取相应的Gröbner basis[25]，以确认该多项式组是否可解。

对于**对偶综合流程**，其算法流程类似于传统的对偶综合，主要区别在于所使用的推理引擎不再是纯粹的布尔逻辑推理引擎，而是混合了布尔逻辑和有限域算术的混合推理引擎。同时，传统对偶综合中的编码器将被图 4 中的 $E+C$ 代替，以便将出错情形考虑在内。

1.2.4. 研究内容

基于上述的模型和流程，以及 FEC 对现有形式化方法提出的挑

战，我们将研究以下内容:

第一个研究内容为：高效的有限域算术和布尔逻辑混合推理。如前所述,在将前向纠错机制中的有限域算术映射到布尔逻辑的过程中，会产生大量的异或操作。这将极大的削弱SAT和BDD的效率[8]。Alexey Lvov [8][7]首次提出了有限域算术和布尔逻辑的混合推理算法。然而该算法使用的case splitting导致了状态空间爆炸问题。因此，高效的有限域算术和布尔逻辑混合推理算法是提高整体算法效率的关键。

第二个研究内容为：面向有限域的对称性削减算法。如前所述，为了尽量减小校验码所占用的带宽，通常会为编码器的编码结果c选择很大的帧尺寸，其中包含数百个有限域元素。从语义角度看，一个FEC帧$c=(c_0,\ldots,c_{n-1})$内部的各个元素c_i之间存在一定的对称性；然而从语法角度看，公式(2)中指出c中的各个元素c_i是不对称的，需要处理的组合个数是$C_n^{\frac{d-1}{2}}2^{t\frac{d-1}{2}}$，其中$t$是每个$c_i$占用的比特数。因此，弥合两者的差异将极大地减少需要处理的组合数量。

第三个研究内容为：基于模板的对偶综合算法。从公式(17)-(19)的求解过程可知，FEC解码器从原理上来说是阻塞式的，需要在y上进行多次迭代。如果任由推理引擎自由搜索整个状态空间，不仅时间开销很大，而且会产生非常不规则和高冗余的解码器结构。而另一方面，现有的研究指出了多种高度规则和低复杂性的解码器结构，如Euclid算法[27]、Berlekamp-Massey算法[13]和Guruswami-Sudan算法[18]等。因此，我们将研究如何从上述结构中提取参数化的模板，然后将解码器的综合问题变为参数空间的搜索问题。

我们将在以下各个小节中给出对这些问题的预期解决方案。

1.2.5. 拟解决的关键科学问题

针对上述研究内容，我们拟解决下述相应的关键科学问题:

第一个关键科学问题是：增量式低开销的 Gröbner basis 提取算法。

如上所述，提取Gröbner basis[25]是在有限域上进行形式化推理的关键步骤。由于有待验证的FEC结构非常复杂，产生的多项式组的尺

寸将会非常巨大。这将导致每次提取Gröbner basis的运算时间开销也将会非常巨大。

而在由图 4 所示的信道模型中，生成的多个错误位置组合和错误值之间，存在高度的重叠和相似性。以 100G以太网[42]为例，对长度为$n=528$ 的$c=(c_0,\ldots,c_{527})$，可能的出错位置最多为 7 个。则两个出错位置组合{0,..,5,6}和{0,..,5,7}有很大的重叠。同时，对于同一个出错位置组合，不同的错误值之间也有很大的重叠。

这种重叠提供了这样一种可能性，即从一种情形的 Gröbner basis 中，增量式的产生另一种情形的 Gröbner basis，而仅仅引入很小的计算开销。

第二个关键科学问题是：基于 cyclic code 的对称性削减问题。

如上所述，所有可能的错误组合的个数为$C_n^{\frac{d-1}{2}}2^{t\frac{d-1}{2}}$，这对于形式化验证工具来说是一个非常大的搜索空间。需要使用合适的对称性削减技术来减少可能的组合个数。

公式(1)展示的是一般性的Reed Solomon编码的产生矩阵。而日常使用的Reed-Solomon编码是cyclic的，即任意c的右移和左移版本都仍然是合法的编码结果。同时Reed-Solomon编码在公式(1)中使用$a_i=a^{i-1}$和$v_i=a^{b(i-1)}$，其中a是有限域$GF(q^m)$的generator。

这就带来了削减对称性的可能。对于两个错误向量 e 和 e'，当 e' 是 e 的左移或者右移版本时，在计算他们的 Gröbner basis 时，也存在一定的对称关系。这种对称关系可以用于快速的从 e 的 Gröbner basis 中计算出 e'的 Gröbner basis。

第三个关键科学问题是：解码算法的模板结构设计和参数提取

常用和高效的解码算法，如Euclid算法[27]、Berlekamp-Massey算法[13]和Guruswami-Sudan算法[18]，他们中的每一个都不是完整定义的算法，而仅仅代表了一大类算法的共同核心特点。在每一大类内部，还有很多不同的变种。如Berlekamp-Massey算法[13]提出于上世纪 60 年代，但是近年仍然有新的变种出现，如reformulated inversionless Berlekamp-Massey[29]。而Guruswami-Sudan算法[18]作为近年出现的全新研究方向，也在不断的产生新的研究成果[30]。

因此，如何为每一类算法，设计一个合适的模板，以涵盖各种可能的情形，将是我们需要重点解决的一个关键问题。而另一方面，过于一般化的模板，有可能带来求解复杂性的提升，甚至有可能导致不可解。因此，需要在通用性和求解复杂性之间取得良好折中。

1.3. 拟采取的研究方案及可行性分析

我们将按照下图所示的流程安排本项目的研究方案:

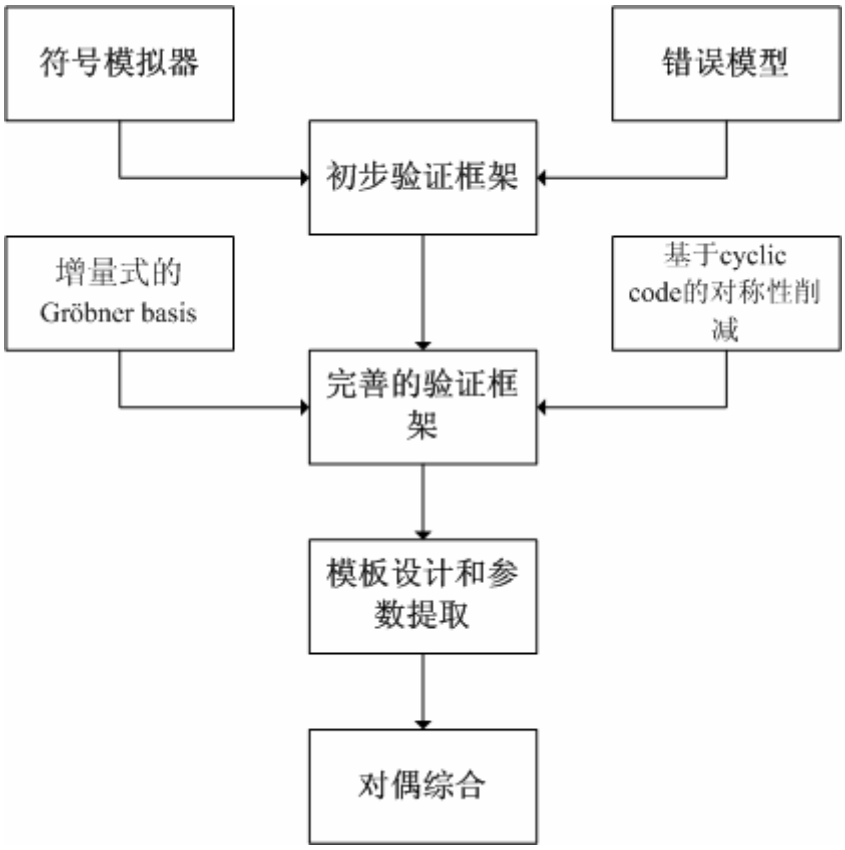

图 4 项目流程

各个关键技术的详细描述如下:

1.3.1. 面向有限域算术操作的符号模拟器

该模拟器的目的在于从 FEC 的 Verilog 或 VHDL 源代码，产生其对应的网表。该网表可以视为一个有向无环图 $G=<V,E>$。其中 V 是节点列表，每种节点完成不同的逻辑功能，包括布尔逻辑中的与门 AND 和反向器 INV。此外，和传统的纯布尔逻辑网表不同，V 中还包含主要的有限域算术操作，如加法、乘法、求反、平方、平方根、标准基上的投影和重映射等。

纯布尔逻辑网表的产生可以直接使用现有的商用 EDA 工具，如 Synopsys 的 Design Compiler。而对于有限域算术操作，在源代码中通常会使用单独的模块完成。我们将在使用 Design Compiler 时指明这些模块不能被展平。这样最终产生的网表中将明确的包含对有限域算术操作的引用。

进一步的，通过读入上述产生的网表，并从输出开始，向输入进行回溯。如果在回溯过程中遇到状态单元，则递归的对他们的状态迁移函数进行回溯。在回溯过程中收集被遍历到的节点集合，以及他们之间的连接关系。如此即可将 FEC 的输出表示为输入的有限域多项式。

1.3.2. 信道错误模型的符号表示

如图 4 所示的系统架构可以被视为一个完整的网表，其中分别包含编码器 *E* 和解码器 *D* 的两个网表。而解码器的输入为 *y*，而编码器的输出为 *c*。他们之间的关系由 $y=c+e$ 表示。

n 个 *e* 中最多有 $(d-1)/2$ 个不为 0，该约束可以表示为额外的 CNF 公式，并和上述网表进行合取。如此即可涵盖所有可能的错误组合。

1.3.3. 初步的验证框架

将上述图 3 所示的网表、错误模型约束和$u \neq u'$转变成有限域多项式组，并使用singular[28]计算其Gröbner basis[25]，以确定其是否可解。

原理上说，该算法能够解决 FEC 的形式化验证问题。然而正如前面指出的，该算法存在状态空间爆炸问题。需要进行以下的研究，以改善算法复杂性。

1.3.4. 增量式的 Gröbner basis 产生算法

有限域代数工具，如singular[28]中，通常使用如图 2 所示的Buchberger[24]算法求解Gröbner basis[25]。而该算法并不考虑多个相似的ideal $\langle f_1,\ldots,f_s\rangle$和$\langle g_1,\ldots,g_s\rangle$之间的相似性。

为了将这种相似性考虑进来，我们将参考近年来在增量式SAT[31]求解方面的研究进展，开发能增量式产生Gröbner basis的算法。

同时，在错误模型中的”最多不超过 $(d-1)/2$ 个错误”这样的约束是很明显的Cardinality约束。这类问题的求解在MaxSAT[32]中有很长的研究历史和很丰富的成果积累。我们也将借鉴这些研究成果改进我们的算法。

最后，Cardinality 约束和抽象代数中的 degree，rank 和 dimension 等概念有着内在的联系。这将直接决定特定多项式组的可解性。而特定约束集合的不可解是对偶综合中的重要组成部分。我们也将探索他

们之间的这种关系能否用于加速对偶综合算法。

1.3.5. 基于 cyclic code 的对称性削减算法

由于 cyclic code 本身所具有的特殊代数结构，使得其编解码的算法的复杂性和硬件实现开销大大降低。所以 cyclic code 成为 Reed-Solomon 的标准实现。

同时，该特性也为削减我们算法中的对称性提供了可能。首先，针对图 4 中的y和u，需要针对他们的每一个子元素y_i和u_i，检查他们和其他y_j和u_j是否存在移位的关系。该过程中，由于输出的对称集合是明确的，同时输入对称集合内部的对应关系也是明确的，因此并不存在传统对称性检测算法[33]中的搜索过程，故整个算法是多项式复杂性的。

1.3.6. 解码算法的模板结构设计和参数提取

该步骤的关键在于两部分:

首先，我们将致力于研究每一种主要的FEC解码算法，如Euclid算法[27]、Berlekamp-Massey算法[13]和Guruswami-Sudan算法[18]等。并从中提取结构特征和可调节的参数，用于后继的搜索算法。

其次，我们将基于上述的结构和参数，开发用于参数搜索的算法。

这两方面的工作是紧密相关的。从通用性方面来看，我们需要尽量通用的结构模型，以便覆盖尽量多的解码器类型。而反过来，过于一般性的结构，将导致第二步中的算法复杂性恶化甚至不可解。上述两个方面之间的折中将是我们研究的重点。

1.4. 本项目的特色与创新之处；

首先，本项目首次提出了针对 FEC 中多位错的验证算法。

其次，本项目首次提出了针对有限域算术和 FEC 的对偶综合算法。

再次，从手段上首次提出了有限域上的增量式 Gröbner basis 构造算法，能够有效提升其他有限域形式化方法的效率。

最后，从手段上首次提出了有限域上的对称性削减算法，能够有

效提升其他有限域形式化方法的效率。

1.5. 年度研究计划及预期研究结果

研究计划安排如下:

起始时间	结束时间	研究内容
2016/1	2016/2	符号模拟器
2016/3	2016/4	错误模型
2016/5	2016/6	初步验证框架
2016/6	2017/6	增量式的 Gröbner basis 算法
2017/7	2018/6	基于 cyclic code 的对称性削减
2018/7	2019/6	模板设计和参数提取
2019/7	2019/10	对偶综合算法
2019/11	2019/12	总结和撰写结题报告

预期研究成果:

预期申请国际专利 2 项，国内专利 8 项，培养研究生 4 名，并在下列集成电路和形式化方法领域的顶级国际会议和期刊上发表论文 8 篇:

IEEE Transactions On Computer-Aided Design of Integrated Circuits and Systems(TCAD)
ACM Transactions on Design Automation of Electronic Systems(TODAES)
International Conference on Formal Methods in Computer-Aided Design(FMCAD)
Design Automation Conference(DAC)
International Conference on Computer Aided Design(ICCAD)
Design, Automation, and Test in Europe(DATE)
Asia and South Pacific Design Automation Conference(ASPDAC)

2. 研究基础与工作条件

2.1. 工作基础

本项目组的工作基础包括:

首先，本项目负责人的所在单位国防科技大学计算机学院是国际领先的超级计算机研制单位，本项目负责人沈胜宇从 2009 年开始负责了两代天河超级计算机的物理编码层和前向纠错机制的设计，为天河超级计算机荣获四次 TOP500 第一名奠定了坚实基础，并以此工作获得国家科技进步特等奖一项。同时，在 863 项目”40Gbps 串行端口

控制器”中，负责设计针对 40Gbps 信道的 FEC 编码器和解码器。

其次，在本项目负责人沈胜宇的上一个自然科学基金项目”面向通讯应用的自动对偶综合方法研究”中，我们首创了对偶综合的概念并提出相关算法，以从任意编码器的Verilog源代码自动产生其对应的解码器代码[1] [2] [3] [4] [5][6]。这些工作有效的解决了 10G以太网[40]和PCI Express[41]等常见传输标准的解码器自动产生问题。

再次，本项目负责人沈胜宇长期以来致力于形式化方法的研究，在该方面具有深厚的理论基础和广泛的兴趣范围。

2.2. 工作条件

本项目负责人所在的国防科技大学计算机学院，是国际领先的超级计算机研制单位。在高速信号传输技术方面具备多年的技术积累、强大的人才储备和先进的测量设备。

同时，项目负责人沈胜宇在形式化方法和对偶综合领域进行了多年的艰苦卓绝而富有成效的研究工作，并对相关的背景知识和最新进展充分了解。积累了大量可供重用的代码。

所有这些都为我们完成本项目的研究工作奠定了坚实基础。

2.3. 承担科研项目情况

项目负责人沈胜宇目前没有承担包括自然科学基金在内的科研项目。但是作为主要骨干参与下列科研项目：

名称	经费来源	起止年月	负责的内容
片上多核处理器验证理论与关键技术	国家自然科学基金重点项目	2012.1-2016.12	对偶综合
40Gbps 高速串行接口控制器关键技术研究	863 项目	2013.1-2015.12	FEC 设计

2.4. 完成国家自然科学基金项目情况

上一个项目是面上项目"面向通讯应用的自动对偶综合方法研究"，项目编号 61070132。已经于 2014 年 1 月结题。在结题后，我们继续进行后继研究，专注于对 FEC 和相关有限域代数的研究。这些研究成为本项目申请书的基础。

上一项目的研究工作总结摘要是:

1. 在我们发表于 ICCAD09 的原始对偶综合算法基础上,增加对环形路径条件的检测公式,将解码器在有限长度上不存在的结论,扩展到无限长路径上,从而解决了原始算法不停机的问题。研究成果第一作者发表于 ShengYu Shen: A Halting Algorithm to Determine the Existence of the Decoder. IEEE Trans. on CAD of Integrated Circuits and Systems 30(10): 1556-1563 (2011)。
2. 提出了自动推导外部断言的算法,以减少用户手工工作。该算法迭代的运行成果 1 的算法,使用每次迭代中产生的 SAT 公式,求解并剔除解码器不存在的外部断言。有限状态假设保证了该算法的停机性。研究成果第一作者发表于 ShengYu Shen: Inferring assertion for complementary synthesis. ICCAD 2011: 404-411。
3. 提出了全新算法以发掘多个同时存在的解码器及其对应断言。该算法迭代的构造函数依赖问题的 SAT 公式,以检查当现有已发掘的解码器集合具有相同的输出时,是否能够导致所有潜在解码器的混合 SAT 公式具有不同的输出。若是,则表明仍然存在尚未被发掘的解码器,而混合 SAT 公式在外部配置信号集合的解上的投影,即为新的解码器的迁移关系。该算法迭代运行直至函数依赖问题的 SAT 公式不可满足为止。研究成果发表于 ShengYu Shen: Synthesizing Complementary Circuits Automatically. IEEE Trans. on CAD of Integrated Circuits and Systems 29(8): 1191-1202 (2010)。
4. 提出基于 Craig 插值的特征化算法,以加快生成解码器的速度。该算法构造迁移关系的不可满足 SAT 公式,并从 SAT 求解器产生的 resolution 序列中产生解码器的布尔函数。该算法也发表于 ShengYu Shen: Synthesizing Complementary Circuits Automatically. IEEE Trans. on CAD of Integrated Circuits and Systems 29(8): 1191-1202 (2010)。
5. 由于本小组的开创性工作,使得对偶综合引起了国际学术界的关注和跟进研究。自 2011 年以来,在EDA领域的顶级会议ICCAD[11]和DAC[9],以及顶级期刊IEEE Transaction on CAD of Integrated Circuits and Systems[10]上共发表了 3 篇来自其他研究小组的论文。这些工作给出了意想不到的发现,是对我们研究工作的巨大推进和有益补冲。

相关成果的详细目录:

1. ShengYu Shen, Ying Qin, Kefei Wang, Zhengbin Pang, Jianmin Zhang, Sikun Li: Inferring Assertion for Complementary Synthesis. IEEE Trans. on CAD of Integrated Circuits and Systems 31(8): 1288-1292 (2012)

2. ShengYu Shen, Ying Qin, Liquan Xiao, Kefei Wang, Jianmin Zhang, Sikun Li: A Halting Algorithm to Determine the Existence of the Decoder. IEEE Trans. on CAD of Integrated Circuits and Systems 30(10): 1556-1563 (2011)
3. ShengYu Shen, Ying Qin, Kefei Wang, Liquan Xiao, Jianmin Zhang, Sikun Li: Synthesizing Complementary Circuits Automatically. IEEE Trans. on CAD of Integrated Circuits and Systems 29(8): 1191-1202 (2010)
4. ShengYu Shen, Ying Qin, Jianmin Zhang: Inferring assertion for complementary synthesis. ICCAD 2011: 404-411
5. ShengYu Shen, Ying Qin, Jianmin Zhang, Sikun Li: A halting algorithm to determine the existence of decoder. FMCAD 2010: 91-99
6. ShengYu Shen, Jianmin Zhang, Ying Qin, Sikun Li: Synthesizing complementary circuits automatically. ICCAD 2009: 381-388

3. 资金预算说明

无需购置 5 万元以上固定资产及设备。

4. 其他需要说明的问题

无