

一种高速自适应 Reed-Solomon 译码结构及其 VLSI 优化实现

邱 昕 张 浩 齐中瑞 刘 壹 陈 杰
(中国科学院微电子研究所 北京 100029)

摘 要: 该文给出了一种自适应 Reed-Solomon(RS) 译码器结构。该结构可以自适应地处理长度变化的截短码编码数据块, 适合于高速译码处理。该结构使译码处理不受数据块间隙长短的约束, 既可以处理独立的编码数据块也可以处理连续发送的编码数据块。另外本译码器结构可以保证输出数据块间隔信息的完整性, 满足无线通信和以太网中特殊业务的要求。本文还基于该结构对 RS(255,239)译码器予以实现, 该译码器经过 Synopsys 综合工具综合并用 TSMC 0.18 μm CMOS 工艺实现, 测试结果验证了该译码器的自适应功能和译码正确性, 其端口处理速率可达 1.6Gb/s。

关键词: Reed-Solomon 译码器; 自适应译码; VLSI 实现

中图分类号: TN492

文献标识码: A

文章编号: 1009-5896(2009)02-0484-05

An Architecture and VLSI Implementation for Adaptive Reed-Solomon Decoder

Qiu Xin Zhang Hao Qi Zhong-ruì Liu Yi Chen Jie

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

Abstract: This paper proposes the architecture of an adaptive Reed-Solomon (RS) decoder. This adaptive decoder can decode shorten RS code with variable block length as well as variable message length. The proposed RS decoder is independent of the interval of codeword block. Consequently, it can work not only in a burst mode, but also in a continuous mode. Further, this decoder can provide the integrity of interval information between codeword blocks, thus satisfying the requirements of special services in wireless communication and Ethernet. The VLSI implementation of a RS (255,239) decoder is also presented, which is based on the architecture of the adaptive RS decoder. This adaptive RS decoder has been designed and implemented with TSMC 0.18 μm COMS technology. The testing results validate the function of the adaptive RS decoder. The port rate is up to 1.6Gb/s.

Key words: Reed-Solomon (RS) decoder; Adaptive decoding; VLSI implementation

1 引言

RS码是一种定义在 $\text{GF}(2^m)$ 域内的多进制循环块状码, 它具有同时纠突发错误和随机错误的能力, 因而广泛应用于数据通信和数据存储系统的差错控制中, 作为提高数据传输和存储可靠性的重要手段。因此被广泛应用于许多高速应用, 例如无线宽带通信、HDTV、ATM以及光纤通信等。实际应用中的RS码应用多采用截短码, 通过改变RS码的码率和纠错能力保证链路质量控制和不同应用服务的需求。例如在对802.16^[1]系统的设计中为了满足不同应用对数据速率和纠错能力可变的要求, 定义了不同信道条件下采用的截短码。同时为了保证不同业务的QoS, 不仅会产生独立解码数据块也会产生连续解码数据块, 对于某些特定应用的需求还需要保证解码后数据块之间的间隙与解码前保持不变, 其数据块格式如图1所示。这些对高速自适应的RS解码方案提出

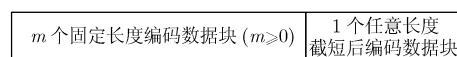


图1 IEEE 802.16 编码数据块结构

了很高的要求。

有不少文献对自适应RS解码器进行了研究^[2-4], 但在这些结构中只能通过单独改变信息位的长度来获得纠错能力的变化, 虽然可以处理连续的数据流但对截短码的适应性仍然有限。文献[5]也给出了一种能处理不同长度截短码的译码结构, 该译码结构在流水线的不同部分采用不同的时钟频率以解决数据冲突的问题, 但是其译码过程破坏了数据块间隔信息, 不能满足一些如视频、音频等高性能要求流数据业务处理的要求, 有一定的应用局限性。本文给出了一种自适应的Reed-Solomon译码器结构, 可以对定义在某 $\text{GF}(2^m)$ 域内的长度变化的截短码进行自适应解码, 不仅可以处理独立的数据块也可以处理连续的数据块。另外为了满足一些特定的应用要求, 还可以使得输入、输出数据块间隙保持不变, 保

证数据块间隔信息的完整性。如果存在反馈信道,解码器可以进一步利用自动重传机制根据信道状态的变化,动态地调整纠错能力。本文结合对RS(255, 239)高速自适应译码器的VLSI实现对该结构加以详细说明。虽然该解码器是为了满足IEEE 802.16.d协议需求而设计,但是该结构的自适应功能使其也适合以太网、磁带记录、高清电视等广泛的应用场合。

2 自适应译码器结构

本文提出的自适应 RS 译码器如图 2 所示,其主要模块包括伴随多项式计算、关键方程求解、错误计算,纠错解码输出。但是为了自适应解码的需要分别在关键方程求解模块之后以及最后解码输出部分加入了 FIFO 对数据加以缓存以保证各个数据块的延迟,避免数据处理冲突。

图 3 所示的数据块间隔示意图显示了数据块间隔在各主要模块输出的变化情况,对数据块间隔变化情况的分析可以解释该译码器结构自适应功能的实现原理。

从图 3 可以看出由于不同处理单元对不同长度数据块处理所需时间不同导致经过处理后的数据间隔发生变化。如果没有缓冲单元,数据处理必然会发生冲突,解码器不能正常工作。而本文提出的自适应解码结构正是通过引入中间级

FIFO 缓冲数据使得解码器避免了数据冲突,可以从容地处理任意长度的截短码数据块,并且不需要考虑数据块间隔的大小,既可以处理单个的数据块也可以处理连续的数据流,所以具有很强的自适应功能。但是对于某些对 QoS 要求很高的业务来说,解码器的处理改变了数据块间隔,破坏了数据块延迟的一致性使其不满足业务要求,所以还需要重新恢复原始的数据块间隔信息,所以引入输出级 FIFO 以恢复原始的数据块间隔信息。

对于输出级 FIFO,解码后的数据不断地写入 FIFO 缓存。为了获得已被破坏的原始数据块间隔信息,可以将编码数据输入有效信号延迟一定周期,利用该延迟信号作为读控制信号控制 FIFO 中的解码数据的读取,利用输入数据有效信号中保留的数据块间隔信息来恢复解码数据块间隔信息。当然对这种高业务性能的要求也需要付出一定的代价,比如处理延迟的增加以及电路规模的增大。

3 解码器实现

本文根据提出的自适应解码器结构对 RS(255,239)进行了 VLSI 实现,其实现的具体内容和优化细节在下面详细给出。

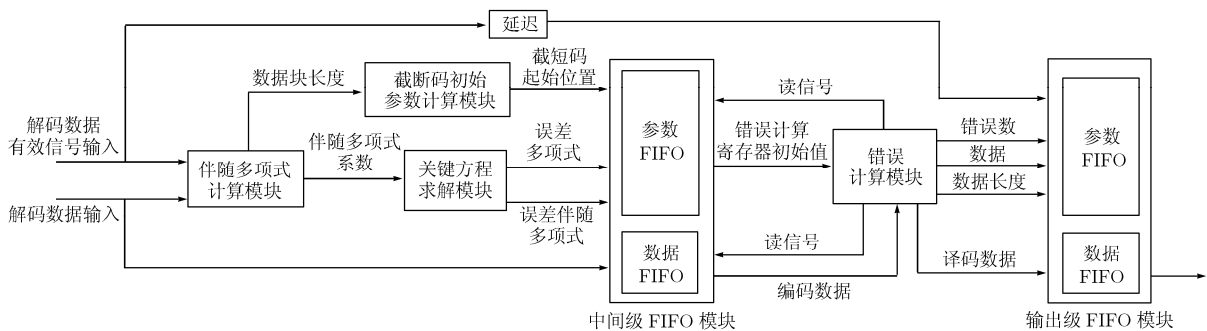


图 2 自适应 RS 译码器结构框图

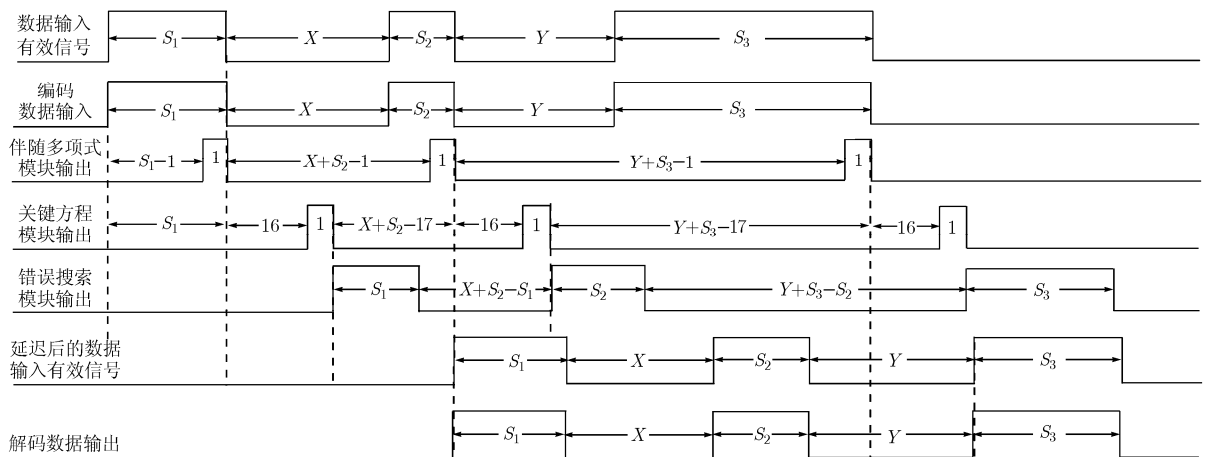


图 3 数据块间隔变化示意图

3.1 伴随多项式计算

首先需要计算的就是 $2t$ 个伴随多项式的系数, 将生成多项式 $g(x)$ 的根 α^i 代入接收到的码字多项式 $r(x)$ 就得到伴随多项式的系数 S_i , 即 $S_i = r(\alpha^i)$ 。对于接收码字多项式 $r(x)$ 有等式 $r(x) = c(x)(x + \alpha^i) + b_i$, 等式中 $c(x)$ 是信息码字多项式, b_i 是 $GF(2^m)$ 域中的常数, 将 $x = \alpha^i$ 代入方程, 则余数 b_i 即为 S_i , 所以通常伴随多项式系数 S_i 是根据通过 $r(x)$ 除以 $x + \alpha^i$ 来计算。该除法电路的实现如图 4 所示。

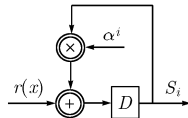


图 4 伴随多项式计算电路

由于该电路中的乘法器系数固定, 为了节省资源提高处理速度可以将该乘法器简化成单输入的固定系数专用乘法器, 用异或门组合逻辑电路来实现, 提高电路处理速度, 减少资源。同时为了对连续编码数据块进行处理, 将输出数据和下一模块的启动信号缓存后输出, 使得本模块在输出当前编码块计算结果的同时已开始处理下一编码数据块的数据, 达到连续处理数据的目的。

3.2 关键方程求解

关键方程的求解最常用的有两种算法: Berlekamp-Massey 算法^[6,7]和 Euclidean 算法^[8,9]。两种算法都是采用循环迭代的方式对伴随多项式系数进行处理, 以求解关键方程得到错误多项式 $\sigma(x)$ 和误差伴随多项式 $\omega(x)$ 。但是在高速译码器的设计中, 大量采用的是扩展的 Euclidean(eE)算法, 通过寻找两个多项式的最大公约数来求解方程。eE 算法的最大优点是计算规整, 循环处理比较简单, 文献[6]中的关键路径可以表示成 $T_{mult} + T_{add} + T_{mux}$, 其中 T_{mult} , T_{add} , T_{mux} 分别是 GF 域乘法器、 GF 域加法器和 2×1 复用器的路径延迟, 由于该路径延迟已经很小所以足够满足大多数的应用。但是随着解码技术的深入研究, 对 BM 算法的改进取得新的进展, 文献[10]中认为 BM 算法完全可以通过重构而具有规则的解码结构, 并且发现新结构不仅在数据处理速率上与 eE 结构相当, 并且具有更低的复杂度和简单的控制逻辑, 更适合于硬件实现。

由于本文中译码器的设计要求能够处理长度变化的截短码, 所以要求关键方程的求解时间小于或等于伴随多项式计算的最小时间, 才能避免两个解码数据块之间的数据冲突。而最小数据块长度为 $2t+1$ 即一个信息位 $2t$ 个校验位, 所以要求关键方程的求解必须在 $2t+1$ 个时钟周期内完成。所以我们选择采用文献[10]中重构的并行关键方程求解电路在 $2t+1$ 个时钟周期内同时完成错误多项式和误差伴随多项式的计算。另外该重构算法结构简单规整, 整个结构由一个

简单的逻辑控制单元和多个相同的逻辑计算单元串连在一起构成一个反馈回路, 控制和逻辑计算单元功能简单可靠有利于电路实现。硬件电路实现框图如图 5 所示。

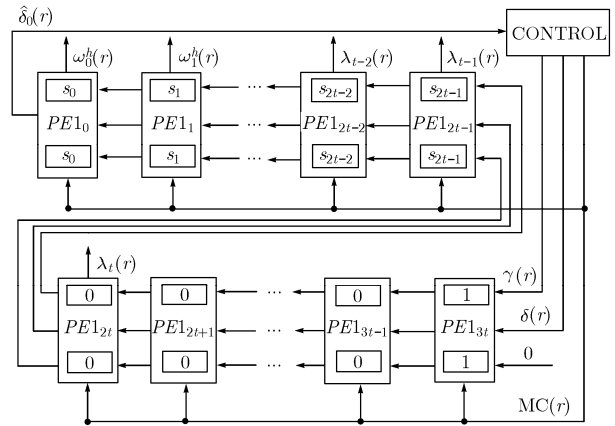


图 5 BM 重构算法硬件电路结构图

该算法关键路径可以用 $T_{mult} + T_{add}$ 来表示, 其中 T_{mult} 和 T_{add} 分别是 GF 域乘法器、 GF 域加法器。虽然重构后的算法比原始的 BM 算法多需要 $t-1$ 个乘法器和 t 个复用器, 但是大大减少了时钟周期的消耗。简单的处理单元结构缩短了关键路径, 另外信号逐级传递的结构也有效地防止了扇出过大对数据处理速率的影响, 大大提高了电路的数据处理速度, 使得该部分不再成为解码器中译码速率提高的瓶颈。

3.3 中间级 FIFO

由于数据块长度是变化的, 为了避免数据冲突, 需要在关键方程计算模块之后用 FIFO 缓冲计算得到的参数和数据, 所以该模块包括一个参数 FIFO, 一个数据 FIFO 和参数计算部分。参数 FIFO 存储后续处理所需要的配置参数, 其中包括数据块长度, 错误多项式阶数, 位置搜索寄存器和错误值计算寄存器的初始值。数据块长度信息是由伴随多项式计算模块计算产生, 用来控制搜索计算的结束。错误多项式阶数由错误多项式系数判断得到, 用来判断误码的符号数。错误位置搜索寄存器和错误值计算寄存器的初始值需要根据截断码的起始位置和错误多项式以及错误伴随多项式系数计算得到, 用来对错误计算模块进行初始化, 以避免补零操作, 直接从截短码的起始位置开始进行错误计算。

参数 FIFO 的读取由错误计算模块来控制, 如果错误位置寻找和错误值计算模块空闲且参数 FIFO 非空则立即读取 FIFO 数据进行计算。参数 FIFO 深度的选择需要满足一定条件, 该条件要求 FIFO 深度必须取大于 L_{max} / L_{min} 的整数, 其中 L_{max} 为最大数据块长度, L_{min} 为最小数据块长度, 才可以保证参数 FIFO 不会溢出。

该模块中的数据 FIFO 根据数据有效信号缓存输入的有效编码数据, 当错误计算模块开始计算时, 根据数据块长度信息读取该数据 FIFO 中的编码数据进行纠错。数据 FIFO

的深度需要满足大于 2 倍的最大编码数据块长度,才可保证数据 FIFO 不会溢出。

通过该中间级 FIFO 模块的引入即可缓冲数据和参数信息,保证在处理不同长度编码数据块时,即使是连续不同长度的数据块也不会导致数据的冲突,以实现自适应的功能。

3.4 错误计算模块

同普通译码器应用一样,本模块中的错误位置搜索算法采用钱搜索算法^[11],而错误值计算模块采用 forney 算法。但是由于需要处理长度变化的截短码,为了减少搜索周期,通过改变寄存器的初始值直接从截短码的起始位置开始计算,在设计上避免了补零计算,节省了处理时间。但是由于关键方程解算采用的是重构的 BM 算法,其输出的错误伴随多项式系数 $\omega^{(h)}(x)$ 有所变化,满足关系

$$\omega(x_i^{-1}) + x_i^{-2t} \omega^{(h)}(x_i^{-1}) = 0 \quad (1)$$

所以 forney 算法可以重写为

$$Y_i = \frac{x_i^{-(m_0+2t-1)} \omega^{(h)}(x_i^{-1})}{\lambda'(x_i^{-1})} = \frac{z^{(m_0+2t)} \omega^{(h)}(z)}{z \lambda'(z)} \Big|_{z=x_i^{-1}} \quad (2)$$

该部分硬件结构如图 6 所示。

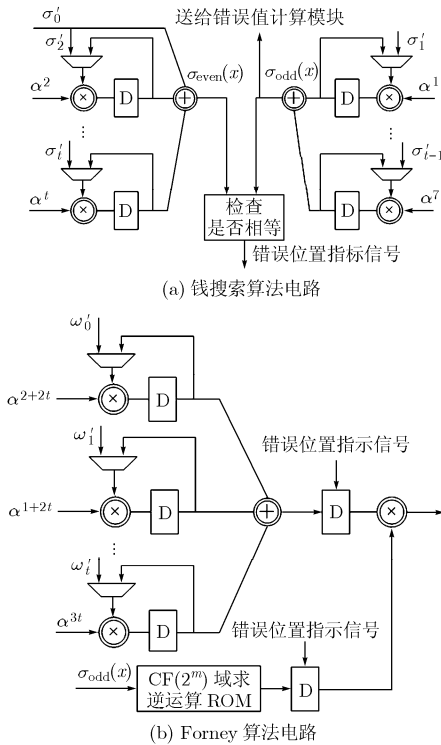


图 6 错误搜索和错误值计算电路结构图

当从 FIFO 取得数据块参数信息后,计算寄存器根据得到的截短码起始位置的寄存器初始值进行初始化,根据得到的数据块长度判断处理该数据块所需的时间。错误搜索和错误值计算同时进行,并且同步输出。

3.5 输出级 FIFO

该模块的作用主要是为了保证解码数据块与编码数据

块间隔信息一致。该模块也包括两块 FIFO,一块用来存放参数,一块用来存放纠错后的数据和原始数据。错误计算模块在处理数据块结束时输出当前数据块的长度、搜索到的错误数、错误多项式的阶数、纠错后的数据和原始未纠错的数据,这些计算结果均送入输出级 FIFO。

在该模块内部,数据块的长度、搜索到的错误数、错误多项式阶数等信息放入参数 FIFO,将纠错后的数据以及原始未纠错的数据放入数据 FIFO。将编码数据有效信号经过延迟,利用该延迟信号作为 FIFO 的读信号读取参数 FIFO 以保证输出间隔信息的完整性。

首先根据读取得到的参数判断搜索得到的错误数是否和错误多项式阶数相等,如果相等则说明解码成功,读取数据 FIFO 中的已纠错数据作为解码数据输出,否则说明解码失败,输出解码失败标志然后将输入的原始未纠错数据输出。当两个编码数据块连续输入时,编码数据有效信号持续为高,则读完当前数据块后继续读取下一个数据块参数信息保证数据的连续性,否则等待下一个读信号高电平的到来再开始 FIFO 的读取。这样将可以保证输出的解码数据块结构与输入的编码数据块结构相同,数据块之间间隔保持不变。

在设计的过程中,为了避免 FIFO 读空或者 FIFO 溢出,对编码数据输入有效信号的延迟周期必须要合适。延迟周期的长度取决于整个解码过程的最大数据延迟,需要稍大于最大长度数据块长度解码输出的延迟。对参数 FIFO 深度的设计同样需要满足大于 L_{max} / L_{min} 的条件。而数据 FIFO 的深度也需要满足大于 3 倍的最大编码数据块长度才可保证数据 FIFO 不会溢出。

4 VLSI 实现与结果

本文基于以上介绍的结构和算法用 Verilog HDL 语言设计了高速自适应 RS(255,239)译码器。首先对完成后的代码使用 Altera FPGA 来对译码器的功能加以验证,并得到该设计的速度和复杂度。当使用 Altera 器件 stratix EP1S10F 484C5 验证时整个译码器占用了 5753 个逻辑单元,可以稳定地工作在 166MHz 的工作频率,最大延迟是 561 个时钟周期。

接下来使用 Synopsys 公司的 Designer Compiler 工具采用 0.18 μm TSMC 工艺的标准单元库对该译码器进行综合和布局布线,其中缓存 FIFO 采用标准单元库中的双口 RAM 实现,最后根据 Designer Compiler 生成的门级网表进行后仿。经过大量不同结构、不同长度编码数据块的验证,数据后仿结果表明该译码器可以成功地进行自适应译码,并且保证数据块之间间隔信息不被破坏,该译码器工作的时钟频率可以达到 200MHz,端口处理速率可以达到 1.6Gb/s,其电路面积为 683092 μm^2 。

5 结论

本文提出了一种新型的自适应译码器结构。该译码器结

构有以下特点: 首先, 该结构可以处理任意长度的截短码, 自适应地处理长度变化的编码数据块; 其次, 该结构可以对任意长度间隙的编码数据块进行译码, 既可以处理单独的编码数据块也可以处理连续的编码数据块; 再次, 该结构保证了输出的数据块延迟一致, 可以满足一些特殊业务对数据块延迟的要求。该译码器结构简洁紧凑、布局合理, 适合于高速处理, 在保证强大自适应功能的同时减少了译码处理延迟, 应用范围广泛。本文进一步以该自适应译码器结构为基础, 对 RS(255,239)译码器进行了 VLSI 优化实现, 在结构设计、算法选择, 电路优化等方面做了大量的工作, 取得了较满意的结果。通过后仿测试得到该高速自适应译码器端口处理速率可以达到 1.6Gb/s, 电路面积为 $683092 \mu\text{m}^2$ 。对该高速自适应译码器的大量验证也证明了本文所提出的自适应译码器结构的有效性和通用性。

参考文献

- [1] IEEE Std 802. 16. IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems[S], 2004.
 - [2] Preez An D, Swarts F, and Agdhasi. F. A flexible Reed-Solomon codec [C]. Proc. Of African 1999, Cape Town, South Africa, IEEE Volume 1, 28 Sept.-1 Oct, 1999 Vol.1: 93-98.
 - [3] Cho Sungrae, Goulart A, and Akyildiz I F, *et al.* An adaptive FEC with QoS provisioning for real-time traffic in LEO satellite networks [C]. ICC 2001. IEEE International Conference on Communications, Helsinki, Finland, 11-14 June 2001, Vol.9: 2938-2942.
 - [4] Akyildiz F, Joe I, Driver H, and Ho Y L. An adaptive FEC scheme for data traffic in wireless ATM network [J]. *IEEE Trans. on Networking*, 2001. 9(4): 419-422.
 - [5] Song Moon Kyou, Kong Min Han, and Won Hee Sun. A variable Reed-Solomon decoder using separate clocks in its pipelined steps [C]. PIMRC 2004. 15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Barcelona, Spain, 5-8 Sept. 2004, Vol.2: 1322-1326.
 - [6] Strollo A G M, Netro N, and Caro D De. An area-efficient high-speed Reed-Solomon decoder in 0.25um CMOS [C]. ESSCIRC 2004. Proceeding of the 30th European Solid-State Circuits Conference, Leuven, Belgium, 21-23 Sept. 2004: 479-482.
 - [7] Buerner T, Dohmen R, and Zottmann A, *et al.* Wijngaarden. On a high-speed Reed-Solomon codec architecture for 43Gb/s optical transmission systems [C]. 24th International Conference on Microelectronics, NIS, Sepspa and Montenegro, 16-19 May 2004, Vol.2: 743-746.
 - [8] Xie Jun, Tu Xiaodong, and Yuan Songxin, *et al.* A low-complexity Reed-Solomon decoder for GPON [C]. 2006 International Conference on Circuits and Systems Proceedings, Kos, Greece, June 2006, Vol.4: 2488-2492.
 - [9] Lee Hanho. An ultra high-speed Reed-Solomon decoder [C]. ISCAS 2005. IEEE International Symposium on Circuits and Systems, 23-26, May 2005, Vol. 2:1036-1039.
 - [10] Sarwate D V and Shanbhag N R. High-speed architecture for Reed-Solomon decoder [J]. *IEEE Trans. on Very Large Scale Integration System*, 2001, 9(5): 641-655.
 - [11] 周晓方. 适用于多种数位传播标准之通道解码器设计与实现 [D]. [硕士论文], 台湾国立中山大学, 2004.
- 邱 昕: 男, 1980 年生, 博士生, 研究方向为无线通信系统算法与 SOC 芯片实现.
- 张 浩: 男, 1975 年生, 博士, 研究方向为无线通信系统算法与 SOC 芯片实现.
- 亓中瑞: 男, 1979 年生, 博士生, 研究方向为无线通信系统算法与 SOC 芯片实现.
- 陈 杰: 男, 1963 年生, 研究员, 研究方向为通信与多媒体 SOC 芯片实现.