

基于公式递推法的可变计算位宽的循环冗余校验设计与实现

陈 容^{①②③} 陈 岚^{*①③} WAHLA Arfan Haider^{①②③}

^①(中国科学院微电子研究所 北京 100029)

^②(中国科学院大学 北京 100049)

^③(三维及纳米集成电路设计自动化技术北京市重点实验室 北京 100029)

摘 要: 循环冗余校验(CRC)与信道编码的级联使用,可以有效改善译码的收敛特性。在新一代无线通信系统,如5G中,码长和码率都具有多样性。为了提高编译码分段长度可变的级联系统的译码效率,该文提出一种可变计算位宽的CRC并行算法。该算法在现有固定位宽并行算法的基础上,合并公式递推法中反馈数据与输入数据的并行计算,实现了一种高并行度的CRC校验架构,并且支持可变位宽的CRC计算。与现有的并行算法相比,合并算法节省了电路资源的开销,在位宽固定时,资源节约效果明显,同时在反馈时延上也有将近50%的优化;在位宽可变时,电路资源的使用情况也有相应的优化。

关键词: 循环冗余校验; 并行算法; 公式递推法

中图分类号: TN911.72

文献标识码: A

文章编号: 1009-5896(2020)05-1261-07

DOI: 10.11999/JEIT190503

Design and Implementation of Cyclic Redundancy Check with Variable Computing Width Based on Formula Recursive Algorithm

CHEN Rong^{①②③} CHEN Lan^{①③} WAHLA Arfan Haider^{①②③}

^①(Institute of Microelectronics, Chinese Academy of Sciences, Beijing 100029, China)

^②(University of Chinese Academy of Sciences, Beijing 100049, China)

^③(Beijing Key Laboratory of Three-dimensional and Nanometer Integrated Circuit Design Automation Technology, Beijing 100029, China)

Abstract: Cyclic Redundancy Check (CRC) is used in cascade with channel coding to improve the convergence of the decoding. In the new generation of wireless communication systems, such as 5G, both code length and code rate are diverse. To improve the decoding efficiency of cascaded systems, a CRC parallel algorithm with variable computing width is proposed in this paper. Based on the existing fixed bit-width parallel algorithm, this algorithm combines the parallel calculation of feedback data and input data in the formula recursive method, realizing a highly parallel CRC check architecture with variable bit-width CRC calculation. Compared with the existing parallel algorithms, the merged algorithm saves the overhead of circuit resources. When the bit-width is fixed, the resource saving effect is obvious, and at the same time, the feedback delay is also optimized by nearly 50%. When the bit-width is variable, the use of resources is also optimized accordingly.

Key words: Cyclic Redundancy Check(CRC); Parallel algorithm; Formula recursion algorithm

1 引言

在无线通信系统的传输过程中,由于路径损耗、阴影衰落、多径传播等原因,可能会导致接收信息发生错误。为了提高传输可靠性,判断接收端

数据的正确性,除了使用信道编码技术之外,还会在发送数据的末尾加入校验位。编码校验中常用的校验技术是循环冗余校验(Cyclic Redundancy Check, CRC),因其误码检测能力强,抗干扰性能优异,在众多通信系统中得到广泛的应用,如: Ethernet^[1]、卫星通信^[2]等。

CRC利用线性码原理,按照一定规则产生校验码,附在原始数据后面一起发送,接收端以同样的规则进行检验,从而判断接收信息是否正确。在低速通信中,CRC采用串行算法,由线性反馈移

收稿日期: 2019-07-15; 改回日期: 2019-10-30; 网络出版: 2019-11-07

*通信作者: 陈岚 chenlan@ime.ac.cn

基金项目: 国家科技重大专项(2018ZX03001006-002)

Foundation Item: The National Science and Technology Major Project (2018ZX03001006-002)

位寄存器(Linear Feedback Shift Register, LFSR)实现^[3,4]。随着技术的发展,数据存储和传输速度大大提高,传统的串行算法已不适合高速通信的场合,必须采用速度更快的并行算法^[5-12]。近年来,随着信道编码的发展,尤其是以迭代译码为基础的Turbo码与LDPC码的运用,CRC会被加入到信道译码中参与迭代过程。CRC的级联使用,不仅可以评估接收器的准确性,同时也可以提高译码的性能^[13-15]。为了兼容信道编码在码长与码率方面的可配置性^[16],CRC在并行计算的位宽上也应该具有一定的灵活性。而现有的并行算法,都没有明确提出位宽变化的CRC校验情况,并且文献中大部分都以CRC校验码的长度作为并行位宽,如16, 32等。文献^[7-9]主要是从流水线的角度上,对并行CRC计算进行了优化,其中文献^[8]有提到非整数倍输入的情况。文献^[6]中的方法可以实现非整数倍的输入,但随着并行度的提高,系统的硬件资源也

会大大增加。针对级联系统中译码分段长度可变的情况,本文提出了一种可变计算位宽的CRC并行算法。

本文在第2节介绍现有固定位宽的CRC并行算法并分析其优劣;第3节在公式递推法的基础上,提出了将输入数据路径和反馈数据路径合并的优化算法,设计并验证了可变计算位宽的CRC并行算法;第4节列出了与现有算法的比较与分析;第5节是结束语。

2 固定位宽的CRC并行算法

传统的CRC校验算法是由LFSR实现的,因为其结构简单且易实现,被广泛应用于BCH和CRC编码的串行实现中^[4]。图1展示了一种经典的 r 位CRC串行校验结构,其中 $\{C_{r-1}, C_{r-2}, \dots, C_0\}$ 为寄存器, $\{g_{r-1}, g_{r-2}, \dots, g_0\}$ 为生成多项式的系数,“ $\oplus$ ”为异或运算。

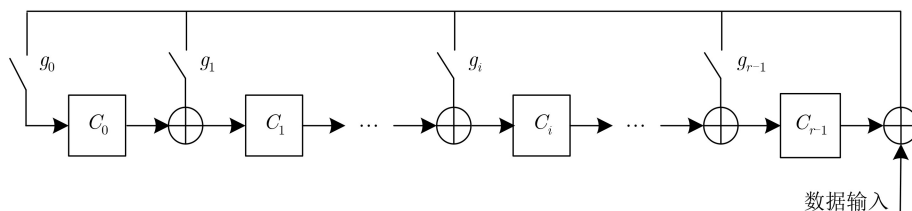


图1 LFSR实现的串行编解码结构

图1中的LFSR结构,在编码时,当所有的数据按比特输入编码器后,寄存器中保存的为生成的CRC值,再经过 r 个时钟周期串行输出。解码时,所有的接收数据(包括CRC附加比特)进入解码器后,若寄存器中的值都为0,则表示接收数据正确,若寄存器中的值不全为0,则表示接收数据有误。

虽然LFSR结构简单易实现,但串行计算需要多个时钟周期,不能快速得到计算结果。为了增加系统的吞吐率,人们提出了一些CRC并行算法,主要有查表法^[5,6]、公式递推法^[7-9],以及以公式递推为基础的展开重定时算法^[10-12]等。为了更好地表述本文提出的方法,下面对查表法和公式递推法作重点介绍。

2.1 查表法

查表法一般按字节或字节的整数倍来并行计算,预先计算出所有信息的CRC值,制作成一张查找表,编码时通过一定的计算再查表即可得到相应的值。

查表法的原理描述如下^[5]:

引理1 在二元域GF(2)上,若 $A(x) = A_1(x) +$

$A_2(x) + \dots + A_n(x)$, 那么对于一个给定的多项式,有 $\text{CRC}(A(x)) = \sum_{i=1}^n \text{CRC}(A_i(x))$ 。

引理2 给定GF(2)上的一个多项式 $B(x)$,对于任意 k ,有 $\text{CRC}(x^k B(x)) = \text{CRC}(x^k \text{CRC}(B(x)))$ 。

基于上述引理,首先将输入数据分段,查找每段的CRC值,最后将分段CRC值进行异或运算,得到最终的CRC结果。查表法执行速度比较快,常用于高速通信场合,但是需要大量的额外存储资源。

2.2 公式递推法

公式递推法是通过串行电路中寄存器值与输入输出之间的关系式,推导出多位并行输入的关系式,并在一个时钟周期内进行并行计算。在文献^[7]中,Jeff对公式递推法有详细的介绍,推导了多位并行输入的关系式,当数据流为 M bit并行输入时,其关系图与关系式表示如图2与式(1)。

$$x(m+1) = x(m) \cdot \mathbf{A}^M + u_M(m) \cdot \mathbf{B}_M \quad (1)$$

其中, $x(m)$ 表示当前状态的CRC值, $x(m+1)$ 为下一状态的CRC值, $u_M(m)$ 为当前 M 位并行输入比特。 $\mathbf{A}^M$ 表示反馈数据计算矩阵,其值为串行反

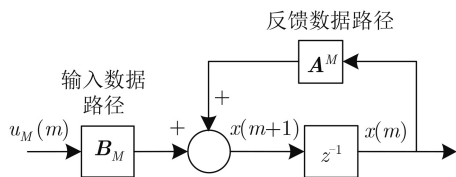


图2 公式递推法M位并行CRC计算

反馈数据计算矩阵的M次方， B_M 表示输入数据计算矩阵，其值为 $[bA^{M-1}; bA^{M-2}; \dots; bA; b]$ 。

A 和 b 分别为串行输入时的反馈数据和输入数据计算矩阵，为

$$A = \begin{bmatrix} g_{r-1} & g_{r-2} & \cdots & g_1 & g_0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \quad (2)$$

$$b = [g_{r-1} \quad g_{r-2} \quad \cdots \quad g_1 \quad g_0] \quad (3)$$

其中， r 为校验位宽， $\{g_{r-1}, g_{r-2}, \dots, g_0\}$ 为校验生成多项式的系数。

相比于查表法要预先计算出所有信息的CRC值，这种公式计算是实时的，节省了大量额外的硬件存储资源。其缺点是前期的公式推导复杂，但由于执行速度快，而且占用硬件资源少，被用于许多应用中，也吸引了许多研究者的关注^[7-12]。

3 基于公式递推法的可变位宽CRC的设计与实现

在新一代的无线通信系统中，迭代译码与CRC的级联使用是必然趋势，如5G通信中LDPC码与Polar码的译码^[17]。前面所述的并行算法多用在独立的CRC结构中，而用在级联系统中时，则需要在中间加上缓冲器以匹配信道译码输出的数据块大小，如图3所示。

缓冲器增加了系统资源与迭代译码的周期，影响译码效率。为了兼容信道编码在码长与码率方面的可配置性，本文在公式递推法的基础上，设计了计算位宽可变的CRC校验器。该校验器可以根据实际译码的有效位宽，并行计算CRC结果并反馈给译码器，不仅不需要缓冲器，还增加了系统的灵活性。

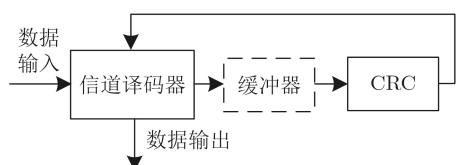


图3 CRC与信道译码的级联使用

3.1 可变计算位宽的CRC算法设计

从信道译码出来的数据段大小并不是固定不变的，因此在没有缓冲器的情况下，就需要CRC在并行计算位宽上具体一定的灵活性，这也是本文设计的CRC所满足的需求。如图4所示，Num表示译码数据的有效位宽，由数据传输块的长度和信道编码中设定的数据段最大长度等参数决定，并且有 $0 < \text{Num} \leq M$ ，其中 M 表示其最大值，由信道译码的并行度决定。

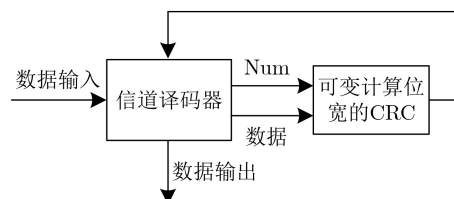


图4 可变计算位宽的CRC级联系统

图4中的“可变计算位宽的CRC”，是在现有固定位宽并行计算的基础上改进而来。在公式递推法中，CRC的计算包括两条数据路径：输入数据路径和反馈数据路径。下面通过矩阵运算，将两条数据路径合并，如图5所示。

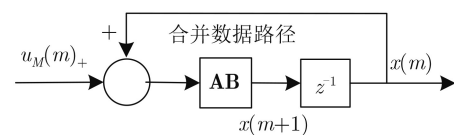


图5 M位固定位宽合并计算

其中， $u_M(m)$ 为当前输入的Mbit数据， $x(m)$ 和 $x(m+1)$ 为当前状态和下一状态的CRC值。根据Jeff的推导，可知对于反馈数据路径，当 $M=1$ 时，计算矩阵可以写为

$$A^1 = \begin{bmatrix} g_{r-1} & g_{r-2} & \cdots & g_1 & g_0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} = [b; I_{r-1} \quad 0] \quad (4)$$

其中， r 为校验位宽， I_{r-1} 为 $r-1$ 维单位矩阵， 0 为相应维度的0值列向量。而对于输入数据路径， $M=1$ 时， $B_1 = [b]$ ，因此可得

$$A^1 = [B_1; I_{r-1} \quad 0] \quad (5)$$

当输入数据位宽 $M=2$ 时，反馈数据计算矩阵为

$$\begin{aligned} A^2 &= A \cdot A = [b; I_{r-1} \quad 0] \cdot A \\ &= [bA; (I_{r-1} \quad 0) \cdot A] = [bA; b; I_{r-2} \quad 0] \end{aligned} \quad (6)$$

此时, 输入数据计算矩阵 $B_2 = [bA; b]$, 因此有

$$A^2 = [B_2; I_{r-2} \ 0] \quad (7)$$

同理, 可以得到, 当输入数据位宽比校验位宽小时, 即 $M < r$,

$$A^M = [B_M; I_{r-M} \ 0] \quad (8)$$

特殊情况, 当输入数据位宽与校验位宽相等时 ($M = r$), 有 $A^M = B_M$, 这点在文献[7]中有提到过, 即

$$A^r = B_r = [bA^{r-1}; bA^{r-2}; \dots; bA; b] \quad (9)$$

而当输入数据位宽大于校验位宽时, 即 $M > r$,

$$\begin{aligned} A^M &= A^r \cdot A^{M-r} \\ &= [bA^{M-1}; bA^{M-2}; \dots; bA^{M-r+1}; bA^{M-r}] \end{aligned} \quad (10)$$

$$B_M = [bA^{M-1}; bA^{M-2}; \dots; bA; b] \quad (11)$$

此时, $B_M = [A^M; bA^{M-r-1}; \dots; bA; b]$ 。

综上所述, 无论 M 为何值时, A^M 与 B_M 之间都有相同的部分, 即输入数据计算矩阵和反馈数据计算矩阵都有包含与被包含的关系。那么, 合并之后的计算矩阵 AB 可以表示为

$$AB = \begin{cases} A^M, & 0 < M < r \\ A^M = B_M, & M = r \\ B_M, & M > r \end{cases} \quad (12)$$

由CRC校验的原理可知, 当输入数据与反馈数据的位宽不同时, 只要保持高位对齐异或相加, 就能得到正确的CRC校验结果。当有效数据位宽 Num 变化时, 对齐的数据位也是变化的, 因此如果将式(12)中的3种情况统一起来, 则需要在反馈路径上加入选择器来保证数据高位对齐, 并在异或之前将两条路径上的数据扩展为相同的位宽, 如图6所示。

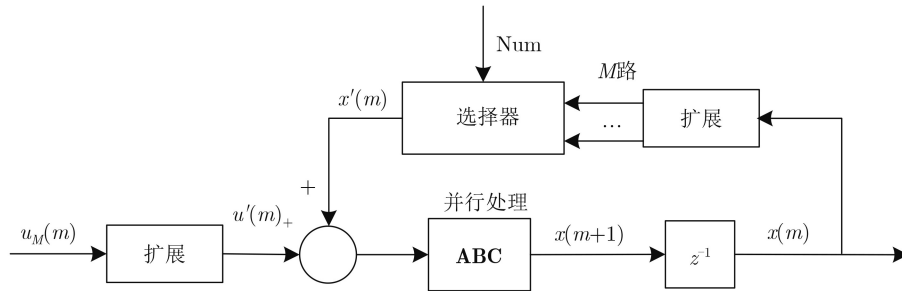


图6 1~M位计算位宽可变的CRC计算

此时, CRC计算的公式可以表示为

$$x(m+1) = (x'(m) + u'(m)) \cdot ABC \quad (13)$$

图6中的校验器主要由“选择器”和“并行处理(ABC)”两部分组成, 两个“扩展”模块完成补零操作。反馈数据经过选择器选出与输入数据对齐的高位再进行异或, 并行处理模块为合成路径, 其计算矩阵为

$$ABC = [AB; I_r] \quad (14)$$

其中, AB 由前面式(12)给出。 I_r 为 r 维单位矩阵, 加上 I_r 的目的, 是解决输入数据位宽小于CRC校验位宽的情况。因为此时有一部分反馈数据值是直接输出的, 不需要与其他数据进行异或, 可以用单位矩阵来表示这部分的电路。

在新的校验器中, 输入数据路径为数组扩展模块, 在 M 位输入数据后面添加 r 个0, 形成 $u'(m)$ 。反馈数据路径包括数组扩展模块和选择器: 首先将 r 位反馈值扩展为 M 路, 第 m 路数据 C_m 的构成为在 r 位反馈数据的前面补 $M - m$ 个0, 并在后面补 m 个0 ($0 < m \leq M$); 然后根据输入有效位宽 Num 的值, 选择其中一路作为 $x'(m)$ 。最终, 两路数据位

宽均为 $M + r$, 通过异或所得结果作为并行处理模块的输入数据, 输出即为新的CRC结果。

3.2 可变计算位宽的CRC实现与验证

现有并行算法的并行度基本上都是CRC校验码的长度, 为了体现本文算法在位宽上的灵活性, 选择最大并行度大于校验位宽的情况进行实现与验证。下面以CRC-24($r = 24$)为例, 取最大并行度 $M = 32$, 介绍可变位宽的CRC实现结构。

所用CRC-24的生成多项式为

$$g_{CRC24}(x) = x^{24} + x^{23} + x^6 + x^5 + x + 1 \quad (15)$$

由多项式的系数可得到 A 和 b 的值, 再通过式(12)和式(14)得出并行计算矩阵 ABC ($M = 32$, $r = 24$)。根据前面所提出的设计方案, 使用Verilog语言实现, 在Quartus II中选择Altera公司的Cyclone III系列中的EP3C5E144C7进行综合, RTL电路图(部分)及硬件资源开销如图7和表1所示。

在每个时钟周期, CRC校验器会根据当前CRC值和输入数据计算出新的CRC值, 直到一个数据包结束, 得到最终的校验结果。随机生成3个

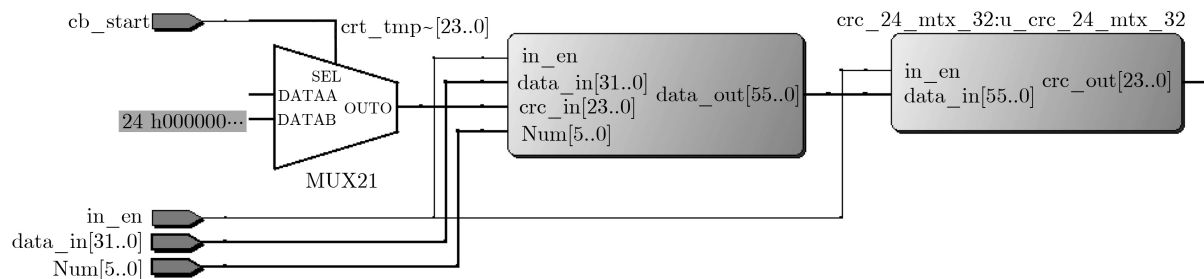


图 7 1~32位并行度可变的CRC编码RTL实现

表 1 硬件资源开销

项目	值
顶层实体名	crc_24 EP3C5E
芯片	144C7(Cyclone III)
逻辑器件数	434/5136 (8%)
寄存器数	26
管脚数	68/95(72%)

数据包，在Modelsim平台上进行CRC编码的仿真测试，并与Matlab的编码结果进行对比，如表2所示。

表2中，总长度为每个数据包所含的比特数，Num为每个数据包所分数据段的有效数据位宽，为了测试所设计CRC校验器的功能，Num在取值上涵盖了并行位宽小于24、等于24、大于24，以及等于最大并行度32的情况。Matlab的结果是根据串行算法计算所得，以二进制表示，而Modelsim的

表 2 仿真测试结果

	总长度 (bit)	Num	Matlab结果	仿真结果
数据1	60	7, 24, 29	0011110101101111 1110110	24'h3d6ff6
数据2	65	23, 32, 10	00111000001001101 1010001	24'h3826d1
数据3	70	24, 15, 31	01111110000001111 1011011	24'h7e07db

仿真结果以十六进制表示。从表2可以看出，仿真测试的结果与Matlab的计算结果是一致的，3个数据包均能得出正确的CRC结果，验证了算法的正确性。

4 比较与分析

表3列出了几个常用的CRC生成多项式^[11]，以

表 3 选用的生成多项式

CRC	生成多项式
CRC-12	$x^{12} + x^{11} + x^3 + x^2 + x + 1$
CRC-16	$x^{16} + x^{15} + x^2 + 1$
CRC-32	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

表 4 电路资源和关键路径长度比较

CRC式子($M=r$)	算法	总计		
		1	异或	关键路径
CRC-12(12)	文献[7]	136	112	9
	文献[8]	120	66	8
	文献[10]	—	103	8
	文献[9]	77	53	8
	固定	52	43	5
	可变	64	78	9
CRC-16(16)	文献[7]	218	186	10
	文献[8]	188	98	10
	文献[10]	—	94	10
	文献[9]	100	60	9
	固定	72	54	5
	可变	88	101	9
CRC-32(32)	文献[7]	1031	967	12
	文献[8]	928	518	12
	文献[10]	—	675	10
	文献[9]	888	461	12
	固定	452	313	6
	可变	484	408	11

CRC-12, CRC-16和CRC-32为代表。根据生成多项式的系数,通过式(12)和式(14)可以得到各自的并行计算矩阵,矩阵中“1”的个数与实现所需要的“异或”资源相关。

通过计算,将电路资源和关键路径长度与文献[7-10]进行比对,如表4所示。在第1行中,“1”表示并行计算矩阵中1的总数,“异或”表示电路实现中需要的两输入异或单元的总数(自适应并行

度的校验器中的选择器部分已经等价换算为“异或”资源并计入总数),“关键路径”表示从校验器的输入到输出的整条路径(包括反馈回路)上所使用的两输入异或单元的最大级数。

表4中,“固定”表示合并数据路径后固定位宽的情况,“可变”表示在反馈路径上加入选择器后位宽可变的情况。可以看出,本文设计的算法,在“固定”和“可变”两种情况下都能够有效节约电路资源,并且在时延上也表现出良好的性能。这是因为合并输入数据和反馈数据的并行计算,节省了将近一半的电路资源和时延,而在反馈路径上增加的选择器,只占用了较少部分的资源与时延。通过对比可以得到,在固定位宽的情况下,从校验器的输入到输出的关键路径长度将近为其他算法的一半,可以大大降低联合译码的反馈时延;在电路资源上,优化程度也较为明显,以 $M=16$ 为例,所用资源不到文献[7]中算法的 $1/3$,相较于文献[8–10],也相应优化了44.90%, 10.00%和42.55%。而在计算位宽可变的情况下,关键路径长度与其他算法相近,在并行度较大的情况下,如 $M=32$,资源使用情况较文献[7,8,10]有明显的优化。

5 结束语

本文针对信道编码在码长与码率方面的可配置性,提出了一种可变计算位宽的并行CRC算法。该算法在现有公式递推法的基础上,将反馈数据路径与输入数据路径合并,增加选择器结构,实现了一种高并行度的CRC校验结构,并且支持位宽可变的CRC计算。合并路径节省了电路资源的开销,也降低了反馈时延,位宽的可变性提高了系统的灵活性,在新的通信传输中具有重要的实用价值。

参 考 文 献

- [1] LI Bin, HUANG Zhiping, SU Shaojing, *et al.* Implementation of CRC in 10-gigabit Ethernet based on FPGA[J]. *Applied Mechanics and Materials*, 2014, 599–601: 1548–1552. doi: [10.4028/www.scientific.net/AMM.599-601.1548](https://doi.org/10.4028/www.scientific.net/AMM.599-601.1548).
- [2] WANG Bingrui, CHEN Pingping, FANG Yi, *et al.* The design of vertical RS-CRC and LDPC code for ship-based satellite communications on-the-move[J]. *IEEE Access*, 2019, 7: 44977–44986. doi: [10.1109/ACCESS.2019.2895746](https://doi.org/10.1109/ACCESS.2019.2895746).
- [3] CAMPOBELLO G, PATANE G, and RUSSO M. Parallel CRC realization[J]. *IEEE Transactions on Computers*, 2003, 52(10): 1312–1319. doi: [10.1109/TC.2003.1234528](https://doi.org/10.1109/TC.2003.1234528).
- [4] MUTHIAH D and RAJ A A B. Implementation of high-speed LFSR design with parallel architectures[C]. 2012 International Conference on Computing, Communication and Applications, Dindigul, India, 2012: 1–6.
- [5] HUO Yuanhong, LI Xiaoyang, WANG Wei, *et al.* High performance table-based architecture for parallel CRC calculation[C]. The 21st IEEE International Workshop on Local and Metropolitan Area Networks, Beijing, 2015: 1–6.
- [6] BAJARANGBALI and ANAND P A. Design of high speed CRC algorithm for ethernet on FPGA using reduced lookup table algorithm[C]. 2016 IEEE Annual India Conference, Bangalore, India, 2016: 1–6.
- [7] DERBY J H. High-speed CRC computation using state-space transformations[C]. GLOBECOM'01. IEEE Global Telecommunications Conference, San Antonio, USA, 2001: 166–170.
- [8] KENNEDY C and REYHANI-MASOLEH A. High-speed CRC computations using improved state-space transformations[C]. 2009 IEEE International Conference on Electro/Information Technology, Windsor, Canada, 2009: 9–14.
- [9] HU Guanghui, SHA Jin, and WANG Zhongfeng. High-speed parallel LFSR architectures based on improved state-space transformations[J]. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2017, 25(3): 1159–1163. doi: [10.1109/TVLSI.2016.2608921](https://doi.org/10.1109/TVLSI.2016.2608921).
- [10] JUNG J, YOO H, LEE Y, *et al.* Efficient parallel architecture for linear feedback shift registers[J]. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2015, 62(11): 1068–1072. doi: [10.1109/tcsii.2015.2456294](https://doi.org/10.1109/tcsii.2015.2456294).
- [11] CHENG Chao and PARHI K K. High-speed parallel CRC implementation based on unfolding, pipelining, and retiming[J]. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2006, 53(10): 1017–1021. doi: [10.1109/TCSII.2006.882213](https://doi.org/10.1109/TCSII.2006.882213).
- [12] AYINALA M and PARHI K K. High-speed parallel architectures for linear feedback shift registers[J]. *IEEE Transactions on Signal Processing*, 2011, 59(9): 4459–4469. doi: [10.1109/TSP.2011.2159495](https://doi.org/10.1109/TSP.2011.2159495).
- [13] 李伟华, 焦秉立. 一种基于分段CRC的LDPC译码的改进算法[J]. 电子与信息学报, 2008, 30(5): 1167–1170. doi: [10.3724/SP.J.1146.2006.01763](https://doi.org/10.3724/SP.J.1146.2006.01763).
- [14] LI Weihua and JIAO Bingli. Improved method for LDPC decoding algorithm aided by segmented cyclic redundancy checks[J]. *Journal of Electronics & Information Technology*, 2008, 30(5): 1167–1170. doi: [10.3724/SP.J.1146.2006.01763](https://doi.org/10.3724/SP.J.1146.2006.01763).
- [14] TUSHA A, DOĞAN S, and ARSLAN H. IQI mitigation for

- narrowband IoT systems with OFDM-IM[J]. *IEEE Access*, 2018, 6: 44626–44634. doi: [10.1109/ACCESS.2018.2864892](https://doi.org/10.1109/ACCESS.2018.2864892).
- [15] VAN WONTERGHEM J, ALLOUM A, BOUTROS J J, *et al*. On short-length error-correcting codes for 5G-NR[J]. *Ad Hoc Networks*, 2018, 79: 53–62. doi: [10.1016/j.adhoc.2018.06.005](https://doi.org/10.1016/j.adhoc.2018.06.005).
- [16] RICHARDSON T and KUDEKAR S. Design of low-density parity check codes for 5G new radio[J]. *IEEE Communications Magazine*, 2018, 56(3): 28–34. doi: [10.1109/MCOM.2018.1700839](https://doi.org/10.1109/MCOM.2018.1700839).
- [17] 王琼, 罗亚洁, 李思彤. 基于分段循环冗余校验的极化码自适应连续取消列表译码算法[J]. *电子与信息学报*, 2019, 41(7): 1572–1578. doi: [10.11999/JEIT180716](https://doi.org/10.11999/JEIT180716).
- WANG Qiong, LUO Yajie, and LI Sifang. Polar adaptive successive cancellation list decoding based on segmentation cyclic redundancy check[J]. *Journal of Electronics & Information Technology*, 2019, 41(7): 1572–1578. doi: [10.11999/JEIT180716](https://doi.org/10.11999/JEIT180716).
- 陈 容：女，1991年生，博士生，研究方向为5G通信关键技术和物理层基带芯片设计。
- 陈 岚：女，1968年生，研究员，主要研究方向为纳米及SoC芯片设计方法学、移动通讯系统低功耗技术及物联网芯片技术等。
- WAHLA Arfan Haider：男，1988年生，博士生，研究方向为基于机器学习的智能无线网络和车载网络。