突发通信中Turbo码的FPGA实现

沈业兵，罗常青，安建平， 程 波

北京理工大学 电子工程系通信与网络实验室，北京100081

2008-04-11

摘 要： 给出了低复杂度和低延迟的Turbo码编译码的FPGA实现方案，方案中分量码译码算法采用Max-Log-Map算法。基于提出的设计方案，在Xilinx的FPGA芯片上实现了帧长在64～1024之间可变的短帧长Turbo编译码模块。仿真和测试结果表明，该模块的误码率性能优良、译码延时较小、数据吞吐量大，可用于低信噪比条件下突发数据通信中的差错控制。

关键词： Turbo码 FPGA Max-Log-Map算法 突发通信

Turbo码是一种低信噪比条件下也能达到优异纠错性能的信道编码。早期为了强调Turbo码接近香农限的优异性能，研究的码字长度非常大[1～2]，存在译码复杂度大、译码时延长等问题。突发数据通信以传输中小长度的数据报文业务为主，所以突发通信中的Turbo码的码长也是中等长度以下的。本文面向突发数据通信中的信道编码应用，研究了短帧长Turbo码编译码算法的FPGA实现。实现中采用了优化的编译码算法，以降低译码复杂度和译码延时。最后仿真和测试了Turbo译码器的纠错性能和吞吐量。

1 Turbo码编码器的FPGA实现

Turbo码的编码器是由两个RSC（递归系统卷积码）分量编码器和一个交织器组成。RSC码不仅具有系统码的优点，而且对于一个RSC码，总存在一个具有完全相同栅格结构的NSC码（非系统卷积码）。本系统中使用两个相同的RSC编码器，生成的多项式都是G＝[1，15/13]，系统编码率为1/3。

交织器的功能是利用随机化的思想将两个相互独立的短码组合成一个长的随机码。本课题中Turbo码交织器的实现是构造一个交织地址发生器，并根据输入的帧长信息，实时地产生交织地址序列。

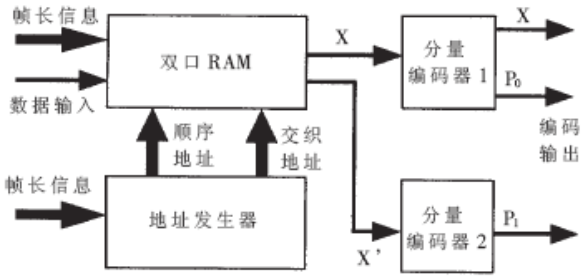

图 1 Turbo 码编码器的 FPGA 实现结构图

图1为编码器的FPGA实现结构图。编码前，地址发生器获取帧长信息，完成交织地址生成的准备过程。编码时，信息序列被依次写入双口RAM，待写完一帧数据后，地址产生器开始生成顺序地址和交织地址。双口RAM按两个地址读取信息序列X和交织后的信息序列X＇进行RSC编码；最后编码器输出系统位X和校验位P_0和P_1。

2 Turbo码译码器的FPGA实现

Turbo码译码器比较复杂，下面从译码器的接口、内部结构、内部的时序控制、分量译码MAX-Log-MAP算法和SISO模块的实现五个方面来详细阐述译码器的FPGA实现。

2.1 译码器的接口

Turbo码译码器顶层模块的接口管脚如表1所示。

热点专题

- 中国电子学会Xilinx杯开放源码硬件创新大赛
- 赛灵思公司Virtex-5系列FPGA
- 3G知识
- IPTV
- 触摸屏技术
- RoHS

杂志精华

- 基于CC2430的无线传感器...
- 无线传感器网络应用系统综述
- 无线传感器网络在野外测量中的...
- 基于竞争的无线传感器网络
- 用于矿井环境监测的无线传感器...
- 具有自适应通信能力的无线传感...
- 基于传感器网络技术的深孔测径...
- 基于无线传感器网络的家庭安防...
- 基于ATmega128L与C...
- 无线传感器网络中移动节点设备...

表 1 译码器顶层管脚定义

信号名称	信号类型	bit 宽度	信号描述
clk	输入	1	时钟信号
reset	输入	1	复位信号，高有效
din	输入	位宽×3	译码器输入的软判决数据
iteration	输入	4	译码迭代次数
packet_len	输入	11	帧长
fd	输入	1	第一位数据到达有效信号，高有效。当其有效时，译码器按时钟从 din 连续接收一帧数据
ready	输出	1	译码输出有效信号，高有效。当其有效时，译码器按时钟从 dout 口输出一帧译码数据
dout	输出	1	译码输出

2.2 译码器的内部结构

Turbo码译码器由两个软输入/软输出分量译码器、交织器以及相应的解交织器构成。译码是信息在两个分量译码器之间迭代运算的过程。在迭代运算中，上一次运算得到u_k的外信息$\Lambda_e(uk)$作为下一次运算uk的先验信息$\Lambda_a(u_k)$。Turbo码分量译码器译码算法主要有MAP类（最大后验概率译码算法）和SOVA类（软判决Viterbi译码算法）[3]。本文采用运算复杂度和性能都适中的MAX-Log-MAP算法。Turbo码译码器FPGA实现的内部结构如图2所示。

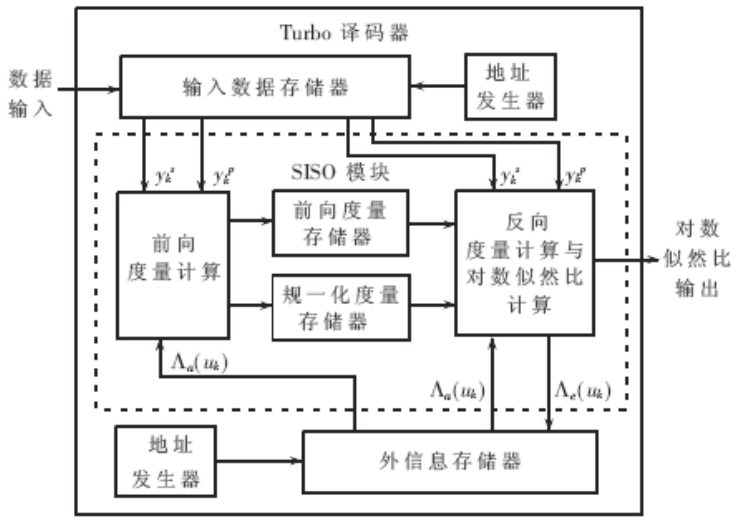

图 2 Turbo 码译码器 FPGA 实现结构框图

地址发生器与编码器相同，用于数据的交织和解交织。输入数据存储器用于存储输入的接收数据，包括系统信息序列存储器以及各个校验序列存储器。外信息存储器用于存储迭代译码产生的外信息。由于外信息要作为下一次译码的先验信息，所以这里的外信息存储器有两块，交替存储两个分量译码器的外信息。SISO模块即为软输入、软输出分量译码器。整个Turbo码译码器有两个SISO分量译码模块。但为了节省资源，本方案只设计了一个SISO模块，将时分复用作为两个分量译码器。图2中，y_k^s表示接收码字中的系统位，y_k^p表示接收码字中的校验位。

2.3 译码器内部的时序控制

Turbo码译码器内部的时序控制由状态机完成。整个译码过程分为初始化、接收数据存储、迭代译码及硬判决输出四个过程，且对应于状态机的INIT、STORAGE、SISO和OUT四个状态。译码器的内部状态转移如图3所示。初始状态INIT完成帧长设定等初始化工作，并完成交织地址生成的准备过程，一旦指示第一个数据输入的fd信号有效（高有效）时，则进入STORAGE状态；状态STORAGE完成将接收数据序列存入单口RAM中，待一帧数据写完后，指示存储完毕的rdyStr信号置高，进入SISO状态；在状态SISO下，SISO分量译码器根据设定的迭代次数对接收数据进行迭代译码。当迭代完成时，rdySiso置高，进入OUT状态；对数据硬判决输出并计数，此时输出有效信号ready置高，待全部判决完毕后返回INIT状态。

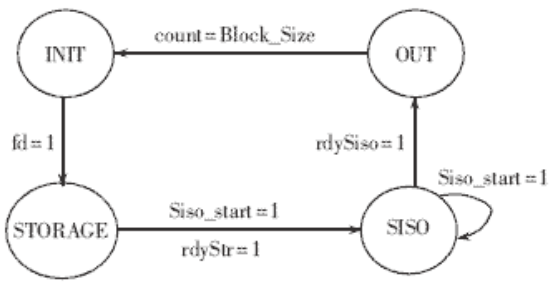

图 3 Turbo 码译码器的内部状态转移图

2.4 分量译码算法——MAX-Log-MAP算法

MAP算法需要大量的乘法运算和指数运算以及大量的存储，运算十分复杂。Log-MAP算法则将MAP算法中的乘法运算转换为对数域中的加法运算（不需要对数运算），适合工程实现。因此在工程实现时，可以将原来在对数域内的加法运算转换为取两个数的较大者加上一个修正项的运算。如果将修正项的运算也省略，则Log-MAP算法可简化为MAX-Log-MAP算法。MAX-Log-MAP算法的主要计算步骤如下[4~5]：

(1)计算Turbo码编码网格图上分支的路径度量值：

$$M_k(E)=\begin{cases}\Lambda_a(u_k)-\max(0,\Lambda_a(u_k))+\frac{1}{2}\times L_c y^s_k+\frac{1}{2}\times L_c y^p_k(2c^p_k-1), & u_k=1\\ -\max(0,\Lambda_a(u_k))-\frac{1}{2}\times L_c y^s_k+\frac{1}{2}\times L_c y^p_k(2c^p_k-1), & u_k=0\end{cases} \quad (1)$$

式中，$L_c=4\sqrt{E_s}/N_0$ 为信道置信值。

(2) 联合式(1)，计算前向度量 $A_k(s)$ 和反向状态度量 $B_k(s)$：

$$A_k(s)=\max_{e,S^s(e)}[A_{k-1}(s^E_k(s))+M_k(E)],\ k=1,2,\cdots,N-1 \quad (2)$$

$$B_k(s)=\max_{e,S^s(e)}[B_{k+1}(s^E_{k+1}(s))+M_{k+1}(E)],\ k=1,2,\cdots,N-1 \quad (3)$$

(3) 计算对数似然比和外信息对数似然比：

$$\Lambda_k(u,o)=\Lambda_a(u_k)+\Lambda_k(c^s,I)+\max_{e,u(e)=1}\left[\frac{1}{2}\Lambda_k(c^p,I)(2c^p_k-1)+A_{k-1}(s^S_k(e))+B_k(s^E_k(e))\right]-\max_{e,u(e)=1}\left[\frac{1}{2}\Lambda_k(c^p,I)(2c^p_k-1)+A_{k-1}(s^S_k(e))+B_k(s^E_k(e))\right] \quad (4)$$

$$\Lambda_e(u_k)=\Lambda_k(u,o)-\Lambda_a(u_k)-\Lambda_k(c^s,I)=\max_{e,u(e)=1}\left[\frac{1}{2}\Lambda_k(c^p,I)(2c^p_k-1)+A_{k-1}(s^S_k(e))+B_k(s^E_k(e))\right]-\max_{e,u(e)=0}\left[\frac{1}{2}\Lambda_k(c^p,I)(2c^p_k-1)+A_{k-1}(s^S_k(e))+B_k(s^E_k(e))\right] \quad (5)$$

式(5)中，$\Lambda_k(c^s,I)$ 和 $\Lambda_k(c^p,I)$ 是系统位和校验位对应的系统信息，计算如下：

$$\Lambda_k(c^s,I)=\ln\frac{p(c^s_k=1,I)}{p(c^s_k=0,I)}=L_c y^s_k \quad (6)$$

$$\Lambda_k(c^p,I)=\ln\frac{p(c^p_k=1,I)}{p(c^p_k=0,I)}=L_c y^p_k \quad (7)$$

由于Lc值对译码性能影响不大[6]，为了方便定点实现，本文中简化为Lc=1。

2.5 SISO模块的实现

分量译码器的FPGA实现的SISO模块采用模块化设计，主要包括前向度量计算模块、反向度量计算及对数似然比计算模块、前向度量存储器以及归一化度量存储器。由于前向度量计算和反向度量计算均需要计算分支度量，因此可以预先计算并存储分支度量。但在本方案中，为了节省存储空间，并没有对分支度量进行存储，而是在前向与反向度量计算时均计算一次，而且在反向度量计算收敛后同时计算对数似然比。

用FPGA对算法进行定点实现时，需要考虑到溢出的问题。为防止计算过程中出现溢出，对前向度量和反向度量计算过程进行归一化处理。若某时刻的归一化度量值选择当前时刻前向度量中的最大值，则归一化便是前向度量和反向度量减去此最大值。归一化后的前向度量和反向度量计算公式如下：

$$A_k(s)=\max_{e,S^s(e)}[A_{k-1}(s^E_k(s))+M_k(E)]-\max_{s_k}\left[\max_{e,S^s(e)}[A_{k-1}(s^E_k(s))+M_k(E)]\right],\ k=1,2,\cdots,N-1 \quad (8)$$

$$B_k(s)=\max_{e,S^s(e)}[B_{k+1}(s^E_{k+1}(s))+M_{k+1}(E)]-\max_{s_k}\left[\max_{e,S^s(e)}[A_k(s^E_k(s))+M_{k+1}(E)]\right],\ k=1,2,\cdots,N-1 \quad (9)$$

SISO模块内部处理流程分为初始化、前向度量计算和存储、反向度量计算和对数似然值计算三个部分，且对应于状态机的三个状态INIT、FSM和RSM。SISO模块的内部时序如图4所示。INIT状态完成内部寄存器的初始化设置，当外部输入信号Siso_start有效时，启动SISO模块，进入FSM状态；FSM状态中，每8个时钟周期内，用式(1)和式(2)计算出一个时刻对应的8个前向度量值，并选择出其中的最大前向度量值作为归一化度量值，用式(8)计算归一化后的前向度量值。启动一次前向度量写信号，存储当前计算得到的8个前向度量值和当前归一化度量值。当所有前向度量计算完毕时，启动Fsmrdy信号，进入RSM状态；每10个时钟周期内，用式(1)和式(2)计算出一个时刻对应的8个反向度量值，用式(9)计算归一化后的反向度量值，用式(4)和式(5)计算出相应时刻的对数似然比和外信息对数似然比，并将外信息对数似然比存储起来。当所有计算都完成时，启动Rsmrdy信号，进入INIT状态。

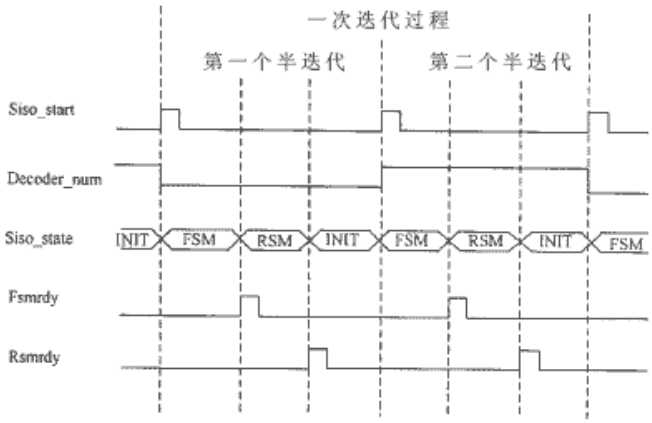

图 4 SISO 内部一次迭代过程的时序图

由于本方案中SISO模块将时分复用作为两个分量译码器，对应于一次译码迭代的两个半迭代过程。因此图4中的Decoder_num为低时，SISO模块作为第一个分量译码器，进行第一个半迭代运算；Decoder_num为高时，SISO模块作为第二个分量译码器，进行第二个半迭代运算。每次半迭代产生的对数似然比信息作为下次半迭代的先验信息。用两块RAM存储两次半迭代产生的外信息对数似然比。第一个半迭代时，从第二个外信息存储器中读取上一次半迭代产生的外信息对数似然比作为先验信息，计算得到外信息对数似然比后存储到第一个外信息存储器中；第二个半迭代时，从第一个外信息存储器中读取上一次半迭代产生的外信息对数似然比作为先验信息，计算得到外信息对数似然比后存储到第二个外信息存储器中。每帧数据译码的第一次迭代中的第一个半迭代的先验信息设为0。

迭代满足迭代终止准则后，译码器停止迭代，由信息的对数似然比值硬判决输出译码结果。工程中常用的迭代终止准则是设置最大迭代次数。最大迭代次数的设定需要综合考虑误码率性能和系统吞吐量性能。

3 Turbo码编译码器的性能

基于以上提出的Turbo码编译码器的FPGA实现方案，本文在Xilinx公司的Virtex2系列的XC2V500-6fg256 FPGA芯片上，实现了帧长在64～1 024范围之间可变的Turbo编译码器。输入数据4bit量化，内部数据位宽选择12bit，编码器模块和译码器模块在同一块FPGA芯片上实现。综合后时钟最小周期为7.188ns ，对应最高时钟频率为139.121MHz，所占的资源如表2所示。

表 2 编译码器综合后资源表

内部模块名	使用数量(个)	总数(个)	占有率(%)
逻辑片	2273	3072	73
触发器	1192	6144	19
4 输入 LUT	4281	6144	69
块 RAM	18	32	56
全局时钟	1	16	6

延迟与吞吐量是衡量译码器性能的两个主要指标。延迟定义为从第一个数据输入到第一个数据输出间的时间差。吞吐量定义为平均每秒能处理的数据量。在帧长为1 024、迭代次数为5的条件下，译码器延时约为1.4ms，吞吐量约为0.72Mbps。

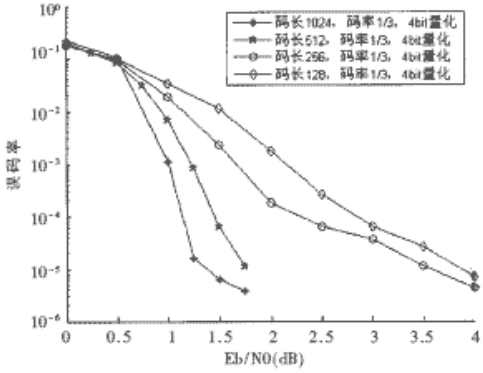

图 5 Turbo 码译码器在不同帧长条件下的误码率测试结果

最后，对帧长为128、256、512和1 024四种条件的Turbo码译码器进行了误码率性能测试。测试系统中加入高斯白噪声，数据采用 BPSK调制，译码器5次迭代。测试结果的性能曲线如图5所示。测试结果表明，在信噪比低于4dB的条件下，跳频数传通信系统采用Turbo编译码方案，误码率小于10-5，达到了数据传输可靠性的要求。由于译码器的帧长在64～1 024范围内可变，因此非常适合应用在突发数据通信中的差错控制中。

参考文献

1 Berrou C, Glavieux A, Thitimajshima P. Near shannon limit error-correcting codeing and decoding: turbo codes. in Proc.ICC′93, Geneva, Switzerland, May. 1993:1064～1070

2 Berrou C. Near optimum error correcting coding and decoding-turbo-codes. IEEE Transcations On Communications, 1996;44(10)

3 万 蕾. Turbo码及其在第三代移动通信系统中的应用. 北京理工大学博士学位论文，2001

4 Robertson P, Villebrun E, Hoeher P. A comparison of optimal and suboptimal MAP decoding algorithms operation in the log domain. in Proc.ICC ' 95,Seattle,WA,June 1995：1009～1013
5 刘东华. Turbo码原理与应用技术. 北京：电子工业出版社，2004
6 Worm A, hoeher P, When N. Turbo-decoding without SNR estimation. IEEE Commmun,2000;(4):193～195

在线联系

添加到收藏夹

关于“突发通信中Turbo码的FPGA实现”，我有如下需求或意向：

用户名： 密码： 验证码： 5829 欢迎注册 提 交

相关应用

地址：北京海淀区清华东路25号电子六所大厦
联系电话：82306084 / 82306085 传真：62311179 京ICP备05053646号
推荐分辨率1024*768 IE6.0版本