

硬件友好的 3GPP-LTE Turbo 交织器设计^①

姚彦斌^②* ** ** 周一青* ** 林江南* ** ** 萧 放* **

(* 中国科学院计算技术研究所无线通信技术研究 中心 北京 100190)

(** 北京市移动计算与新型终端重点实验室 北京 100190)

(*** 中国科学院大学 北京 100049)

摘 要 研究了第三代合作伙伴计划长期演进技术(3GPP-LTE)中的二次置换多项式(QPP)交织器的硬件设计优化,提出了一种零延时、低复杂度的 QPP 交织器设计方案。该方案从算法层面出发简化了 QPP 交织器中定义的复杂运算。得益于算法优化的结果,优化后的 QPP 交织器能够以较低的代价映射到硬件电路上。结果表明:与一些传统方案相比,该方案设计的 QPP 交织器大大降低了硬件实现的复杂度,在 SMIC 40nm 工艺下,交织器的面积只有0.040mm²。另外,所设计的 QPP 交织器具有零时延的特点,能够有效提高 Turbo 译码的译码效率。基于该交织器所设计的 Turbo 译码器能够稳定工作在 400MHz 下,译码速率达到572.85Mbps(10 次半迭代),而译码器面积仅有0.82mm²。

关键词 长期演进(LTE), Turbo, 二次置换多项式(QPP), 交织器, ASIC 设计, 零延时, 低复杂度

0 引 言

Turbo 码由于能够提供接近于香农限的信道传输容量,因而被广泛应用于各类无线通信协议中作为前向纠错(forward error correction, FEC)的主要方案,例如高速数据分组接入(high speed data packet access, HSDPA)、长期演进(long term evolution, LTE)等^[1-3]技术。Turbo 译码器以一种迭代的方式来进行译码,逻辑上包括两个软输入软输出(soft-input soft-output, SISO)译码器,一个按照序列的自然顺序进行译码,另一个以交织后的顺序进行译码。一个 SISO 译码的过程被称为一次半迭代。在两次半迭代之间,外信息通过交织/解交织之后,从一个 SISO 传递到另一个 SISO,实现信息在两个 SISO 之

间的交互。在这里,为了最小化相邻比特之间的相关性而引入交织器。交织器可以以一个伪随机的方式来打乱比特序列的顺序,从而达到减小相邻比特相关性的目的,其中某些交织器的设计较为复杂,这对 Turbo 译码器的整体设计提出了极高的挑战,需要设计复杂的硬件电路来实现。

为了缓解交织器带来的瓶颈,LTE 中定义了一种二次置换多项式(quadratic permutation polynomial, QPP),即 QPP 交织器。该交织器具有两个特点:一是该交织器是冲突避免的,意味着从 N 个并行 SISO 计算出来的 N 个外信息在解交织之后,总能够被存入到 N 个不同的存储器中;二是该交织器是最大向量化的,意味着 N 个并行 SISO 所需要访问的 N 个交织地址总是指向 N 个存储器的同一个位置。基于以上两个特性,研究人员提出了不少交

① 国家自然科学基金(61431001)和北京市青年拔尖人才(2015000021223ZK31)资助项目。
② 男,1987 年生,博士生;研究方向:面向矢量 DSP 的通信信号处理算法,大规模通信硬件加速器架构研究;联系人,E-mail: yaoyanbin@ict.ac.cn
(收稿日期:2016-09-07)

交织器的设计方案。从实现层面来看,交织器的设计主要集中于交织算法与交织网络的实现上,目前大部分交织设计方案都集中于后者的设计上^[4-10]。对于交织算法的实现,绝大部分方案都采用了 QPP 交织器的递推公式来设计硬件电路^[5-8]。在交织算法的优化方面,文献[9]提出了一种基于置换模式的优化交织器方案,该方案可利用一个统一的交织硬件电路来计算所有并行 SISO 的交织地址。其交织电路的设计虽然较优,但由于需要保存不同配置下的初始交织模式,因而总体的硬件复杂度较高。

实际上,在 QPP 交织器中,导致硬件复杂度高的最主要因素是 QPP 交织器中涉及到的取余运算。该运算在硬件实现中是用除法器来实现,特别在高阶基底、多 SISO 并行的配置下需要多个并行除法器,硬件代价较大。本文从算法层面出发,调研了 QPP 交织器的内在结构。通过在算法层面将交织器中的取余运算简化,其简化的结果能够以一个较低的代价映射到硬件电路上。得益于简化的交织地址生成方法,交织器的硬件复杂度很低。在 40nm 工艺节点下,其面积只有 0.040mm²,优于传统的 QPP 交织器设计方案。另外,交织地址的计算与顺序地址的计算相比,不存在相对延时,能够保证 Turbo 译码的译码速率。结果表明:Turbo 译码器能够稳定工作在 400MHz 下,在 10 次半迭代下,译码速率能够达到 572.85Mbps,而译码器面积仅有 0.82mm²。

1 QPP 交织器的递推计算公式

LTE 定义的二次置换多项式(QPP)交织器用式

$$\pi(i) = (f_1 i + f_2 i^2) \bmod K \quad (1)$$

表示,其中, i 和 $\pi(i)$ 分别代表自然序列的下标与交织后的序列下标。参数 f_1 和 f_2 与码块长度 K 有关,需要巧妙进行设计来确保满足 QPP 交织器的性质。在 LTE 中,总共设计了 188 种码块长度,其长度从 40 到 6144。

式(1)中存在取余运算且被除数数值较大,在实现中硬件复杂度大。一般而言,交织地址的产生可以通过如下所示的递推公式来生成^[11]:

$$\pi(i+1) = [\pi(i) + \delta(i)] \bmod K \quad (2)$$

$$\delta(i+1) = [\delta(i) + b] \bmod K \quad (3)$$

其中初始值为 $\pi(0) = 0, \delta(0) = f_1 + f_2$ 。另外, b 为一个常量,即 $b = 2f_2$ 。由于该递推公式中只涉及到简单的加法、取余运算,交织地址的计算可以相对有效地在硬件上实现。

为了应对 LTE 高速译码的需求,Turbo 译码器普遍采用了高阶蝶型运算来提高译码器的译码吞吐率。高阶蝶形运算可以在一个时钟周期内进行多级状态度量的计算,也就需要交织器在一个时钟周期内提供多个交织地址。一般地,如果将 $\pi(i+k)$ 代入到式(1)中,并进行化简,容易得到递推公式更为一般性的结果:

$$\pi(i+k) = [\pi(i) + \delta(i)] \bmod K \quad (4)$$

$$\delta(i+k) = [\delta(i) + b] \bmod K \quad (5)$$

其中初始值为 $\delta(0) = [f_1 \cdot k + f_2 \cdot k^2] \bmod K, \pi(0) = 0$, 另外, $b = [2f_2 k] \bmod K$ 。

2 硬件友好的 QPP 交织器

QPP 交织器的递推公式在一定程度上缓解了原交织器实现硬件复杂度高的问题。然而在多个 SISO 并行的高速 Turbo 译码架构中,仍然存在不少问题:一是递推公式中存在取余运算,在硬件实现中,需要除法器来进行计算。有多少个并行的 SISO 译码器就需要多少个除法器,这会导致较大的硬件开销。二是我们必须为每个并行的 SISO 提供其交织地址递推计算的初始值,该初始值只能够用式(1)计算,这也涉及到取余运算。为了降低交织器的复杂度,使其能够以一个较低的代价映射到硬件电路上去,其核心问题在于化简交织器中的取余运算。为了便于描述本文所提出的交织器算法,本节在式(2)和(3)的基础上进行说明。

2.1 交织地址递推公式的简化

在式(2)中,因为第一个加法项小于等于 K 第二个加法项小于 K ,因此两个加法项相加的结果一定小于 $2K$,即

$$\because \pi(i) \leq K, \delta(i) < K$$

$$\therefore \pi(i) + \delta(i) < 2K$$

因此,式(2)中的取余运算完全可以用比较、选

择等简单运算来进行替代

$$\pi(i+1) = \begin{cases} \pi(i) + \delta(i) & \pi(i) + \delta(i) < K \\ \pi(i) + \delta(i) - K & \text{其他} \end{cases} \quad (6)$$

同理可得

$$\delta(i+1) = \begin{cases} \delta(i) + b & \delta(i) + b < K \\ \delta(i) + b - K & \text{其他} \end{cases} \quad (7)$$

利用式(6)和(7)来递推计算交织地址,从根本上避免了取余运算。在硬件实现中,特别是多个 SISO 并行时,可以大大降低硬件实现复杂度。

2.2 并行 SISO 的初始地址产生

通过上一节的简化,交织地址可以通过递推的方式以较低的硬件复杂度来产生。然而,我们仍需要为每个并行的 SISO 确定其初始的交织地址。如果直接通过式(1)来计算得到,那在硬件设计时仍然需要除法器

一般地,由于码块长度是确定的,传统的做法是直接在一个查找表(look-up table, LUT)中保存所有并行 SISO 的初始交织地址。然而,LTE 中设计了共计 188 种不同的码长,而并行 SISO 的数量又可能有 1/2/4/8 等四种配置。如果要将所有不同配置下的并行 SISO 初始交织地址都保存下来,则需要存储一个较大的 LUT,这会导致硬件复杂度的增大。

注意到并行 SISO 的初始交织地址可以表示为 $p \cdot L$, 其中 $p \in \{0,1,\cdots,P-1\}$ 为并行 SISO 的标号, P 为 SISO 的个数 $P \in \{1,2,4,8\}$, L 为单个并行 SISO 的译码长度 $L = K/P$ 。首先可以将式(1)变换为

$$\pi_{\text{init}} = [(f_1 + f_2 i) \times i] \bmod K$$

将 $i = p \cdot L$ 和 $K = P \cdot L$ 代入可得

$$\pi_{\text{init}} = [(f_1 + f_2 pL) \times p \cdot L] \bmod (P \cdot L)$$

在上式中,由于进行取余运算的两项具有公因子 L ,可将其提出得

$$\pi_{\text{init}} = \{[(f_1 + f_2 pL) \times p] \bmod P\} \times L \quad (8)$$

经过以上变换后,对码长 K 的取余运算简化为了对并行 SISO 个数 P 的取余运算。为了进一步进行化简,首先引入一个定理:

定理 1:对于 $A \bmod B$,如果 B 等于 2 的幂次,即 $B = 2^n$,则在硬件实现中,取余运算可以简化为 $A \& (B - 1)$ 。其中运算“&”代表比特按位“与”。

证明:见文献[12]。

由于在 QPP 交织器中并行 SISO 的个数 P 的取值限定为 2 的幂次,故根据定理 1 可得,式(6)中的取余运算可在硬件实现时利用简单的比特按位“与”运算来实现,也即

$$\pi_{\text{init}} = \{[(f_1 + f_2 pL) \times p] \& (P - 1)\} \times L \quad (9)$$

经过以上变换之后,QPP 交织器中的取余运算从算法层面被替换为了若干简单运算。通过一系列的化简,为交织算法高效地映射到硬件电路上提供了极大的便利。

3 零延时的 Turbo 交织器架构设计

3.1 整体架构设计

图 1 所示为 Turbo 并行 SISO 的译码架构示意图。从解调器得到的三路软比特信息分别被存放到系统信息存储阵列、第一和第二校验信息存储阵列中,其中每个存储阵列由 P 个存储器构成,可以以 P

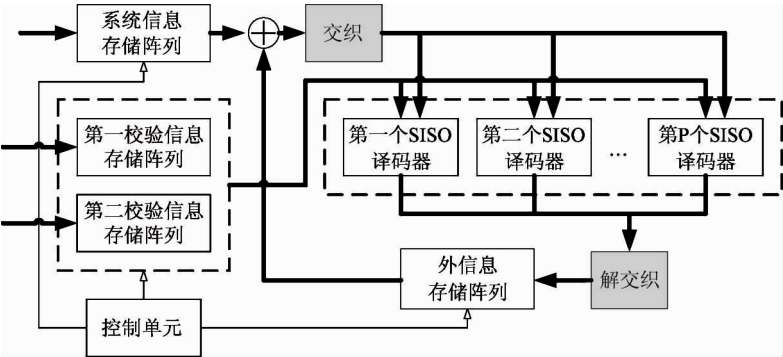


图 1 并行 SISO 译码架构示意图

的并行度来对阵列中的数据进行同时访问。在译码过程中,首先,将系统信息与外信息相加,之后与相应的校验信息一起送入 P 个并行的 SISO 译码器。并行的 SISO 译码器组处理输入的数据,并将输出的外信息存储到外信息存储单元中,完成一次半迭代的运算。其中,在顺序译码阶段(偶数次半迭代)系统信息与外信息累加之后的结果不经过交织单元,而是直接输入给并行 SISO 译码器。而译码器的输出也不经过解交织模块,直接存入到外信息存储阵列中。

需要指出的是,针对交织器的调用,存在两种调

度策略:非均衡的调度策略与均衡的调度策略,如图 2 所示。在非均衡的调度策略中,外信息存储阵列按照信息序列的自然序来进行存储。在偶数次半迭代中,不进行交织与解交织;而在奇数次半迭代中外信息与系统信息相加的结果进行交织,而译码之后的外信息经过解交织存储到外信息存储阵列中。在均衡的调度策略中,偶数次半迭代只进行外信息的解交织;而奇数次半迭代进行外信息的交织。外信息存储阵列按照半迭代次数的不同分别按照自然序或者交织序来存储外信息。

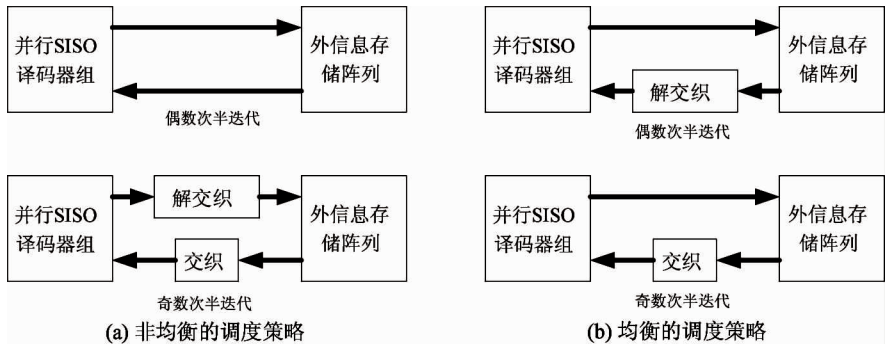


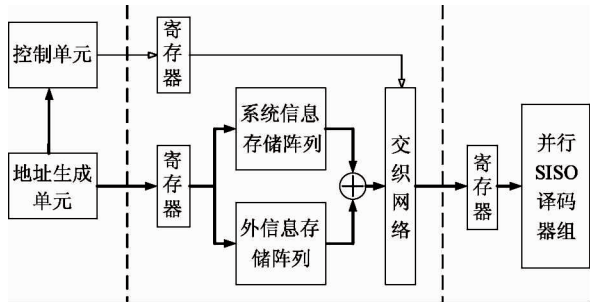
图 2 调度策略

交织器的均衡调度策略可以使偶数次和奇数次半迭代过程中处理负荷较为均衡,解决非均衡的调度策略中奇数次半迭代负荷较重的问题。在本文中,由于所设计的交织器具有零时延的特点,相对自然序的译码而言,交织序的译码不存在相对延时,因此也不存在调度策略的问题,故直接采用非均衡的调度策略较为明晰(如图 1 所示)。

3.2 零延时交织器设计

图 3 所示为零延时交织器以及相关模块的时序架构设计。在第一个时钟周期,地址生成单元生成系统信息、外信息存储阵列的访问地址。另外,在该时钟周期内,控制单元利用地址生成单元的结果为交织网络生成控制信号。在第二个时钟周期,经过寄存的访问地址分别送入到系统信息、外信息存储阵列。经过一定的延时之后,对应的系统信息与外信息被读出并相加。另一方面,在偶数次半迭代时,相加后的结果旁路(Bypass)掉交织网络,直接送到后一级;在奇数次半迭代时,相加后的结果送进交织

网络,在寄存后的交织网络控制信号的控制下完成交织后再送到后一级进行处理。



对于地址生成单元来说,在偶数次半迭代中生成顺序访问的地址;在奇数次半迭代生成交织序的访问地址。其中,顺序访问的地址产生电路较为简单,只需要设计一个计数器。该计数器在每个时钟周期进行自加即可,在此不再赘述。在奇数次半迭代过程中,地址生成单元中存在两个递推计算单元

用于分别对式(6)和式(7)进行计算。如图 4 所示,在上一个时钟周期计算出来的 $\pi(i)$ 和 $\delta(i)$ 首先进行加法运算。 $\pi(i)$ 和 $\delta(i)$ 相加得到的结果一方面与 K 进行比较,另一方面再减去 K 。比较的结果用于生成二选一控制器的控制信号。当 $\pi(i)$ 和 $\delta(i)$ 相加的结果大于 K 时,选择 $\pi(i) + \delta(i) - K$ 的结果作为 $\pi(i + 1)$ 的结果;反之选择 $\pi(i) + \delta(i)$ 作为 $\pi(i + 1)$ 的结果。递推计算能够控制在一个时钟周期内,与顺序的地址产生相比,不存在相对延时。注意在高阶蝶形的 Turbo 译码架构中,需要多个地址生成单元并行计算,例如在面向基 4 蝶型的交织地址生成单元中,存在两个交织地址递推计算单元,其中一个地址递推计算单元计算偶地址,另外一个计算奇地址,如图 5 所示。

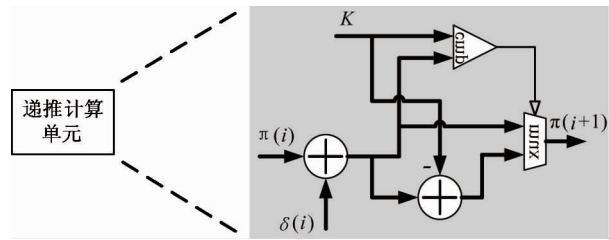


图 4 递推计算单元示意图

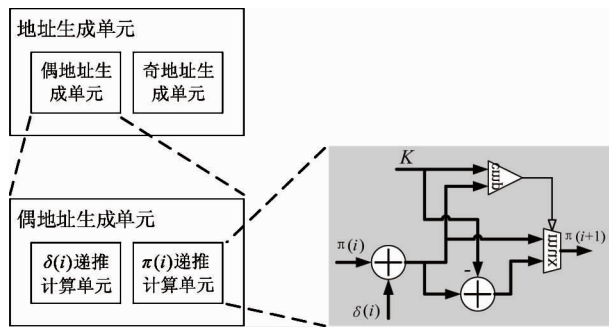


图 5 基 4 蝶型架构的交织地址生成单元示意图

经过上述设计后,不论是顺序译码阶段(偶数次半迭代)还是交织序译码阶段(奇数次半迭代),其处理的时延都是一致的。与顺序译码相比。交织译码处理的不同点在于:一是地址生成单元的生成地址不同;二是在交织译码阶段,数据需要经过交织网络。在交织译码阶段,虽然交织地址的生成相比顺序地址生成较为复杂,但是由于从算法层面优化了交织地址的计算算法,算法映射到硬件电路上后

只需要两个简单的递推计算单元就能完成,大大简化了硬件电路设计规模。而在交织阶段,虽然需要经过一个交织网络,但从后文的实现层面来看,不会成为系统时序优化的瓶颈。

4 实现结果分析

为了验证本文所设计的零延时交织器的性能高低,我们在 SMIC 的 40nm 工艺下实现了包含所提出的零延时交织器的 Turbo 译码器,并从时序情况和硬件复杂度两个方面来对交织器进行分析。

4.1 交织器的时序分析

图 6 所示为交织器的地址生成单元及其相关电路的时序情况。上一节中已经提到:对于基 4 的 Turbo 译码来说,交织地址的生成单元中存在 4 个并行的递推计算单元,其中最大延时为 0.95ns。两个递推计算单元计算出的 $\pi(i)$ 送到交织网络控制信号生成单元,经过 0.71ns 之后得到控制信号。因此,该部分组合逻辑的最长路径为 1.66ns。

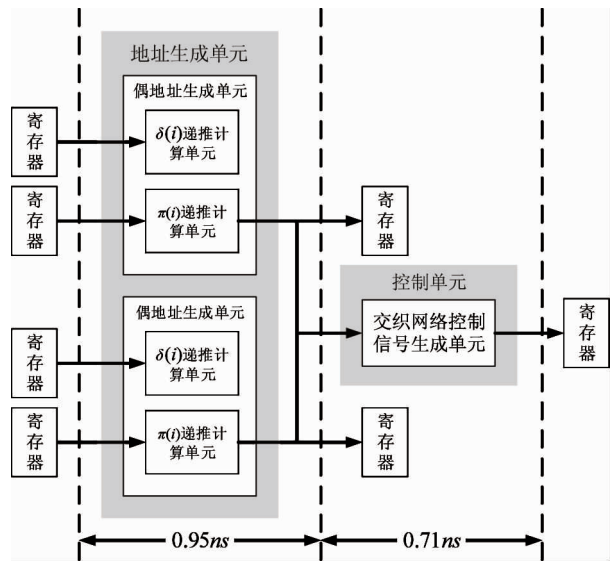


图 6 交织器地址生成单元及相应电路的时序分析结果

图 7 所示为交织网络以及相关电路的时序情况。经过寄存的地址首先送到系统信息与外信息的存储阵列,经过 1.21ns 的延时之后,存储阵列的输出数据有效。两个存储阵列对应的系统信息与外信息相加,其中加法延时为 0.32ns。在控制信号的控制

下,加法的结果送入交织网络,最后被寄存下来。数据经过交织网络的最长路径延时为0.56ns。容易看出,该部分组合逻辑的路径延时为2.09ns。

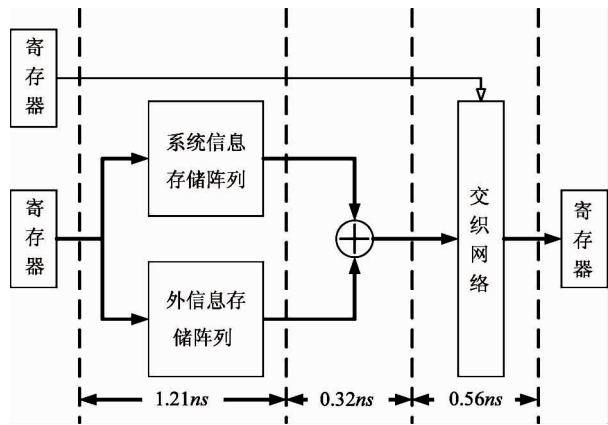


图 7 交织网络及相关电路的时序分析结果

综上可得,在确保交织译码阶段与顺序译码阶段相比不存在相对延时的情况下,交织部分的两个关键路径,一个延时为1.66ns;另一个为2.09ns。该关键路径导致的时钟最高主频的限定分别为602.4MHz和478.5MHz。综合考虑交织器的处理性能和功耗面积表现,该时延结果不足以成为系统时序优化的瓶颈。

4.2 交织器的硬件复杂度比较

为了验证所设计的交织器的硬件复杂度情况并将其与已有的技术方案进行对比,我们在 SMIC 的 40nm 工艺下实现了本文所提出的交织器以及文

献[5]和[9]中提出的交织器方案,所实现的交织器支持基 4 蝶形、最高 8 个 SISO 并行。如表 1 所示,从结果来看,本文提出的交织器具有非常小的硬件开销。交织器的面积仅占0.040mm²,与含有除法器的交织器^[5]相比,硬件复杂度仅为后者的6.1%。另外,也优于采用置换模式的交织器方案^[9]。

表 1 交织器的硬件复杂度比较

	面积(单位:mm ²)	归一化面积比
本方案	0.040	1.00
文献[5]	0.652	16.300
文献[9]	0.106	2.65

4.3 基于该交织器的 Turbo 译码器性能比较

为了进一步验证所提出的交织器对 Turbo 译码整体性能优化带来的好处,本文在 40nm 工艺下实现了基于该交织器的 Turbo 译码器,并将其与一些主流方案进行了比较。如表 2 所示,所设计的 Turbo 译码器能够稳定工作在 400MHz 主频下。在 10 次半迭代时,最大译码速率能够达到572.85Mbps;与此同时,译码器整体面积只有0.82mm²(包含存储器),而不含存储器的译码器面积仅仅为0.43mm²。综合来看,得益于在交织器上的优化,所设计的 Turbo 译码器取得了较优的硬件效率。与其它方案相比,在使用较低硬件资源的情况下,却获得了较高的译码速率。

表 2 本方案与其它 Turbo 译码器的比较

	本文方案	文献[4]	文献[5]	文献[6]	文献[7]
蝶形单元基底	radix-4	radix-2	radix-4	radix-2	radix-2
并行 SISO 个数	8	8	8	64	8
CMOS 工艺(nm)	40	90	130	65	90
面积(mm ²)	0.82	2.10	3.57	8.3	6.1
主频(MHz)	400	275	302	400	625
最大译码速率 (Mbps,10次半迭代)	572.85	130	390.6	1536.02	438.83
归一化硬件效率 (Mbps/mm ²) ^{&}	1332.21 ^{\$}	312.65	1155.35	488.29	363.33

\$ 由于其余方案在计算面积时未考虑存储器面积,这里只考虑不含存储器的面积,为 0.43mm²

& 不同工艺下的面积按照如下因子进行归一化:(40nm/65nm)²=0.379;(40nm/90nm)²=0.198;(40nm/130nm)²=0.0947

5 结 论

本文提出了一种硬件优化的二次置换多项式 (QPP) 交织器。该交织器通过从算法层面化解取余运算, 算法能够以较低的代价映射到硬件电路上, 硬件实现复杂度极低。另外, 得益于简化的硬件电路, 交织器的时序优化较为容易。所设计的交织器具有“零”延时的特点, 与顺序译码阶段相比不存在相对延时, 能够有效提高 Turbo 译码器的译码效率。

参考文献

[1] The 3rd Generation Partnership project. 3GPP Technical Specification 25.213 v9.2.0 Release 9, Universal Mobile Telecommunications System (UMTS); Spreading and modulation (FDD), http://www.etsi.org/deliver/etsi_ts/125200_125299/125213/09_02_00_60/ts_125213v090200p.pdf, 2010

[2] The 3rd Generation Partnership project. 3GPP Technical Specification 36.212 v8.8.0 Release 8, Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and channel coding, http://www.etsi.org/deliver/etsi_ts/136200_136299/136212/08_08_00_60/ts_136212v080800p.pdf, 2009

[3] Zhou Y, Wang J, Sawahashi M. Downlink transmission of broadband OFCDM systems-part III: turbo-coded. *IEEE J on Selected Areas in Commun*, 2006, 24(1): 132-140

[4] Wong C C, Chang H C. Reconfigurable turbo decoder with parallel architecture for 3GPP LTE system. *IEEE Trans on Circuits Syst II*, 2010, 57(7): 566-570

[5] Studer C, Benkeser C, Belfanti S, et al. Design and implementation of a parallel turbo decoder ASIC for 3GPP-LTE. *IEEE J of Solid-State Circuits*, 2011, 46(1): 8-16

[6] Sun Y, Cavallaro J R. Efficient hardware implementation of a highly-parallel 3GPP-LTE/LTE-advance turbo decoder. *Integration the VLSI Journal*, 2011, 44(4): 305-315

[7] Shrestha R, Paily R P. High-throughput turbo decoder with parallel architecture for LTE wireless communication standards. *IEEE Trans on Circuits Syst I*, 2014, 61(9): 2699-2710

[8] Yoo I, Kim B, Park I C. Tail-overlapped SISO decoding for high-throughput LTE-advanced turbo decoders. *IEEE Trans on Circuits Syst I*, 2014, 61(9): 2711-2720

[9] Kim B, Yoo I, Park I C. Low-complexity parallel QPP interleaver based on permutation patterns. *IEEE Trans on Circuits Syst II*, 2013, 60(3): 162-166

[10] Wang J, Zhang K, Kroll H, et al. Design of QPP interleavers for the parallel turbo decoding architecture. *IEEE Trans on Circuits Syst I*, 2016 63(2): 288-299

[11] Moision B, Hamkins J. Coded modulation for the deep-space optical channel: Serially concatenated pulse-position modulation. *Development Biology*, 2005, 283(1): 113-127

[12] Adam H. Faster division and modulo operation—the power of two. [http://blog.teamleadnet.com; teamlead-net](http://blog.teamleadnet.com/teamlead-net), 2012

Design of a hardware-friendly turbo interleaver for 3GPP-LTE

Yao Yanbin * * * * *, Zhou Yiqing * * *, Lin Jiangnan * * * * *, Xiao Fang * * *

(* Wireless Communication Research Center, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190)

(** Beijing Key Laboratory of Mobile Computing and Pervasive Device, Beijing 100190)

(*** University of Chinese Academy of Sciences, Beijing 100049)

Abstract

The optimization of the hardware design of quadratic permutation polynomial (QPP) interleavers for 3GPP-LTE was studied, and a scheme for design of zero-delay, low-complexity QPP interleavers was presented. The design scheme can simplify the complex computations defined by OPP interleavers at the algorithm level, and it makes an optimized OPP interleaver be readily mapped onto its hardware circuit under a low cost because of the benefit from the optimization. The implementation result shows that the proposed scheme can greatly reduce the complexity of hardware implementation compared to traditional approaches, and the area of the designed interleaver is only 0.040mm^2 under the technology of CMOS 40nm. Furthermore, the designed interleaver reveals its zero-delay property, and the ability to effectively improve the efficiency of Turbo decoding. Actually, after putting the whole Turbo decoder into practice, the decoder can work at 400 MHz, the area of which is only 0.82mm^2 while the peak decoding throughput can reach 572.85Mbps with 10 half-iterations.

Key words: long term evolution (LTE), Turbo, quadratic permutation polynomial (QPP), interleaver, ASIC design, zero delay, low complexity