低功耗、低复杂度TURBO码实现研究

姜小波，陈 杰，仇玉林

(中科院微电子所 北京 朝阳区 100029)

【摘要】提出了一种改进的TURBO码算法，该算法改进了传统算法的路径度量的运算和可靠度的运算，可以显著地减少TURBO解码器使用的硬件资源，存储器的使用是传统算法的一半。根据改进的算法，该文还提出了一种新的加比选运算单元，综合结果和功耗分析显示，新的ACS单元面积和功耗分别是传统ACS单元的32.7%和26.8%。

关 键 词 TURBO码; 软输出维特比算法; 低功耗; 加比选
中图分类号 TN764; TP331.2 文献标识码 A

An Implementation of Low Power and Low Complexity TURBO

JIANG Xiao-bo，CHEN Jie，QIU Yu-lin

(Institute of Microelectronics of Chinese Academy of Science Chaoyang Beijing 100029)

Abstract A modified TURBO algorithm is proposed with efficient operation. This algorithm remarkably reduces the amount of hardware of the TURBO decoder, where the scale of decoder memories can be reduced to the half memories required by conventional algorithms. Based on the modified TURBO algorithm, a new Add-Compare-Select (ACS) unit is designed. The simulation results show that the area and power consumption of proposed ACS unit are 32.7% and 26.8% of conventional ACS units, respectively.

Key words TURBO; soft output viterbi algorithm; low power; add-compare-select

文献[1]提出了TURBO码，可以达到接近商农限的性能。TURBO码的优异性能吸引了人们的广泛关注。TURBO解码一般采用两种算法：最大后验概率算法(MAP)和软输出维特比算法(Soft Output Viterbi Algorithm，SOVA)。其中MAP算法性能好一些，但复杂度较高。SOVA算法性能差一些，但复杂度低。目前有报导，SOVA算法的性能已经提高到和MAP算法接近的程度。虽然有关于SOVA实现的研究[2]，但主要是在量化精度方面。SOVA算法还需要存储路径度量差值(称为可靠度，表示幸存路径的可靠度)，对可靠度进行量化在集成电路实现上是不现实的。因为计算可靠度是在关键路径上，对可靠度进行量化增加了关键路径的长度，造成性能的恶化。基于此，本文改进了传统的TURBO码解码算法。

1 SOVA算法

1.1 软输出VA算法

软输出维特比算法1989年提出[3]。SOVA算法和VA算法大致相同，但它的输入多了一个先验概率，SOVA算法还要对每一个估计值给出一个概率值。算法步骤如下：

(1) 设置初始值：t=0；S_0=0；$S_m[n]$=0。(2) 时间增加1：计算分支度量。和VA算法有所区别，SOVA算法的分支度量计算考虑了先验信息。根据公式，对于每一个节点，计算时刻t进入同一个节点的路径度量。对于每一个节点，比较t时刻内进入同一个节点的两条路径，存储路径最小的那条。对于每一个节点，计算t时刻内进入同一个节点的两条路径的差值(可靠度)，并存储。重复(2)直到时刻$t=\tau$。(3) 回退，获得信息序列的估计值。(4) 更新可靠度。

从算法上比较SOVA和VA的不同之处：(1) SOVA要计算存储进入同一个节点路径度量的差值(可靠度)；(2) SOVA算法要进行可靠度的更新。可靠度的更新在SOVA算法中非常重要，计算和存储可靠度增加了SOVA

收稿日期：2004 – 02 – 23
基金项目：国家863计划基金资助项目(2002AA1Z1130)
作者简介：姜小波(1972 –)，男，博士生，主要从事数字信号处理器结构和信道解码器集成电路实现方面的研究.

算法的复杂度和存储器的开销。

1.2 改进的软输出**VA**算法

本文对算法做了改进：TURBO码一般用在帧长很长的应用中，导致路径度量很大，从而增加了运算单元的复杂度和功耗。改进的算法对路径度量的值进行了限制，主要改进了算法的第(2)步。

路径度量计算：如果路径度量小于一个门限值，则不改变路径度量。如果路径度量大于门限值，则对所有状态的路径度量做除2的运算。可靠度计算：如果可靠度小于一个门限值，则可靠度不改变。如果可靠度大于门限值，则令可靠度等于门限值。

1.3 模拟仿真结果

改进的路径度量的算法对VA算法没有影响，因为对所有路径度量除2没有改变路径度量的大小关系，但对可靠度有影响，从而对SOVA算法有影响。改进的路径度量的算法模拟仿真结果如图1所示。其中红星线是传统算法的仿真结果，取不同的门限值对性能产生不同的影响。门限值越小，对性能的影响越小。当门限值为128时，如图中菱形线所示，改进的算法和传统的算法性能仅相差0.2 dB左右。仿真用的TURBO码采用3 G标准中使用的8状态TURBO码，帧长为1 024 bit。

模拟仿真发现，在设置门限值时，TURBO解码的性能有所退化。这一结果和前面的分析吻合。改进可靠度运算后，模拟仿真结果表明，阈值大于32时，TURBO解码器的性能没有什么影响；当阈值设为16时，TURBO解码器性能出现严重退化。模拟仿真结果如图2所示。模拟采用8状态的TURBO码，帧长是640 bit。

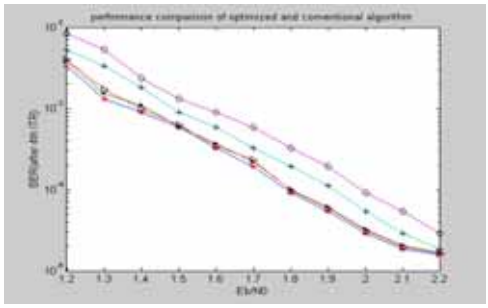

图1 改进路径度量模拟仿真结果

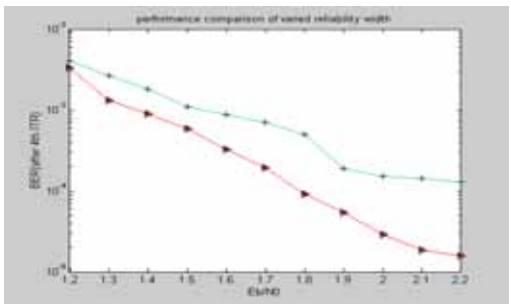

图2 改进可靠度的模拟仿真结果

2 TURBO码实现分析

2.1 可靠度位宽选择

SOVA算法中，可靠度的存储占用了大量的存储器。所以，可靠度的位宽是SOVA解码器设计要考虑的一个重要参数。一般可靠度的位宽选择根据文献[4]，它的最大值为：

$$R_{\text{Reliability}}=P_{\text{PMMAX}}-P_{\text{PMMIN}}=(K-1)(B_{\text{BMMAX}}-B_{\text{BMMIN}})$$

式中 K是约束长度；B是分支度量。3GPP标准中，K是4，软判决一般采用3 bit量化。这样给出未加外部信息时的可靠度范围$R_{\text{Reliability}}<3\times(16-(-16))=96$。TURBO解码时，分支度量还要考虑先验信息，则：$R_{\text{Reliability}}<3\times(16+48-(-16-48))=384$。根据这个规则，TURBO码的可靠度要选择位宽为9的存储器。假设信息帧长为1 024 bit，则TURBO解码器存储可靠度占的存储器为1 024×9×8×2＝144 Kb的存储器。其中8是TURBO的状态数，每个状态都要存储一个可靠度值；2是TURBO解码器用的SOVA解码器数量。而根据模拟仿真结果，可靠度存储器的位宽可设为5。TURBO解码器存储可靠度占的存储器为：1 024×5×8×2＝80 Kb，是原来的56%。

2.2 运算单元设计

SOVA解码器的主要运算单元是加比选单元(Add-Compare-Select Unit，ACSU)。SOVA解码器的ACSU由两个加法器，一个比较器，一个减法器构成。其中，加法器的位宽是一个重要的考虑因素。3GPP标准中，最大的帧长是5 Kb，则路径度量的范围是$p_{\text{pathmetric}}<5\text{ K}\times16=80\text{ K}$。要满足这个要求，加法器必须采用18位的加法器。

有文献对路径度量采用规范化的方法来限制路径度量的范围，这种方法是每次运算后，8个路径度量都

减去其中最小的路径度量，这种方法可以限制路径度量的范围。每次运算都要通过比较器选择最小的路径度量，增加了功耗，降低了电路的性能。

传统的ACS单元如图3所示，由两个18位的加法器，一个减法器和一个选择器构成。根据改进的算法，提出了一种新的ACS单元，由两个8位的加法器，一个8位的减法器，一个8位的移位器，两个选择器，少量控制单元构成。如图4所示。该ACS单元和传统ACS单元相比，多了一个8位的移位器和一个选择器。用来在检测到溢出时对路径度量除2。控制单元检测ACS单元是否溢出。SOVA解码器由8个ACS单元构成，任何一个ACS单元溢出，就对所有的路径度量除2。新ACS单元进行了RTL仿真和综合，并进行了功耗分析，并与传统的ACS单元进行了对比。

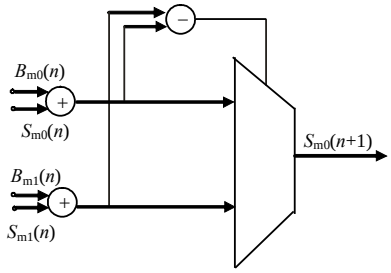

图3　传统ACS单元

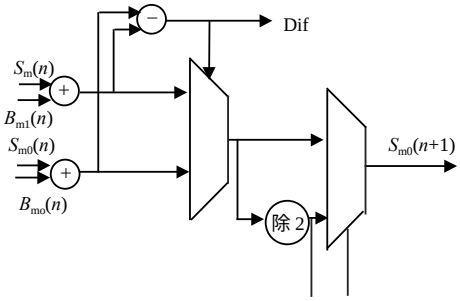

图4　改进算法ACSU功能单元

虽然新ACS单元和传统ACS单元相比，多了一级规范化的运算。但由于运算单元的位宽减少了10 bit，性能和功耗优于传统ACS单元。综合和功耗分析结果如表1、2所示。

表1　ACS单元综合结果

Clock/ns	Conv/gates	Modi/gates	比值/(%)
5	918	300	32.7
10	306	170	55.6

表2　ACS单元功耗分析结果

Clock/ns	Conv/mW	Modi/mW	比值/(%)
5	23.5	6.3	26.8
10	4.5	2.4	53.3

表1中给出了传统ACS单元和新ACS单元综合结果。综合是用Synopsys综合工具，采用versilicon为中芯国际做的0.18工艺的库。可以发现，在100 MHz情况下，新ACS单元面积是传统ACS单元的55.6%。200 MHz情况下，新ACS单元面积是传统ACS单元的32.7%。新ACS单元节省了大量的面积。性能越高，节省的面积越大。表2给出了传统ACS单元和新ACS单元功耗分析结果。100 MHz情况下，新ACS单元功耗是传统ACS单元的53.3%。200 MHz情况下，新ACS单元功耗是传统ACS单元的26.8%。新ACS单元节省了大量的功耗。而且性能越高，节省的功耗越大。

3　结　论

本文改进了传统的TURBO码解码算法并实现了改进算法的ACSU单元。仿真结果表明，改进的TURBO解码算法性能和未改进的仅相差0.2 dB。

参　考　文　献

[1] Berrou C, Glavieusx A, Thitimajshima P. Near shannon limit error-correcting coding and decoding: turbo codes[C] //In Proc. IEEE Int. Conf. Communications, Geneva, Switzerland, 1993: 1 064-1 070.

[2] Joeressen O J, Vaupel M , Meyr H. Soft-output viterbi decoding: VLSI implemention issues[C]//In Vehicular Technology conference, Secaucus, NJ, USA, 1993: 941-944.

[3] Hagenauer J, Hoeher P. A viterbi algorithm with soft-decision outputs and its applications[C] //In Proc. IEEE GLOBECON. Dallas. TX, 1989: 47.1.1-47.1.7.

[4] Heller J A, Jacobs I M. Viterbi decoding for satellite and space communications[J]. IEEE Transactions on Communication Technology, 1971, COM19(5): 835-848.

编　辑　漆　蓉