

# 一种基于异构多核 DSP 的 IEEE 802.11a 接收端基带处理的研究和实现\*

徐力, 王沁, 史少波

(北京科技大学计算机与通信工程学院, 北京 100083)

**摘要:** 现有基于异构多核 DSP 的 IEEE 802.11a 接收端实现方法中 DSP 核空闲等待时间较长, 不能充分体现多核 DSP 的高性能计算能力。结合多核 DSP 的特点, 通过核内细粒度流水和核间粗粒度流水的方法, 来提高多核 DSP 的执行效率, 并在目标异构多核 DSP 上实现完整的 IEEE 802.11a 接收端基带处理。实验结果表明, 该方法不仅能满足系统吞吐量和实时性, 与类似工作相比还能保证较高的 DSP 核平均利用率。

**关键词:** IEEE 802.11a; 异构多核 DSP; 软件无线电; 接收端基带处理

**中图分类号:** TP393      **文献标志码:** A      **文章编号:** 1001-3695(2012)01-0241-05

doi:10.3969/j.issn.1001-3695.2012.01.067

## Research and implementation of IEEE 802.11a receiver baseband processing based on heterogeneous multi-core DSP

XU Li, WANG Qin, SHI Shao-bo

(School of Computer & Communication Engineering, University of Science & Technology Beijing, Beijing 100083, China)

**Abstract:** Nowadays, the DSP core waiting time is longer in some IEEE 802.11a implementations based on multi-core DSP, which can not take full advantage of its high-performance computational capabilities. This paper combined the characteristics of multi-core DSP, using coarse-grained pipeline between DSP cores and fine-grained pipeline in single DSP core to enhance execution efficiency of multi-core DSP. It implemented on target heterogeneous multi-core DSP with a complete IEEE 802.11a receiver baseband processing. The experiment show that this method not only guarantee the system throughput and real-time, but also improved the DSP core average utilization compared similar works.

**Key words:** IEEE 802.11a; heterogeneous multi-core DSP; software-defined radio; receiver baseband processing

## 0 引言

近年来随着人们对无线通信需求的不断增加, 无线通信协议的升级和换代变得越来越频繁。为了能有效地兼容现有的各种协议并能灵活地实现新的协议, 软件无线电 (software-defined radio) 平台逐渐受到人们的高度关注。软件无线电平台一般采用可编程处理器来实现无线通信中的基带数字信号处理, 通过软件的更新来灵活地实现各种协议。而 IEEE 802.11a<sup>[1]</sup> 协议作为无线通信协议的一种, 常常被用做一个基准测试协议来评估软件无线电平台的性能。

文献[2, 3]在不同的软件无线电平台上设计并实现了 IEEE 802.11a 发送端基带处理。而与基带发送端的处理相比, IEEE 802.11a 接收端基带处理更为复杂, 在某些实现中或者不能完全实现 IEEE 802.11a 协议所规定的 54 Mbps 最高传输速率<sup>[4]</sup>, 或者在实现中忽略接收端基带处理中诸如帧同步、载波频偏估计/频偏补偿、信道均衡等重要组成部分<sup>[5]</sup>。此外, 虽然通过专用的硬件加速器可以提高软件无线电平台对某些复杂计算的处理能力, 如 Viterbi 硬件加速器, 但是在软件无

线电平台中接收端的软件实现部分却成了约束整个系统吞吐量的瓶颈。

本文根据多核 DSP 能够进行并行处理的特点, 通过接收端基带处理在多核 DSP 上核内细粒度流水和核间粗粒度流水来提高多核 DSP 的执行效率, 并最终在多核 DSP 上设计并实现了一个完整的 IEEE 802.11a 接收端基带处理。本文采用的多核 DSP 基于 VLIW 指令集体系结构, 九个异构 DSP 核通过开环互连结构连接。同时, 多核 DSP 还使用一个 Viterbi 硬件加速器来实现卷积码的解码。

## 1 IEEE 802.11a 接收端基带处理

如图 1 所示, IEEE 802.11a 物理层帧包括前导码和 OFDM 符号部分。前导码部分由短训练符号 (short training symbol) 和长训练符号 (long training symbol) 组成; OFDM 符号又分为信号符号 (signal symbol) 和数据符号 (data symbol)。接收端从模/数转换器中获得发送信号的采样, 来实现 IEEE 802.11a 接收端基带处理。

接收端基带处理的过程如图 2 所示。其中主要的处理模

**收稿日期:** 2011-04-28; **修回日期:** 2011-06-03      **基金项目:** 国家“863”计划资助项目 (2008AA01Z134); 国家自然科学基金资助项目 (61172049, 61173150, 61003251); 高等学校博士点科研基金资助项目 (20100006110015); 北京市教育委员会科研基地共建项目

**作者简介:** 徐力 (1984-), 男, 博士研究生, 主要研究方向为计算机系统结构、多核处理器设计、多核 SoC (cnxl1984@gmail.com); 王沁 (1961-), 女, 教授, 博导, 博士, 主要研究方向为计算机系统结构、集成电路设计、无线传感器网络; 史少波 (1986-), 男, 博士研究生, 主要研究方向为计算机系统结构、多核处理器设计、多核 SoC。

块包括帧同步 (frame sync)、小数倍载波频偏估计 (fractional CFO)、载波频偏补偿 (CFO compensation)、整数倍载波频偏估计 (integer CFO)、去保护间隔 (guard remove)、信道估计 (channel estimation)/64 点 FFT、信道均衡 (channel equalization)、去导频 (pilots remove)、解调 (demodulation)、解交织 (de-interleave)、解增信删余 (de-puncture)、Viterbi 解码 (viterbi) 和解扰 (de-scramble) 等。

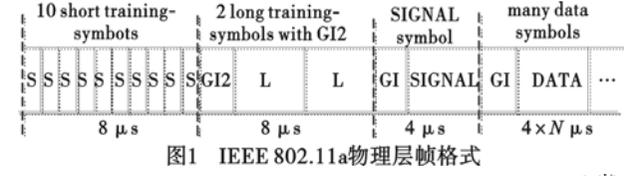


图1 IEEE 802.11a物理层帧格式

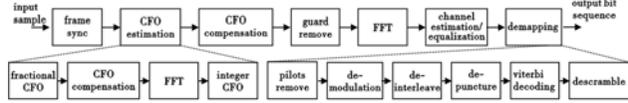


图2 IEEE 802.11a接收端基带处理的过程

1.1 帧同步

接收端帧同步的主要目的是识别并找到物理层帧的起始位置。在物理层帧格式中,10个内容相同的短训练序列符号用来实现帧同步。本文采用 Schmidl 等人<sup>[6]</sup>提出的延迟相关同步算法来实现接收端帧同步,计算式为

$$P(n) = \sum_{k=0}^{L-1} r(n+k+16) \times r^*(n+k) \quad Q(n) = \sum_{k=0}^{L-1} |r(n+k)|^2$$

$$M(n) = \frac{|P(n)|^2}{Q(n)^2} > Th_{syn} \quad (1)$$

其中: $P(n)$ 是接收端中从第 $n^{\text{th}}$ 个采样值开始的 $L$ 个连续采样值与经过 $D$ 个延时窗口后 $L$ 个连续采样值之间的自相关能量值; $Q(n)$ 是从第 $n^{\text{th}}$ 个采样值开始的 $L$ 个连续采样值的能量值;归一化系数 $M(n)$ 用来与判决门限作比较,检测一个帧是否到来。实现中把能量检测窗口 $L$ 设为16,延时窗口 $D$ 设为16。判决门限值与帧起始位置的识别率有关。帧同步判决采样点的仿真如图3所示。

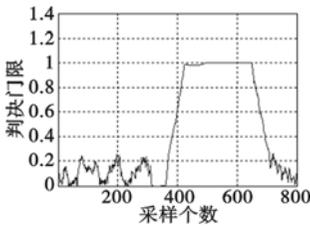


图3 帧同步判决采样点

本文选用0.75<sup>[7]</sup>作为判决门限值。如式(2)所示,在实现时一般采用等效运算判断当前的采样是否满足判决条件。

$$|P(n)|^2 > Th_{syn} \times Q(n)^2 \quad (2)$$

1.2 载波频偏估计

由于发送端和接收端的发送频率与采样频率不同,导致发送端与接收端之间存在载波频率偏移,而基于OFDM的通信系统对载波频偏相当敏感,载波频偏会破坏子载波之间的正交性,造成子载波间干扰,所以需要对于子载波频偏进行估计,然后用估计的频偏值对采样进行频偏补偿。对无频偏补偿和有频偏补偿采样的解调结果对比如图4所示。从图4可以看到,通过频偏补偿才能实现正确的解调。

载波频偏估计包括小数倍频偏估计和整数倍频偏估计两部分。小数倍频偏估计值在时域中计算得到,而整数倍频偏估计值在频域中获得。根据小数倍频偏估计值和整数倍频偏估

计值得到最终载波频偏估计值,然后用该值去补偿余下的接收端采样。载波频偏估计计算需要长训练序列符号,其结构如图5所示, $L_1$ 和 $L_2$ 内容相同, $x_1(n)$ 、 $x_2(n)$ 分别为 $L_1$ 和 $L_2$ 的一半, $GI_2$ 为循环前缀。

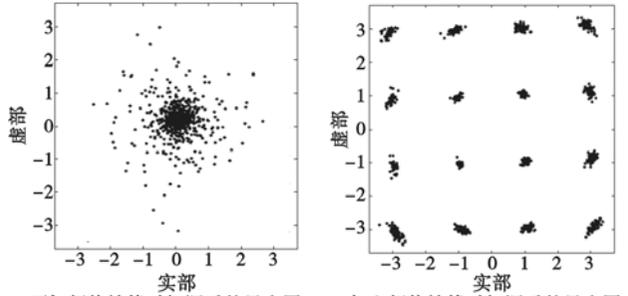


图4 16QAM调制的子载波在频域上的星座图

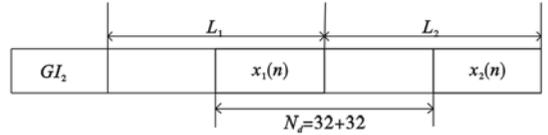


图5 IEEE 802.11a长训练序列结构

a) 小数倍频偏估计。假设在小数倍频偏中频率偏移为 $\Delta f_c$ ,相位偏移为 $\Delta \varphi$ ,那么 $x_1(n)$ 和 $x_2(n)$ 在接收端的时域表达式 $r_1(n)$ 、 $r_2(n)$ 可写为

$$r_1(n) = r(n) = x_1(n) e^{-j(2\pi\Delta f_c n T_s + \Delta \varphi)} + \eta_1(n T_s)$$

$$r_2(n) = r(n + N_d) = x_2(n) e^{-j(2\pi\Delta f_c (n + N_d) T_s + \Delta \varphi)} + \eta_2(n T_s)$$

where  $n = 0, \dots, L - 1$  (3)

其中: $T_s$ 是接收端的采样频率; $\eta_i$ 是高斯白噪声。这里定义:

$$R_i = \sum_{n=0}^{L-1} r_1(n) r_2^*(n) = \sum_{n=0}^{L-1} r(n) r^*(n + N_d) \quad (4)$$

因为 $x_1(n) = x_2(n)$ , $n = 0, 1, 2, \dots, L - 1$ ,在不考虑噪声的情况下,可以得到

$$R_i = e^{j2\pi\Delta f_c N_d T_s} \sum_{n=0}^{L-1} |x_1(n)|^2 \quad (5)$$

所以相位相关值 $\arg(R_i) = 2\pi\Delta f_c N_d T_s$ ,小数倍频率偏移值 $\Delta f_c = \arg(R_i) / (2\pi N_d T)$ 。

b) 整数倍频偏估计。对长训练序列符号进行小数倍频偏补偿后,经过64点FFT,得到长训练序列的频域值。整数倍频偏会造成频域中序列偏移,在这里采用带有频率偏移的长训练序列与原始长训练序列(在发送端定义)之间的相关值去估计整数倍频偏值。整数倍频偏估计值 $\Delta f_i$ 可以表示为

$$\Delta f_i = 64 - idx_{\max\_value} \quad (6)$$

其中: $idx_{\max\_value}$ 是最大相关值的位置。

基于整数倍和小数倍频偏估计,系统总的频偏为

$$\Delta f = \Delta f_i - \Delta f_c \quad (7)$$

实现中采用改进CORDIC算法<sup>[8]</sup>计算频偏估计中辐角的计算和进行频偏补偿。得到频偏估计值之后,需要对长训练序列符号进行频偏补偿,然后进行下一步的信道估计。而OFDM符号经过频偏补偿后,进行下一步的信道均衡处理。

1.3 信道估计

为了消除通信信道对信号的影响,IEEE 802.11a接收端需要对接收到的信号作信道均衡处理,一般采用信道估计方法得出信道系数。由于发送信号经过信道衰弱后,接收信号可以表示为

$$R(k) = H(k) \times X(k) + W(k) \quad (8)$$

其中: $W(k)$ 、 $R(k)$ 、 $H(k)$ 、 $X(k)$ 分别为高斯白噪声和第 $k$ 个子

载波的接收信号,信道系数、发送信号,因此,如果不考虑高斯白噪声,那么

$$H(k) = R(k) \times X(k)^{-1} \quad (9)$$

两个长训练符号中除去循环前缀并经过频偏补偿后,通过 64 点 FFT 转换到频域上。信道估计主要使用这两个长训练序列来实现。假设  $\hat{L}$  是接收端和发送端已知的长训练序列,  $\hat{L}_1$  和  $\hat{L}_2$  分别为接收端的两个长训练序列,  $\hat{L}$ 、 $\hat{L}_1$ 、 $\hat{L}_2$  都是信号在频域上的信息,则第  $k$  个子载波的信道系数为

$$H(k) = (\hat{L}_1(k) + \hat{L}_2(k)) / (2\hat{L}(k)) \quad (10)$$

其中:  $(\hat{L}_1(k) + \hat{L}_2(k)) / 2$  是这两个长训练序列的均值。

得到的信道系数用来对 OFDM 符号进行信道均衡。对于接收端第  $m$  个符号中第  $k$  个子载波  $S_m(k)$  的信道均衡可以表示为

$$\hat{S}_m(k) = S_m(k) / H(k) \quad (11)$$

由于在实现中复数的除法运算比较复杂,一般在信道估计中计算  $C(k)$  来替代  $H(k)$ ,如式(12)所示。

$$C(k) = 1/H(k) = 2\hat{L}(k) / (\hat{L}_1(k) + \hat{L}_2(k)) \quad (12)$$

$$\hat{S}_m(k) = S_m(k) \times C(k) \quad (13)$$

经过信道均衡之后,OFDM 符号的子载波采样被解调成比特流数据,然后这些数据通过解交织、解增信删余、Viterbi 解码以及解扰,最终将得到的数据发送给数据链路层进行进一步处理。

## 2 多核 DSP 体系结构

本文选用的 DSP 处理器为开环互连结构的异构多核 DSP。采用异构多核体系结构可以满足接收端基带处理中不同服务质量(性能、功耗)的要求,可以提高 DSP 核的平均利用率,并能进一步减少处理器功耗开销。而且开环互连结构相对于总线互连结构和 Mesh 结构来说互连转发单元(switch unit)简单,易于控制和硬件实现。同时,鉴于无线通信基带数字信号处理具有同步数据流(synchronous data flow)的特点<sup>[9]</sup>,适合用开环互连结构的多核 DSP 实现。

多核 DSP 由异构 DSP 核和共享存储器(share memory, SM)组成,如图 6 所示。DSP 核又包括 DSP 单元。私有存储器(private memory, PM)和控制单元,每个 DSP 核运行在同一时钟频率下。私有存储器用来存储程序执行时的中间数据,以减少片外存储器访问开销,进一步节约处理器核功耗。开环互连结构中转发单元的数据转发功能采用共享存储器实现,控制转发功能采用总线方式实现。共享存储器和私有存储器都是双端口的存储模块。控制单元用来判断当前总线上的控制信号是否属于当前 DSP 核。基于这种方式,核间的数据和控制可以异步传输。如图 6 所示,当 DSP 核 2 执行当前任务时, DSP 核 1 可以把发送给 DSP 核 2 的数据存入 DSP 核 1 与 DSP 核 2 之间的共享存储器中。同时 DSP 核 1 将控制信号通过总线发送给 DSP 核 2,并通知 DSP 核 2 该任务的数据已经存在共享存储器中。DSP 核 2 的控制单元判断后,将该控制信号存入一个 FIFO 单元。DSP 核 2 在完成当前任务后执行存入 FIFO 单元中控制信号所指定的任务。这样使得不同 DSP 核能更有效地执行各种任务,提高 DSP 核的平均利用率。

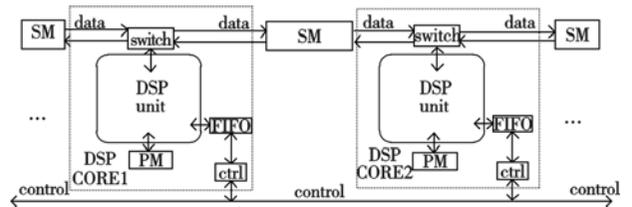


图6 多核DSP体系结构

DSP 核都是基于超长指令字(VLIW)指令集体系结构的 32 bit DSP。VLIW 指令集体系结构具有较好的指令集并行性,能有效提高 DSP 核的处理能力。如图 7 所示, VLIW 指令除了固定长度的立即数槽(immediates slot)、分支槽(branch slot)、控制槽(ctrl slot)、私有存储器控制槽(PM slot)、公有存储器控制槽(SM slot)之外,还包括数量不等的计算单元槽(CU slot)。每个计算单元槽对应于一个计算单元的具体微指令。多核 DSP 上的计算单元主要包括逻辑运算单元(ALU)、乘累加器(MAC)和移位器(SHIFTER)。采用不同数目的计算单元,会影响 VLIW 指令的长度。所以,可根据不同处理的实际需要,在实现处理前对这些计算单元进行配置(数目和种类),并形成相应 VLIW 指令集的 DSP 核。整个 VLIW 指令通过取指、译码、执行 3 级流水进行执行。在取指阶段,处理器根据 PC 值取出指令并放到指令寄存器上;译码阶段对指令寄存器上的指令进行译码,并将译码结果发送到各个 CU 和私有存储器控制槽、公有存储器控制槽;执行阶段 CU 和私有存储器、公有存储器执行相应的动作。

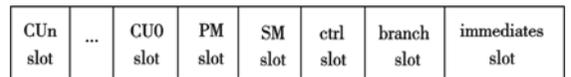


图7 多核DSP VLIW指令结构

## 3 IEEE 802.11a 接收端基带的实现

除了 Viterbi 使用硬件加速器实现外,还使用多核 DSP 上的汇编语言来设计 IEEE 802.11a 接收端基带中其他处理模块,实现一个完整的 IEEE 802.11a 接收端。IEEE 802.11a 接收端基带处理中不同处理模块可以映射到相同或者不同 DSP 核上进行计算。本文根据 VLIW 指令集体系结构和异构多核的特点,使接收端基带处理在多核 DSP 上以核内细粒度流水和核间粗粒度流水的方式执行,来保证系统的实时性和吞吐量。

### 3.1 细粒度流水

根据基带处理的特点,如下所示接收端基带中各种处理模块的算法实现中都具有一定长度的内部循环体。

接收端基带处理中处理模块的算法过程:

```

LOOP 1:N
  读入数据
  执行
  写回数据
END LOOP

```

同时,内部循环体的执行在整个模块处理中占用比例很高,而且这些内部循环体的前后迭代过程大部分是前后数据无关的。所以,当处理模块分配到多核 DSP 中的 DSP 核上时,数据可以通过 DSP 核内部细粒度流水执行,来实现处理模块高速执行。

把处理模块中的内部循环展开,将一次迭代的内部循环体称做任务,并将任务处理过程分为读入数据、执行和写回数据

三个阶段。由于基于 VLIW 指令集体系结构的处理器在一个周期内能并行执行多条微指令,可以同时进行任务的读入数据、执行和写回数据,所以对单个任务来说,任务的执行不需要所有输入数据都得到才能开始,当一个任务得到执行开始时所需的输入数据就可以开始执行。同时,也没有必要等到整个执行全部完成后才把数据写回。在执行阶段中得到某个输出结果就可以立即写回,单个任务在 DSP 核上的处理过程如图 8(a)所示。此外,处理模块中内部循环体的前后迭代过程大部分是前后数据无关。所以对相继两个任务来说,当前一个任务的读入数据阶段、执行阶段和写回数据阶段结束时,下一个任务的读入数据阶段、执行阶段和写回数据阶段就可以开始。相继两个任务在 DSP 核上的执行过程如图 8(b)所示。这样可以实现任务在 DSP 核内细粒度流水,提高模块处理时 DSP 核的执行效率。

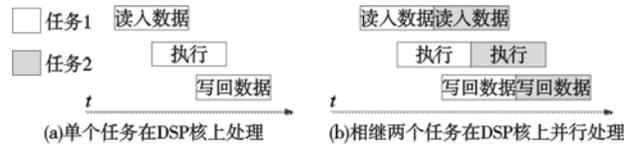


图8 任务在DSP核上的实现方式

同时,对接收端基带处理进行研究和分析之后可以看到,IEEE 802.11a 接收端处理主要包括如表 1 所示的四种不同类型任务。根据细粒度流水的实现方法,IEEE 802.11a 接收端中不同模块的执行时间以及所需 DSP 核的类型如表 2 所示。在这里只统计处理过程最为复杂的 54 Mbps 传输速率下(64QAM 调制方式、3/4 卷积码码率)IEEE 802.11a 接收端处理实现时的情况。表 1、2 中,输入/输出中 1 个采样(sample)为 2 个 32 bit 数据,1 个值(value)为 1 个 32 bit 数据,执行时间单位为周期。

表 1 IEEE 802.11a 接收端基带处理中的任务类型

功能描述	输入	输出	时间
复数乘法	4 samples	2 samples	5 cycles
4 数相加运算	4 samples	1 samples	3 cycles
基 2 蝶形中的一路运算	4 samples	2 samples	8 cycles
1 阶 CORDIC	4 samples	3 samples	8 cycles

表 2 IEEE 802.11a 接收端基带处理中不同模块的执行时间以及所需 DSP 核的类型

模块	输入	输出	时间	DSP 核
frame sync	160 samples	1 value	130 cycles	1alu + 2mac + 1shifter
fractional CFO	64 samples	1 value	146 cycles	1alu + 2mac + 1shifter
CFO compensation	64 samples	64 samples	4 323 cycles	4alu + 2mac + 2shifter
integer CFO	64 samples	64 samples	2 209 cycles	1alu + 2mac + 1shifter
FFT	64 sample	64 samples	978 cycles	2alu + 2mac + 2shifter
guard remove	64 samples	52 samples	64 cycles	/
channel estimation	52 samples	52 samples	162 cycles	1alu + 1shifter
channel equalization	52 samples	52 samples	163 cycles	1alu + 2mac + 1shifter
pilots remove	52 samples	48 samples	52 cycles	/

续表 2

demodulation	48 samples	228 bits	1 890 cycles	1alu + 1mac
de-interleave	228 bits	228 bits	308 cycles	/
de-Puncture	228 bits	432 bits	346 cycles	/
viterbi	432 bits	216 bits	2 376 cycles	/
de-Scramble	216 bits	216 bits	490 cycles	/

### 3.2 粗粒度流水

由于多核 DSP 具有多个 DSP 核,可以将接收端基带中不同处理模块映射到不同 DSP 核上并行执行,如图 9 所示。假设接收端基带中相继两个处理模块被映射到两个 DSP 核上,当数据 1 完成映射在 DSP 核 1 上的处理模块后,可以传递到 DSP 核 2 中完成下一个处理模块,而此时数据 2 可以传递到 DSP 核 1 进行处理。不同数据在 DSP 核间粗粒度流水执行,可以提高整个接收端基带处理的处理速度以及 DSP 核的平均利用率。



图9 不同数据在DSP核间粗粒度流水

虽然采用 DSP 核间粗粒度流水的实现方式可以提高系统吞吐量,但是从图 9 可以看到,系统的吞吐量取决于执行时间最长的那个 DSP 核。如果 DSP 核之间执行时间差别很大,就会导致执行时间短的 DSP 核停顿(processor stall)较长一段时间才能继续执行。假设把 fractional CFO 和 CFO compensation 分别分配到表 2 所列的两个 DSP 核(DSP 核 1 和 DSP 核 2)中,那么,DSP 核 1 和 DSP 核 2 的处理时间分别为 146 和 4 323 个周期。由于粗粒度流水的需要,DSP 核 1 停顿 4 177 个周期后才能计算完的数据发送给 DSP 核 2,以进行下一步处理及输入新的数据。为了进一步提高系统的吞吐量,本文通过处理模块的拆分来均衡粗粒度流水时各个 DSP 核的执行时间,以减少 DSP 核停顿等待时间。

由于基带处理中处理模块具有内部循环体,因此可以将内部循环体拆分产生新模块,然后再次分配到不同 DSP 核上,如下所示。

接收端处理中处理模块的拆分数法过程:

```

LOOP 1:m
    读入数据
    执行
    写回数据
END LOOP //分配到 DSP 核 1,m 的值由用户设定
LOOP m:N
    读入数据
    执行
    写回数据
END LOOP//分配到 DSP 核 2
    
```

在这里规定每个 DSP 核执行时间上限定为 3 000 个周期。由于 CFO compensation 的处理时间是 4 323 个周期,高于约定时间上限,那么,将 CFO compensation 平均拆分成两个处理模块进行处理,拆分结果如表 3 所示。把 CFO compensation step1

和 CFO compensation step2 分别分配到 DSP 核 1 和 DSP 核 2,最终 DSP 核 1 和 DSP 核 2 的处理时间变为 2 351 和 2 205 个周期,均衡了两个 DSP 核的执行时间,从而提高了粗粒度流水时系统的吞吐量。

表 3 CFO compensation 模块拆分

模块	输入	输出	时间	DSP 核
step1	64	64	2 205	4alu + 2mac +
	samples	samples	cycles	2shifter
step2	64	64	2 205	4alu + 2mac +
	samples	samples	cycles	2shifter

### 4 评估和比较

通过细粒度流水和粗粒度流水,异构多核 DSP 需要九个 DSP 核和一个 Vieterbi 硬件加速器就可以实现 IEEE 802.11a 接收端基带处理,而异构多核 DSP 运行在 400 MHz 就能满足 IEEE 802.11a 协议的时间规定。不同 DSP 核中计算单元的数目和种类如表 4 所示。其中,IEEE 802.11a 接收端基带中不同处理模块在多核 DSP 上的映射如图 10 所示。

表 4 异构多核 DSP 中的 DSP 核

DSP 核	DSP 核上的 CU 数目和种类
DSP1	4alu + 2mac + 2shifter
DSP2	4alu + 2mac + 2shifter
DSP3	2alu + 2mac + 2shifter
DSP4	1alu + 2mac + 1shifter
DSP5	4alu + 2mac + 2shifter
DSP6	4alu + 2mac + 2shifter
DSP7	2alu + 2mac + 2shifter
DSP8	1alu + 2mac + 1shifter
DSP9	1alu + 1shifter + Virtebi Accelerator

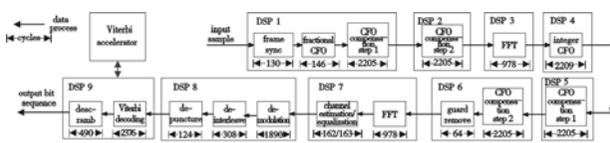


图 10 IEEE 802.11a接收端基带处理的各种模块在多核DSP上的映射

针对目标异构多核 DSP,采用本设计方法对 10 000 个 IEEE 802.11a 数据帧进行接收端处理。根据 3.2 节中所述,为了实现粗粒度流水,其他 DSP 核在执行完其上的处理后需要停顿一段时间。那么,从实验结果得到的 DSP 核执行和停顿状况如图 11 所示。同时,计算得到 DSP 核平均利用率约为 72.65%。

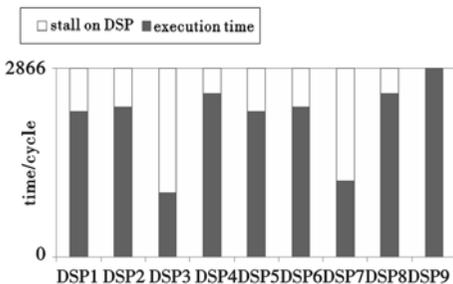


图 11 接收端基带处理实现中DSP核的执行和停顿

文献[10]在一个多核 DSP 上实现了完整的 IEEE 802.11a 基带接收端。其设计方法的实现需要使用 25 个 DSP 核、1 个 FFT 硬件加速器以及 1 个 Vieterbi 硬件加速器。根据文献[10]给出的实验数据可以看到,使用该方法实现 IEEE 802.11a 基带

接收端得到的 DSP 核平均利用率约为 51.25%。本文采用细粒度流水和粗粒度流水,能同时提高多核 DSP 核内和核间的并行计算性能。而文献[10]只考虑了多核 DSP 核间的并行计算性能,没有针对 DSP 核内的并行计算性能作优化。因此,文献[10]中 DSP 核空闲等待时间较长,平均利用率较低。从实验结果的对比可以看到,采用本文设计方法能得到较高的 DSP 核平均利用率,适用于 IEEE 802.11a 接收端基带处理。

### 5 结束语

本文在一个开环互连结构的异构多核 DSP 上设计并实现了 IEEE 802.11a 接收端基带处理。异构多核 DSP 具有九个 DSP 核和一个 Viterbi 硬件加速器。通过接收端处理在多核 DSP 上以核内细粒度流水和核间粗粒度流水的方式执行,来提高接收端处理的吞吐量。实验证明,本文的设计方法既能保证系统实时性,还能有效提高多核 DSP 中 DSP 核平均利用率。

下一步,笔者将在多核 DSP 上实现包括接收端和发送端的完整 IEEE 802.11a 基带处理。同时,准备在多核 DSP 上实现 IEEE 802.11n、IEEE 802.16e 等高速无线通信协议的基带处理,使多核 DSP 能灵活地实现各种复杂的无线通信协议,达到软件无线电的目标。

### 参考文献:

- [1] IEEE Std 802.11a/D7.0—1999, part11; wireless LAN medium access control (MAC) and physical layer (PHY) specifications: high speed physical layer in the GHz band [S]. New York; IEEE Press, 1999.
- [2] VLADIMIROVA T, PAUL J. Implementation of an IEEE 802.11a transmitter module for a reconfigurable system-on-a-chip design [C]//Proc of NASA /ESA Conference on Adaptive Hardware and Systems (AHS). San Francisco;IEEE Press,2009;305-305.
- [3] TANG Yi-yan, QIAN Lie, WANG Yu-ke. Optimized software implementation of a full-rate IEEE 802.11a compliant digital baseband transmitter on a digital signal processor [C]//Proc of IEEE Global Telecommunications Conference. St. Louis; IEEE Press, 2006; 5-2198.
- [4] LIN Yuan, LEE H, WOH M, et al. Soda: a low-power architecture for software radio [C]// Proc of the 33rd Annual International Symposium on Computer Architecture. Boston; IEEE Press, 2006; 89-101.
- [5] TELL E, NUKSSON A, LIU Da-ke. A programmable DSP core for baseband processing [C]//Proc of the 3rd International IEEE-NEW-CAS Conference. [S. l.]; IEEE Press, 2005; 403-406.
- [6] SCHMIDL T, COX D. Rubust frequency and timing synchronization for OFDM [J]. IEEE Trans on Communications, 1997, 45 (12): 1613-1621.
- [7] JIMENEZ V, GARCIA M, SERRANO F. A armada, seign and implementation of synchronization and AGC for OFDM-based WLAN receiver [J]. IEEE Trans on Consumer Electronics, 2004, 50 (11): 1016-1025.
- [8] XU Li, WANG Qin. CORDIC based algorithm for frequency offset estimation [C]//Proc of the 12th IEEE International Conference on Communication Technology. [S. l.]; IEEE Press, 2010; 817-819.
- [9] LEE E, MESSERSCHMITT D. Static scheduling of synchronous data flow programs for digital signal processing [J]. IEEE Trans on Computers, 1987, 32 (1): 24-35.
- [10] TRAN A, TRUONG D, BAAS B. A complete real-time 802.11a baseband receiver implemented on an array of programmable processors [C]//Proc of the 42nd Asilomar Conference on Signals, Systems and Computers (ACSSC). [S. l.]; IEEE Press, 2009; 165-170.