

基于 VHDL 语言的卷积码和 Viterbi 译码的实现

陶杰, 王欣, 张天辉

(军械工程学院 光学与电子工程系, 河北 石家庄 050003)

摘要: 介绍并用 VHDL 语言实现了卷积编码和维特比译码。根据编码器特征设计了一种具有针对性的简洁的维特比译码器结构, 并通过 ModelSim 平台验证了该设计的正确性。

关键词: VHDL 卷积码; 维特比译码; ModelSim

中图分类号: TN927

文献标识码: A

文章编号: 1674-7720(2012)16-0003-03

Based on the VHDL of convolutional code and Viterbi decoding implementation

Tao Jie, Wang Xin, Zhang Tianhui

(Optical and Electronic Engineering Department of the Ordnance Engineering College, Shijiazhuang 050003, China)

Abstract: This paper introduced and implemented convolutional code and Viterbi decoding with VHDL. A simple targeted Viterbi decoder structure was given according to the encoder characteristics and the design was validated through the ModelSim platform.

Key words: VHDL; convolutional code; Viterbi decoding; ModelSim

卷积码自 1955 年由爱里斯(Elias)提出以来, 因其良好的纠错能力, 已经在现代通信系统中得到了广泛应用, 比如, 电力系统通信、卫星通信、移动通信等。Viterbi 译码是 1967 年维特比(Viterbi)基于 Viterbi 算法提出的, 这种译码算法是基于码的网格(trellis)图基础之上的一种最大似然译码算法, 是一种最佳的概率译码算法^[1]。本设计是在认真研究了卷积编码和维特比译码原理以后, 用 VHDL 语言实现了(2,1,2)卷积编码, 并根据这种编码特性简化了相应的维特比译码器结构。

1 卷积编码

卷积码是一种有限记忆系统, 它与分组码类似, 也是先将信息序列分隔成长度为 k 的一个个分组; 不同的是, 某时刻的编码输出不仅取决于本时刻的分组, 而且取决于本时刻以前的 L 个分组^[2], 称 $L+1$ 为约束长度。一般情况下卷积码写成 (n, k, L) 的形式, 其中: n 表示编码器有 n 位编码输出, k 表示编码器有 k 位输入。如果用 R 表示卷积编码的效率, 则 $R=k/n$ 。

图 1 是 (n, k, L) 卷积编码器的一般结构图^[2]。从该图可以看出, 串行信号可由串/并转换器转换成 L 个分组, 并按位存储到 $k-1$ 个存储器中, 存储器中的每一个单元

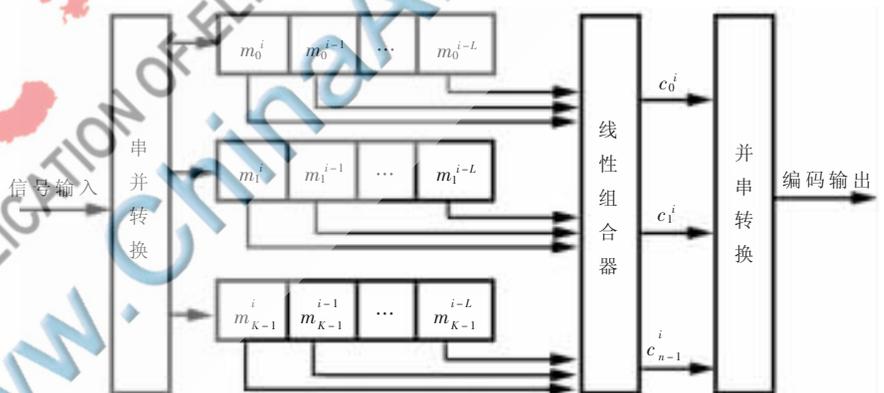


图 1 卷积编码器一般结构图

与线性组合器的连线表示该单元数据参与了线性组合计算, 但实际上是否参与取决于线性组合系数。在二进制系统中, 该系数为“0”和“1”, 当系数为“0”时, 表示该单元数据没参与线性组合计算; 当系数为“1”时, 表示该单元数据参与了线性组合计算。并/串转换器可将线性组合器计算得到的并行结果转化成串行输出。

图 1 示意的是卷积编码器的一般结构, 实际应用时可根据需求进行相应改变。例如图 2 是(2,1,2)码的编码器, 由于只有一位输入, 串/并转换器自然不存在, 线性组合器就是两个模 2 加法器, 存储单元由两个移位寄存器组成, 其生成多项式矩阵可表示为: $G(D)=[1+D+D^2, 1+D^2]$ 。

《微型机与应用》2012 年 第 31 卷 第 16 期

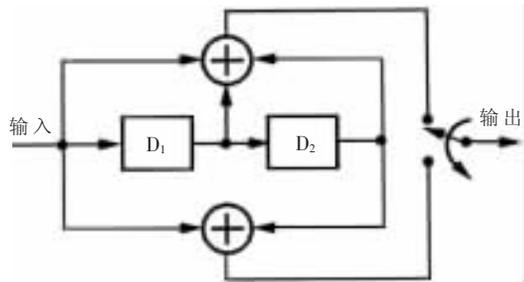


图2 (2,1,2) 卷积码编码器

图3是图2所示卷积码的状态转移图,根据这个状态转移图用VHDL语言实现卷积编码。图3中 S_0 、 S_1 、 S_2 、 S_3 表示该卷积码的4个不同状态,编程时用状态机完成状态之间的跳变和编码输出。

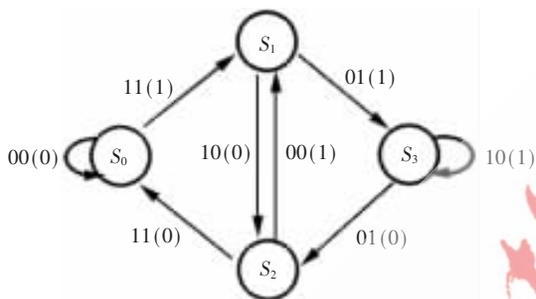


图3 卷积码的状态转移图

部分程序如下:

```

.....
if(clk1='1') then
    state <=next_state;
    .....
else
    state <=state;
    .....
case state is
when "00" =>
    if(datain = '0') then
        next_state <= "00";
        enc_out <= "00";
    else
        next_state <= "10";
        enc_out <= "11";
    end if;
when "01" =>
    .....

```

其中,clk1为编码时钟(另一时钟是码输入时钟,为clk1的2倍频),state为本次编码状态,并根据它的值和数据输入(datain)的值给出卷积码的下一个状态值(next_state)和编码输出值(enc_out)。如此往复循环下去,便可得到卷积码。

2 维特比译码

维特比译码算法是一种最大似然译码算法。根据维特比译码算法的译码步骤,译码算法的实现一般包括:

《微型机与应用》2012年 第31卷 第16期

加比选模块、度量查找表、状态变量存储器等模块,如图4所示。本设计根据该卷积码的特性,找到一种简便的改进译码方法。



图4 一般译码器结构

通过图3可知,到达每一状态的输入码均相同。例如,无论是由 S_0 转变成 S_1 还是由 S_2 转变成 S_1 ,虽然编码输出段不一样,但是码输入都是'1',其他状态之间的转变也是如此。因而,通过这个特征可以去掉该卷积码的维特比译码实现时的度量查找表和路径存储器等模块,只保留加比选模块即可,如图5所示,从而简化了维特比译码算法的实现。再结合该卷积码的状态转移图,画出相应篱笆图,如图6所示。



图5 译码器结构框图

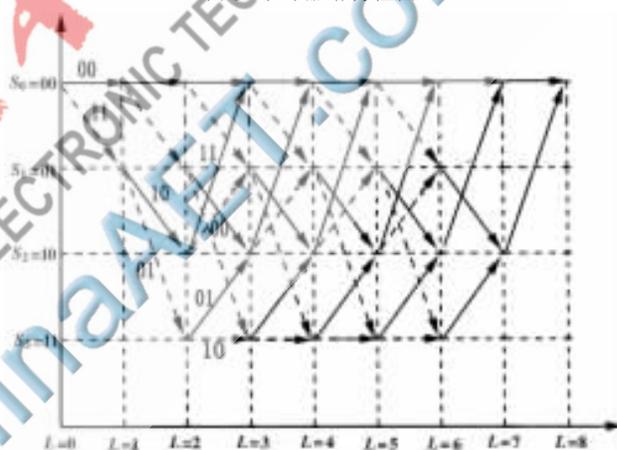


图6 篱笆图

在这个译码器结构中,“加”是指计算在该时刻实际编码输入与图3中状态转移编码输出之间的汉明距,“比”是指比较上述汉明距最小距离,“选”是指通过“比”的结果选择一位最佳码字输出。按照这一结构,在本时刻即可得到一位最佳输出码字,例如,在 $L=0$ 时刻时,实际编码输入若为“01”或“10”,则可以通过判断 $L=1$ 时刻的实际编码输入来选择一位最佳码字输出。本设计运用VHDL语言编程实现这一译码过程时,采用了两个时钟,在一个clk周期内将16bit卷积码输入到译码器内,然后再由其8倍频时钟clk1完成“加”、“比”、“选”、译码输出等动作。由于信号只能采用阻塞赋值,因而,“加”与“比”、“选”分别在两个clk1时钟周期内完成,并且“加”先于后两者1个时钟周期。

3 仿真结果

3.1 卷积编码仿真结果

依据“卷积编码”单元分析,运用VHDL语言在Model-Sim下实现的图2编码器仿真结果,如图7所示,当输入码分别

欢迎网上投稿 www.pcachina.com 5

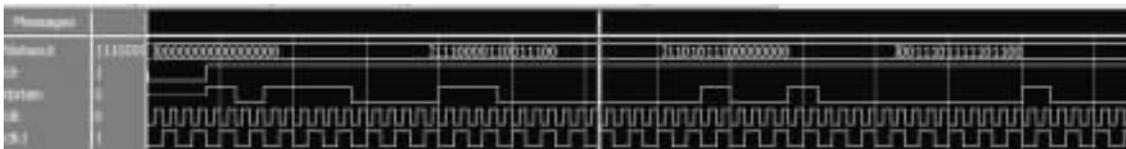


图7 (2,1,2)卷积码编码输出波形

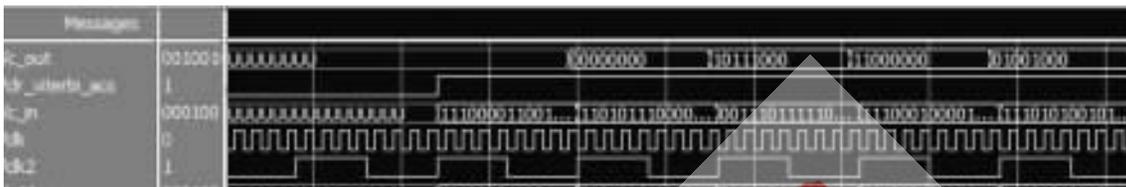


图8 (2,1,2)卷积码的维特比译码输出波形

为:10111000、11000000、01001000 时,对应输出码序列:1110000110011100、1101011100000000、0011101111101100。可以通过输入码和生成多项式矩阵来验证此卷积编码是正确有效的。图7中 dataouta 为编码输出端,clr 为编码器清零端,datain 为串行码输入端,clk 为码输入时钟,clk1 为编码时钟。

3.2 维特比译码仿真结果

依据“维特比译码”单元分析,运用 VHDL 语言在 ModelSim 下实现的图5译码器仿真结果如图8所示,当输入码分别为:1110000110011100、1101011100000000、0011101 111101100 时,对应输出码序列为:10111000、11000000、01001000。

图8中 c_out 为译码输出端,clr_viterbi_acs 为译码器清零端,c_in 为卷积码并行输入端,clk 为译码处理时

钟,clk2 为并行码输入时钟。

本文利用 VHDL 语言在 ModelSim 平台上实现与验证了(2,1,2)卷积编码和维特比译码,并根据给出的编码器的特征简化了译码器结构,从而减小了用 VHDL 语言编程的复杂度,同时也提高了程序的健壮性。该设计可以为同类型采用卷积编码和维特比译码的方案提供一定的帮助。

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,2001.
 - [2] 曹雪虹.信息论与编码[M].北京:清华大学出版社,2004.
- (收稿日期:2012-03-28)

作者简介:

陶杰,男,1983年生,硕士,主要研究方向:信息与通信工程专业。