文章编号:1004 - 5694(2001)03 - 0061 - 04

维特比算法在 C55x DSP 中的实现*

蒋国荣,郑建宏
(重庆邮电学院　移动通信工程研究中心,　重庆 400065)

摘　要:讨论了维特比算法(VA)及其改进的软输出维特比算法(SOVA)在 C55x DSP 中的实现问题,介绍了 C55x 的功能特点,并阐明它们在维特比算法三个主要步骤的运用。定量分析了维特比译码算法对 C55x DSP 的 MIPS 资源的消耗,为有关开发提供了参考依据。

关键词:维特比算法;C55x DSP;实现
中图分类号:TN911.72　　文献标识码:A

Implementation of Viterbi Algorithm on C55x DSP

JIAN Guo-rong, ZHENG Jian-hong
(*Chongqing Mobile Communications Research Center, Chongqing University of Posts and Telecommunications, Chongqing* 400065, *China*)

Abstract: This paper describes the implementation of Viterbi algorithm and its modified version-Soft Output Viterbi Algorithm on C55x DSP. Specific features of C55x are introduced and their use is demonstrated in the three main steps of Viterbi algorithm. Also a quantitative analysis of the C55x MIPS consumption of Viterbi algorithm is provided as a reference of exploitation.

Key words: Viterbi algorithm; C55x DSP; implementation

0　引　言

维特比算法(VA)是在 1967 年作为一种高效的最大似然卷积译码算法而提出的,现被广泛应用于均衡、多用户检测、模式识别等诸多领域。本质上,维特比算法是对一阶马尔可夫过程的状态序列进行最大似然估计的算法。1989 年,Hagenauert 和 Hoeher 对维特比算法作了改进[1],得到一种可输出比特判决可靠度的算法,称为软输出维特比算法(SOVA)。SOVA 可进行迭代,作为次优迭代译码算法用于 Turbo 译码。由于维特比算法的通用性,市面上有许多专用芯片可用。但人们在开发产品时更加注重其可编程性和可升级性,这种灵活性所带来的好处显而易见。因此,用 DSP 实现维特比算法具有很大的实用意义。

本文讨论维特比算法和 SOVA 在 C55x DSP 中的实现问题。首先介绍 C55x 的功能特点,接着从卷积译码的角度分别阐述维特比算法和 SOVA 的实现问题,然后估算维特比算法对 C55x 资源的消耗,最后得出相关结论。

1　C55x 功能特点

C55x 是德州仪器公司(TI™)新一代定点 DSP C5000 系列的代表。它对 C54x 的体系结构进行了

* 收稿日期:2001 - 05 - 01
作者简介:蒋国荣(1974-),男,江西丰城人,重庆邮电学院硕士研究生,研究方向为第三代移动通信技术。

改进和增强，源代码与 C54x 完全兼容，目前版本的 C55x 工作频率可达 160 MHz 左右。C55x 最显著特点是超低功耗，可以预见它有广阔的应用前景。

从 CPU 体系结构方面来看，C55x 具有以下主要特点：①32×16 bit 指令缓冲队列；②双 MAC 和双 ALU 单元；③4 个 40 位累加器；④数据与地址总线多达 12 条；⑤灵活的节能配置。

增强的结构设计使得 C55x 具有并行执行功能，这一强大功能可大大节省程序执行周期，尤其对于一些运算密集的应用效果更明显。在实现维特比算法这一特定应用方面，C55x 有精心的设计：它扩增了专门用于 VA 的指令 maxdiff，使得计算每个蝶形图单元的所需时钟周期仅为 3 个(C54x 需 4 周期)，同时兼顾到 SOVA 算法的要求，这种改进是令人鼓舞的；转移寄存器增加到 2 个，分别为 TRN0 和 TRN1，简化了维特比算法中回溯的实现；暂存器 T_i 多达 4 个，可减少分支路径度量的搬运存储。

2　维特比算法实现

任何有限状态马尔可夫过程可由网格图来描述，为简化实现，通常将网格图分解成基本组成单元——蝶形图，如图 1 所示。图 1 给出了 $1/n$ 码率的蝶形图，图 1 的左边为起始节点，分别标出状态 $2i$ 和状态 $2i+1$；右边为目的节点，分别标状态 i 和状态 $i+S/2$，这里 S 为总状态数，图中分支上还标出了分支度量，网格图的各阶段由 $S/2$ 个这样的蝶形图组成。

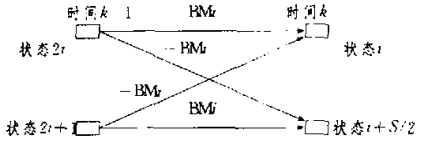

图 1　网格图的基本单元——蝶形图

Fig. 1 Butterfly chart

在实际应用中，卷积编码方案几乎都是采用 $1/n$ 码率，因为它的蝶形图中状态节点只引出两条分支路径，便于快速实现译码。要得到其他更高码率时，通过对该码率进行打孔即可。

维特比算法是根据带噪观测值来确定网格图的最大似然路径，实现过程分为 3 部分：分支度量计算，状态度量更新和回溯。

2.1　分支度量计算

计算分支度量是将各比特度量相加，而比特度量由最大似然函数推导出。至于如何将各比特度量相加，及根据最大度量值还是最小度量值来选择幸存路径，则取决于译码器输入是硬判决还是软判决。当输入硬判决时，分支度量就是汉明距离，且根据最小路径度量来选择幸存路径。软判决输入优于硬判决，因为它提供额外的可靠性信息。软判决输入的分支度量是输入符号(一个输入符号定义为 n 个量化值组成的向量)与分支输出之间的欧氏距离经简化后得到，计算方法如下：

$$BM_i=\sum_{i=0}^{n-1}SD_i\times(2X_i-1) \qquad (1)$$

其中，n 为码率倒数，SD_i 为输入符号的量化分量，$X_i\in\{0,1\}$ 为分支输出。此时，根据最大路径度量来选择幸存路径。给定输入符号，式(1)只有 2^n 种可能的值。由蝶形图的对称性，可知每个蝶形图 4 条分支的分支度量绝对值相同，且与某状态节点相连的 2 条分支，其度量符号正好相反。利用这一特点，在每级的所有蝶形图进行度量更新时，只需计算 2^{n-1} 个分支度量，从而大大降低分支度量的计算量。下面是 C55x 分支度量计算程序：

```
;1/2 码率分支度量计算
mov  *ar0+, T3        ;T3←SD0,1
add  *ar0, T3, T2     ;T2=SD0,1 + SD2,3=(+ +)
sub  *ar0+, T3, T3    ;T3=SD2,3 - SD0,1=(+ -)
```

分支度量直接保存在 $T2$ 和 $T3$ 寄存器当中。由于 C55x 只提供 1 个这样的寄存器，因此当 $n>3$ 时，须开辟内存空间来存储。

2.2　状态度量更新

状态度量更新按照蝶形图依次进行，一次更新 2 个状态，如图 2 所示。这种度量更新实际上是一个加比选(ACS)的过程，即先计算出汇入某状态的 2 条分支的分支度量，在此基础上，将 2 个分支度量分别与其前面的状态度量相加，比较 2 个相加的结果，选择其较大者存储起来作为该状态新的度量值。度量更新是维特比算法运算量最大的部分，占整个算法运算量的 90%以上。因此，这部分处理是否得当，直接决定维特比算法的运算量。

C55x 针对维特比算法这一特定应用进行了精

心设计，使得每个蝶形图的状态度量更新，包括存储更新后的状态度量在内，只需3个周期。对于SOVA，则需增加一条存储度量差的指令。下面是蝶形图的更新例程：

```
;蝶形图度量更新例程
addsub    Tx, *ar5+,ac0
subadd    Tx, *ar5+,ac1
maxdiff   ac0,ac1,ac2,ac3
|| mov    ac2, *ar7+, *ar6+
mov       ac3, *ar5-, *ar4+;这条指令仅用于SOVA
```

其中，Tx 为存储分支度量，ar5是蝶形图起始状态度量指针。addsub和subadd是用来进行加比选当中的累加操作，比较选择操作由maxdiff指令来完成，整个更新例程操作如图2所示。

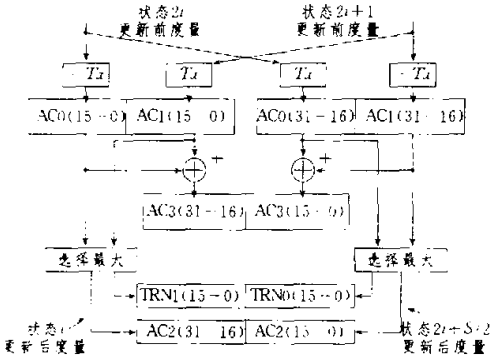

图2 蝶形图更新操作示意图

Fig. 2 The renewed butterfly operating diagram

虚线部分的累加器AC3记录每状态节点的2条竞争路径度量之差，这部分专门用于SOVA，有关SOVA的实现将在下节讨论。累加器AC2的高位字和低位字分别存放更新后的状态度量。用于TRN0和TRN1是C55x提供的2个16位专门记录幸存分支路径的寄存器，记录操作隐含在maxdiff指令中。由于每次比较操作只向2个寄存器分别存入1比特，经过16个蝶形图后，TRN0和TRN1将存满，此时要及时将它们的内容存储到状态转移表中，以免被冲刷。在保存幸存路径度量时，利用C55x的并行功能，节省了执行周期。

值得注意的是，这里并行保存的是上一蝶形图的更新结果。由于每一阶段蝶形图更新是连续进行，所以这不会产生问题。度量更新要求配置2块大小为 S（状态数）的状态缓冲区，分别存储更新前与更新后的状态度量。完成某阶段所有蝶形图的更新后，两缓冲区交换着使用。

2.3 回溯

回溯是由后往前从状态转移表中重构信息比特的过程。连续译码时，为减少时延和节省内存，回溯在处理完网格图若干阶段后进行。由卷积码的性质，N_d 阶段后，S 条幸存路径将汇合到同一起始节点，一般取 $N_d=6*K_c$。这个共同起始节点之前的各节点均属于最大似然路径。为避免过于频繁地回溯，可每次释放8或16比特[2]，此时的 N_d 不能取得太小，否则影响译码性能。

由于C55x配备2个TRN0和TRN1转移寄存器，实现回溯比较方便。但具体实现有赖于状态转移表的布置。进行回溯的伪码如下：

```
;根据路径度量最大原则，找出起始状态节点
repeat(Nd)
{
  ;直接从状态节点提取输出比特;
  ;确定状态节点在状态转移表中所对应的转移比特位置;
  ;根据转移比特，更新状态节点。
}
```

当译码器的输入数据块不长（如 $L<300$）时，通常是在处理完网格图所有阶段后，进行一次性回溯。编码时网格图一般收尾于零状态，所以回溯从零状态开始进行。

3 SOVA实现

为了产生反映比特可靠度的软输出，对于网格图每一阶段的各状态，SOVA算法不仅要求记录幸存分支，而且要求记录2条竞争路径的度量差作为本次判决可靠度的度量。在C55x中，这些操作都由maxdiff指令一并完成。

SOVA首先执行标准维特比算法，确定最大似然路径，并输出信息比特 X。在此基础上确定对应的软输出，确定软输出方法是：对于沿最大似然路径的每一状态节点，沿该节点所丢弃的竞争路径进行回溯，得到另一信息比特序列 X'，逐位比较 X' 与最大

似然路径相对应的输出信息比特，若相同则不更新对应位置的软输出；若不同则更新，更新方法是将对应软输出与状态节点的绝对度量差相比较，取较小值。

同样根据卷积码性质，这种比较和更新在回溯长度 N_d 上进行。SOVA 的回溯程序可在标准 VA 回溯例程的基础上稍作修改得到。首先，沿节点所丢弃的竞争路径回溯可按标准 VA 回溯例程进行；其次，增补对该状态节点 2 条竞争路径上的输出信息比特的比较，以及增补对该状态节点绝对度量差与对应信息比特原先的软输出的比较。依此，本次回溯可刷新前面 N_d 个信息比特的软输出。

从 C55x 实现角度看，标准维特比算法与 SOVA 的主要不同在于回溯这一环节。

4　MIPS 估算

基于以上讨论，现在估算对于码率 $1/n$ 和约束长度 Kc(状态数 $S=2^{Kc-1}$)的编码方案，标准维特比算法对 C55x 资源的要求，假定输入块长度 L，块速率为 R_B。

每阶段所需周期数＝分支度量计算＋蝶形图运算＋转移比特存储＋指针操作＝$f(n)+3\times S/2+S/16+2$，回溯所需周期数 $\approx 12\times L$.

$f(n)$ 为分支度量所需周期数，$n=2$ 时，$f(n)=3$；$n=3$ 时，$f(n)=9$；$n=6$ 时，分支度量存取时需访问内存，$f(n)\approx 60$。

MIPS 为：

$$\{\text{每阶段所需周期数}\times L+\text{回溯所需周期数}\}\times R_B\approx\{[f(n)+3\times S/2+S/16+2]\times L+12\times L\}\times R_B \quad (2)$$

式(2)只考虑一次性回溯的情况，且没有包括初始化等开销。连续译码时，要进行若干次回溯，此时所需周期可根据参数 N_d 和回溯次数来考虑。根据式(2)，表 1 给出在 TD-SCDMA 系统部分信道的维特比译码对 C55x MIPS 消耗的估算。

表 1　TD-SCDMA 部分信道的维特比译码对 C55x MIPS 消耗估算

Tab. 1 Estimation of C55x MIPS loss by viterbi decodes of TD-SCDMA channels

信道类型	块长度 (L)	块速率 (R_B)	约束长度 (K)	码率 ($1/R$)	MIPS
TCH-T	198	50	9	1/3	4.2
TCH-T/9.6	248	50	9	1/3	6.6
TCH-T/14.4	298	50	9	1/2	6.2
SACCH-T	208	25/6	9	1/2	0.37
BCCH-T	209	200	9	1/3	17.7

5　结　论

本文探讨了用 C55x 实现 VA 及 SOVA 的问题。VA 最关键部分是加比选操作，C55x 由于其并行执行的特点，可达到 3 周期/蝶形图，这对于 $Kc\leqslant 9$ 卷积码 C55x 能轻松胜任。值得指出，现在可用的 TI C55x 版本可信的性能约为 120 MIPS。对于要求多码道同时工作 TD-SCDMA 系统，依照第 4 节的分析并充分考虑其他的任务(如上行编码等)，C55x 只能对约 16 个码道的数据进行实时译码，因此，对于 3 G 应用来说它尚存在一定局限。

参　考　文　献

[1] HAGLNAUER J, HOEHER P. A Viterbi algorithm with soft-decision outputs and its applicatlons [C]. in Proc IEEE GLOBECOM, 1989, 1680-1686.

[2] WILSON STEPHEN G. Digital and Coding (chap. 6)[M]. Pentice Hall, 1996.

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

[4] Texas Instruments, TMS320C55x Mnemonic Instruction Set Reference Guide, 2000[Z].

[5] Texas Instruments, TMS320C55x Programmer's Guide(Preliminary Draft), 2000[Z].

[6] SIEMENS, AEK3002A V1.0, TD-SCDMA Physical Layer, 2000[S].

(编辑：郭继笃)