基于FPGA自适应高速RS编译码器的IP核设计*

李云鹏[1]，王新梅[1]，谢显中[2]
(1. 西安电子科技大学ISDN国家重点实验室,陕西 西安 710071;
2. 重庆邮电学院 移动通信工程研究中心,重庆 400065)

摘 要:针对IP核设计方法讨论了一种可动态配置编码方案的高吞吐率RS编译码器。该编译码器采用Euclid算法实现译码,编译码过程采用流水线结构提高速率。整个设计使用VHDL语言描述,并在Xilinx公司的Virtex系列上实现验证。

关键词:Euclid算法;有限域乘法;IP核;流水线结构;VHDL语言
中图分类号:O157.4,TN762 **文献标识码**:A

An IP Core Design for Reconfigurable Parameter High-Speed Reed-SolomonEncoder/Decoder on FPGA

LI Yun-peng , WANG Xin-mei, XIE Xian-zhong
(1. *ISDN National Key Laboratory of Xidian Universtiy, Xi'an* 710071, *P. R. China*; 2. *Mobile Comm. R&D Center, Chongqing University of Posts and Telecommunications, Chongqing* 400065, *P. R. China*)

Abstract: This paper discusses a method of IP Core design for high-performance Reed-Solomon Encoder/Decoder, which has high throughput and can reconfigure coded-parameter according to the channel changes in use. The Euclid Algorithm is used to implement decoding in this paper. Through pipeline, the latency of decoding process reduces dramatically. The whole design is described in VHDL, and implement verification is realized on Virtex family of Xilinx.

Key words: Euclid algorithm; finite field multiplier; IP core; pipeline; VHDL

0 引 言

RS(Reed-Solomon)码是一类特殊的BCH码,它的纠错能力很强,而且能够纠正随机的和突发的错误,在移动通信、卫星通信、微波传输、DVB等中有着非常广泛的应用。RS码的实现方案有许多,如目前已经非常成熟的有在DSP上软件实现和专用ASIC电路实现方案。本文讨论其FPGA实现技术。近年来随着微电子工艺的进步,在单个芯片上已可以集成成千上万乃至上亿个晶体管,实现了以前需要印刷电路板甚至机架才能完成的功能。在这样高的集成度下,设计难度已变得非常高,设计代价事实上主导了芯片的代价。这不仅要求设计者必须具备系统和芯片两方面的知识,同时也必须充分考虑市场竞争的压力,最大限度地缩短设计周期。凡事从零作起的思路显然不能适应这种新情况,而采用前人成功的设计经验和设计资料是解决这个问题的明智选择。IP(Intellectual Property)核就是一个可重复利用的模块。IP核也有人称为系统宏单元、虚拟部件、芯核,是指一个经过精心设计、通过具体电路验

* 收稿日期:2002-06-05
基金项目:本文受新一代野战网装备子项目"无线传输可靠性"资助(YB—JM0005)。
作者简介:李云鹏(1978-),男,江苏人,硕士研究生。主要研究方向为移动通信中信道编码与调制关键技术的应用研究。

证的能提供正确的接口信号的可重用的功能模块。IP 模块的再利用，除了缩短 SOC 芯片设计时间外，还可以降低设计和制造的成本，提高可靠性，因而将会给IC 产业和电子工业带来巨大的经济效益。本文就是针对IP 核设计可动态配置参数高速的RS 编译码器。

从实现形式和应用层次上来看，IP Core 可以有 3 种不同的表现形式：软核(Soft-core)、固核(Firm-core)、硬核(Hard-core)。软核以硬件描述语言的形式提交，其性能通过时序模拟进行验证。由于软核不依赖于任何实现工艺或实现技术，具有很大的灵活性。使用者可以方便地将其映射到自己所使用的工艺上去，可复用性很高。本文IP 核设计就是以软核形式提交，只是在 Xilinx 公司的 Virtex 系列上实现并通过验证，并未根据具体芯片电路，实现工艺做面积，功耗的优化。针对IP 核设计的特点，本文设计的重点和难点是如何做到参数化设计，在固化电路后还能动态配置码形；如何在电路复杂性与工作速度上作出适当折中。

1　RS 编译码器的总体方案

为了可动态配置参数，RS 译码器应能产生多种码形的Reed Solomon 的宏编码器和译码器，其主要参数有：码长n；校验符号数目k；符号的位数m；不可约域多项式g 和生成多项式的第一个根。这样可以针对通信条件不稳定、信道变化大需实时调整纠错编码方案的情况，由这些参数可产生任意一个所需的码形。若为尽可能的简化电路复杂性，同时又不失灵活性，可将符号位数m 和不可约域多项式g 予以一定的限制，如$m=8$ 时不可约域多项式限制为

$$g(x) = x^8 + x^7 + x^2 + x + 1$$

$$g(x) = x^8 + x^4 + x^3 + x^2 + 1$$

上二式分别是CCSDS 的推荐标准和DVB 的标准。这些参数在通过初始化信号的指示根据需要动态加以配置。其译码单元的模块框图如图 1 所示。

图 1 中，各模块采用流水线作业，可以同时进行工作。本设计采用无阻塞设计，即编译码器可在任何时刻任何状态下接受外来数据，在具体应用时就不需要在其外再加上缓存。

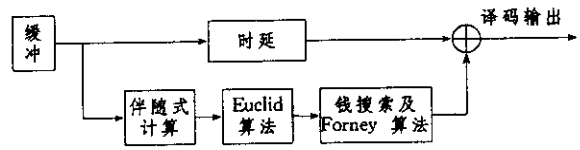

图1　译码单元的模块框图

Fig. 1 A modular block diagram of decoding cell

它的计算错误位置多项式$\sigma(x)$单元采用改进的Euclid 算法，其流程如图2 所示。这种算法的主要优点是将错误位置多项式和错误值多项式一起求出，缺点是需作一次求逆运算。

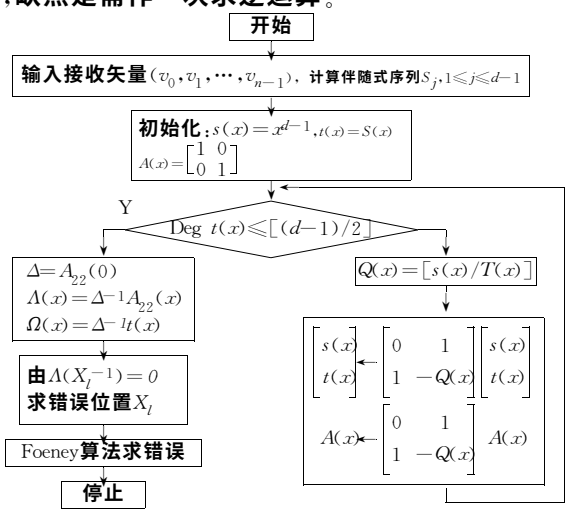

图2　Euclid算法译码流程

Fig. 2 Flow of euclid algorithm decoding

2　设计中的关键问题讨论

2.1　有限域乘法

有限域乘法在RS 编译码过程中是一个被反复用到的运算，其优化实现是RS 编译码器设计中最关键的问题，它对整个系统硬件资源的简化，性能的提高有非常大的影响。因为本设计的域运算不涉及大域运算，即符号位被限制在 16 位以下，故可以为提高编译码的速率选择并行有限域乘法器(硬件复杂度可以接受)。最基本的有限域乘法器是乘数与被乘数都是由自然基用m 位二进制数表示的乘法运算。这种有限域乘法器是由一般的不带进位的$m \times m$ 位并行乘法器，加上求模单元得到。而本设计中采用改进的Booth 算法产生部分积，利用Wallace 加法树减少部分积的数目，最后用超前进位加法器来产生$2m$ 位乘积数的乘法器可在运算速度与器件复杂性上取得平衡。

由于参数化的需要，本设计中还涉及到另一种

有限域乘法器，即乘数由自然基用 m 位二进制数表示，被乘数由本原根的幂次表示的乘法运算。这种乘法器的通用化设计是本设计实现参数化，可动态配置化的最大困难，用VHDL语言描述不当则只能实现功能仿真，要想能综合便需考虑实际的电路模型。在软件实现中常采用生成一个大的映射表，可方便地作出2种表达形式的转换。硬件实现则考虑到这样做需存储单元太多，不易实现。本文采用的方法是只将域中的正规基元素的2种表达方式用映射表存储，在做乘数由自然基用 m 位二进制数表示，被乘数有本原根的幂次表示的乘法运算时，首先将幂次数转换成 m 位二进制数，然后调用映射存储表就可用最多 m 个 m 位二进制数连乘的形式表示出被乘数，因此做最多 m 次第一种有限域乘法后，即可求出结果。其函数VHDL程序如下：

```
CONSTANT index_alpha : index_type := ("00000010","00000100","00010000",
        "00011101","01001100","10011101","01011111","10000101");
FUNCTION gfi_mul (
    aa: STD_ULOGIC_VECTOR(mm-1 DOWNTO 0);
    index: INTEGER RANGE 0 TO nn )
    RETURN STD_ULOGIC_VECTOR IS
    VARIABLE bb: STD_ULOGIC_VECTOR(mm-1 DOWNTO 0):= (others => '0');
    VARIABLE temp: STD_ULOGIC_VECTOR(mm-1 DOWNTO 0):= (others => '0');
BEGIN
    bb:=aa;
      temp:= int2suv(index,mm);
      for i in 0 to mm-1 loop
        if temp(i)='1' then
         bb:= gf_mul(bb,index_alpha(i));
        end if;
  end loop;
RETURN bb;
  END gfi_mul;
```

（常量index_alpha是$GF(2^8)$域上存储的正规基元素用自然基以 m 位二进制数表示形式。）

2.2 译码模块的实现

Euclid算法是本设计中整个译码过程的核心算法，其所需运算量较大。Euclid算法是用来计算错误位置多项式和错误值多项式的。其错误位置多项式为：

$$\Lambda(x)=\prod_{l=1}^{v}(1-\alpha^{j_l}x)=(1-\alpha^{j_1}x)(1-\alpha^{j_2}x)\cdots(1-\alpha^{j_v}x)=1+\Lambda_1x+\cdots+\Lambda_vx^v$$

上式中，$\Lambda(x)$的根α^{-j_l}正是错误位置$X_l=\alpha^{j_l}$的逆。求错误位置就是求解$\Lambda(x)$的根。要求解错误位置多项式的根即首先要确定$\Lambda_1,\Lambda_2,\cdots,\Lambda_v$。已知伴随式多项式$S(x)$，即错误值多项式为：

$$\Omega(x)=S(x)\Lambda(x)\quad(\mathrm{mod}x^{d-1},\ \deg\Omega(x)<\deg\Lambda(x)\leqslant(d-1)/2$$

Euclidean算法：设，$s^{(0)}(x)=x^{d-1}$，$t^{(0)}(x)=S(x)$，$\boldsymbol{A}^{(0)}(z)=\begin{bmatrix}1&0\\0&1\end{bmatrix}$，按下面的公式递推：

$$Q^{(r)}(x)=\frac{s^{(r)}(x)}{t^{(r)}(x)}\text{（表示 }s^{(r)}(x)\text{ 除以 }t^{(r)}(x)\text{ 的商）}$$

$$\boldsymbol{A}^{(r+1)}(x)=\begin{bmatrix}0&1\\1&-Q^{(r)}(x)\end{bmatrix}\boldsymbol{A}^{(r)}(x),$$

$$\begin{bmatrix}s^{(r+1)}(x)\\t^{(r+1)}(x)\end{bmatrix}=\begin{bmatrix}0&1\\1&-Q^{(r)}(x)\end{bmatrix}\begin{bmatrix}s^{(r)}(x)\\t^{(r)}(x)\end{bmatrix}$$

一旦$\deg t^{(R)}(x)<(d-1)/2$，停止递推。取$\Omega(x)=t^{(R)}(x)$，$\Lambda(x)=A_{22}^{(R)}(x)$后，再把所得到的两个多项式乘以$\Lambda(x)$的最低项的系数的逆元即是所需的错误值多项式和错误位置多项式。

这种算法主要缺点是涉及到有限域多项式除法。而用改进的Euclid算法可使其整个过程只作一次有限域求逆运算，其方法是将过程中用到的有限域除法转换为有限域乘法（具体做法是分子多项式乘以分母多项式的最高项系数，经过反复迭代即可），由于在最后要把得到的错误值多项式和错误位置多项式转变为尾一多项式，即各项除以多项式的最低项系数所得的多项式，所以计算迭代过程中所乘系数并不影响所求多项式的结果。这里需作一次有限域上的求逆运算。域元素α的逆元是α^{2^m-2}，例如在$GF(2^8)$域上，α的逆元即为$\alpha^{2^8-2}=\alpha^{254}$。采用的Euclid及Fermat定理相结合的求逆方法，利用平方及乘法运算实现求逆。即：

$$m1:1\to\alpha^2\to\alpha^4\to\alpha^8\to\alpha^{16}\to\alpha^{32}\to\alpha^{64}\to\alpha^{128}$$

$$m2:1\to\alpha^{0+2}\to\alpha^{2+4}\to\alpha^{6+8}\to\alpha^{14+16}\to\alpha^{30+32}\to\alpha^{62+64}\to\alpha^{126+128}=\alpha^{254}$$

用这种方法求逆，整个过程是在固定的 m 个节拍里完成，它所需电路复杂性较低，而且便于控制以

及被其他模块使用。改进的Euclid 算法大大降低了其运算量，与另一种RS 译码算法BM 算法相比所需运算量是基本等同的，还可以获得同时得到错误位置多项式和错误值多项式的好处。

在确定了 $\Lambda(x)$之后，求解 $\Lambda(x)$ 的根仍是很困难的。1964 年，钱闻天提出了一个求 $\Lambda(x)$根的实用方法。

因为解 $\Lambda(x)$的根就是确定 $v(x)=v_0+v_1x+\cdots+v_{n-1}x^{n-1}$中的哪几位产生了错误，若校验第 k 位是否有错，只需检验 $\alpha^{-(n-k)}$是否是 $\Lambda(x)$的根，若 $1+\Lambda_1\alpha^k+\Lambda_2\alpha^{2k}+\cdots+\Lambda_t\alpha^{tk}=0$，则第 k 位有错，反之，则正确。

这样依次对每一个 $\alpha^{-(n-k)}(k=1,2,\cdots,n)$ 进行检验，就求得了$\Lambda(x)$的根 ，这个过程称为钱搜索。钱搜索的硬件实现也很简单，其 VHDL 语言描述如下：

```
FOR i IN 1 TO tt LOOP
  sigma(i) <= gfi_mul(sigma(i),i);
END LOOP;
sigmasum := sigma(0);
FOR i IN 1 TO tt LOOP
  sigmasum := sigmasum XOR sigma(i);
END LOOP;
```

（sigma 为错误位置多项式寄存器）

需要特别指出的是，在缩短码实现上，应对钱搜索进行预计算，使其搜索范围缩小，VHDL 语言描述如下：

```
pre_cnt := nn-ll;
    FOR i IN 1 TO tt LOOP
     sigma(i) <= gfi_mul(sigma(i),pre_cnt);
    IF pre_cnt >ll-1 THEN
     pre_cnt := pre_cnt -ll;
    ELSE
     pre_cnt := nn+ pre_cnt-ll;
    END IF;
END LOOP;
```

求得错误位置后，利用Euclid 模块所得的错误值多项式，采用Forney 算法即可求出错误值。

2.3　流水线设计和无阻塞设计

流水线作业是提高编译码器吞吐率的重要手段，这主要表现在以下 2 个方面。

（1）在设计实现后通过时序仿真，找出用时最多的单步运算，若大大超过其他单步运算的时间，则这个运算便是提高系统时钟频率的瓶颈。如果此运算又不是在编译码过程中被非常频繁的调用，则可这个运算转为在 2 个甚至多个时钟中完成，以提高整个系统的时钟频率。这样虽然使编译码器的迟延时钟数稍稍变高，但由于整个系统时钟频率的明显提高，从而使编译码器的吞吐率得以提高。

（2）在降低编译码器的迟延时钟数而提高编译码器吞吐率方面。它能使编译码器的各个模块同时进行并行运算，最大可能的减少空等，这样编译码器的迟延时钟数将稍稍大于延时最多的模块而不是等于各模块延时的总和，大大降低编译码的延时周期。要想各模块同时工作需在各模块间加入握手信号用来指示各模块当前状态，按照一定的握手协议实现各模块间的协调工作。同时还需加入一些流水线寄存器以避免冲掉一些数据。运用这种方法设计时，对整个系统规划模块要注意考虑各模块的用时尽量平均一些，以得到最佳效果。

由于RS 译码整个过程中对所处理的那包数据要一直保存加以使用，若要等到一包数据处理结束输出完后才接受下一包数据，就造成了对外来数据的阻塞，而且带来整整一包数据的延时，同时使各模块的流水线作业失去意义，大大降低了系统性能。本文采用双口 RAM 作为缓冲加在前端，一端写入外来的数据，另一端既可以进行读操作又可以进行写操作，用多个指针变量控制读写操作，使得编译码器在处理数据的同时还能接受数据，各模块也能流水作业。这样做需要特别注意的是由于译码处理的各包数据中出错数不同，可能导致后一包数据已处理完而上一包数据却还没有（譬如，前一包数据出错数较多需处理时间较长，而后一包没有出错在前端模块即可判断出）的情况发生。这里需小心处理，加入一些必要的信号作标识以保证译码的正确。

3　实现与验证

本文选用Xilinx 公司的Virtex 系列作为目标器件，用 VHDL 语言描述了整个设计，并在 Xilinx ISE4.1 开发系统中完成整个设计的输入、功能仿真、时序仿真。编译码器的资源占用情况由配置参数的上限决定，规模随可配置参数中符号 （下转 43 页）

零交叉检测器），找到图像中具有“边缘特性”的某些点，用这样的点作为插值结点插值出整个阈值曲面，从而实现对图像的分割。作为零交叉检测器的LoG算子具有一些固有的优点，它原理简单，便于计算，在某种意义上是“自适应”的，即不涉及阈值问题，只要其高斯空间常数σ和卷积窗口尺寸选取合适，就能够给动态阈值曲面的插值过程提供有用和准确的信息，从而避免了非均匀光场给图像分割过程带来的影响。

参 考 文 献

[1] 马尔 D. 视觉计算理论[M]. 北京：科学出版社，1988.

[2] 贾云得. 机器视觉[M]. 北京：科学出版社，2000.

[3] 章毓晋. 图像分割[M]. 北京：科学出版社，2000.

[4] 杨正远，郑建宏. 小波在图像边缘检测中的应用[J]. 重庆邮电学院学报（自然科学版），1997，9(1)：5-9.

[5] NAKAGAWA Yasuo，ROSENFELD Azriel. Some experiment on varible thresholding[J]. Pattern Recognition，1979，11：191-204.

[6] PAL Nikhil R，PAL Sankar K. A review on segmentation techniques[J]. Pattern Recognition，1993，26(9)：1277-1294.

[7] YANOWITZ S D，BRUCKSTEIN A M. A new method for image segmentation[C]. Proc 9ICPR，1988：270-275.

（编辑：郭继笃）

（上接28页）

位数m，校验符号数目k的最大值呈几何增长。

本文讨论了一种可动态配置编码方案的高吞吐率RS编译码器核，它能综合到可编程逻辑中，可以作为一个IP模块在不同的系统中被反复利用。

参 考 文 献

[1] 王新梅，肖国镇. 纠错码—原理与方法[M]. 西安：西安电子科技大学出版社，1991.

[2] 林舒，科斯特. 差错控制编码基础和应用[M]. 王新梅，王育民译. 北京：人民邮电出版社，1983.

[3] STEPHEN B Wcker，VIJAY K Bhargava. Reed-Solomon codes and their applications [Z]. IEEE PRESS 1994.

[4] SJOHOLM S. 用VHDL设计电子线路[M]. 边计年译. 北京：清华大学出版社，2001.

[5] 候伯亨，顾新. VHDL硬件描述语言与数字逻辑电路设计[M]. 西安：西安电子科技大学出版社，2000.

[6] PARR Christor. A new architecture for a parallel finite field multiplier with low complexity based on composite fields[J]. IEEE Transactions on Computers，1996，45(7)：856-861.

[7] 陈弘毅. 信息系统芯片设计方法的若干关键问题[J]. 中国集成电路，2001，(30)：34-39.

（编辑：刘勇）