

LTE 上行 DFT/IDFT 的一种设计实现

刘少雄, 林平分

(北京工业大学 北京市嵌入式系统重点实验室, 北京 100122)

摘要: 根据 3GPP 协议规定, 提出一种适于 FPGA 实现的解决方案。采用分而治之和 WFTA 的算式分解, 最大限度地减少 DFT 的运算量; 采用块浮点动态截取多余位宽, 减少系统面积; 运用 4 个双端口 RAM 读写, 使系统能运行在流水线结构; 采用对称结构存储每一级的旋转因子, 最大化共享因子。

关键词: LTE; SC_FDMA; DFT; WFTA; FPGA; 流水线

中图分类号: TN929.5

文献标识码: A

文章编号: 1674-7720(2011)12-0064-04

An implement of the LTE UL DFT/IDFT

Liu Shaoxiong, Lin Pingfen

(Beijing University of Technology Beijing Embedded System Key Lab Beijing 100122, China)

Abstract: The aim of the paper is to give an FPGA implementation for the DFT. Use divide and conquer and WFTA to decompose the formula, maximum reduction computation of DFT. Use block floating point to dynamic interception bits wide, then reduce system size. Use 4 dual-port RAM to read and write, then the system can run in pipeline. Use Symmetrical structure stored the twiddle factor, maximize sharing factor.

Key words: LTE; SC_FDMA; DFT; WFTA; FPGA; pipeline

为了降低手机终端的功率损耗^[1], LTE 上行链路采用基于 DFT 扩频 OFDM (DFTS_OFDM) 的单载波传输, 又称为单载波 FDMA (SC_FDMA)。DFTS_OFDM 方案的基本结构如图 1 所示。3GPP 协议规定^[2]上行 PUSCH 信道产生 SC_FDMA 符号要求 DFT 点数满足式(1)。

$$M_{sc}^{PUSCH} = M_{RB}^{PUSCH} \times N_{sc}^{RB} \quad (1)$$

其中, $N_{sc}^{RB} = 12$, $M_{RB}^{PUSCH} = 2^{a_2} 3^{a_3} 5^{a_4} \leq N_{RB}^{UL} = 110$, a_2, a_3, a_4 取非负整数。

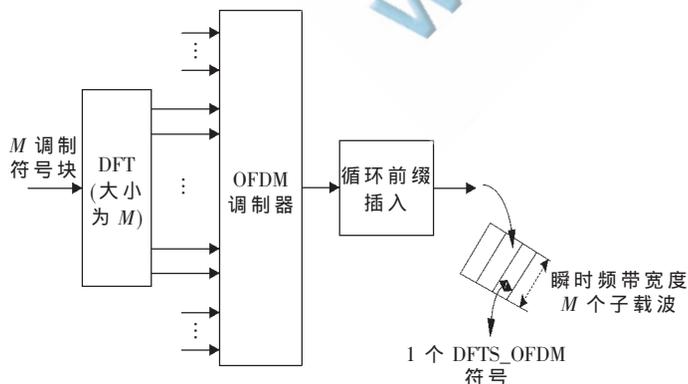


图 1 LTE 上行 DFTS-OFDM 的基本原理

由几个参数的变化可以得到最小 12 点、最大 1296 点共 35 种模式的 DFT^[3]。现在已有的研究方法(如质因子分解结合 WFTA 算法)解决非 2^n 点 DFT, 但此法不够灵活, 不适合长度可变的 DFT。在数字电视 DTMB 系统中, 3780 点 FFT 的处理采用分裂基与质因子分解结合 WFTA 算法实现, 但对于 LTE 上行可配置长度 DFT 的实现还没有一个成熟有效的方法。

根据 LTE 实时系统需求采用 pipeline 流水线结构实现高速可配置的 DFT 设计, 同时在结构和资源利用上进行优化, 最后给出仿真图形以及综合结果, 为上行 LTE 设计提供一种参考。

1 算法推导

由 3GPP 协议可以看出, LTE 中的 DFT 点数由 4、2、5、3 小因子构成, 可以将其分解成如式(2)形式:

$$N = r_1 M_1 = r_1 r_2 M_2 = r_1 r_2 r_3 M_3 \cdots = \prod_{i=0}^{r-1} r_i \quad (2)$$

其中 r_i 为小因子。可以用分而治之的方法分解 DFT 的表达式^[4-5], 减少计算量。N 点 DFT 公式为:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j \frac{2\pi n k}{N}} = \sum_{n=0}^{N-1} x(n) W_N^{nk} \quad (3)$$

网络与通信 Network and Communication

对于 LTE 系统来说最多的级数是 1 200 点, $N=1200=4 \times 4 \times 5 \times 5 \times 3=r_1 r_2 r_3 r_4 r_5$, 所以这里以一个 5 级的分而治之 DFT 推导为例, 采用正位序进倒位序出的方式对参数 k, n 进行重新组织:

$$n = n_1 r_2 r_3 r_4 r_5 + n_2 r_3 r_4 r_5 + n_3 r_4 r_5 + n_4 r_5 + n_5 \quad 0 \leq n_i < r_i$$

$$k = k_5 r_1 r_2 r_3 r_4 + k_4 r_1 r_2 r_3 + k_3 r_1 r_2 + k_2 r_1 + k_1 \quad 0 \leq k_i < r_i$$

将式(4)带入式(3)进行推导:

$$X(k) = X(k_1, k_2, k_3, k_4, k_5)$$

$$= X(k_5 r_1 r_2 r_3 r_4 + k_4 r_1 r_2 r_3 + k_3 r_1 r_2 + k_2 r_1 + k_1)$$

$$= \sum_{n=0}^{N-1} x(n) W_N^{kn} = \sum_{n=0}^{N-1} x(n_1 r_2 r_3 r_4 r_5 + n_2 r_3 r_4 r_5 + n_3 r_4 r_5 + n_4 r_5 + n_5) \times W_N^{(k_5 r_1 r_2 r_3 r_4 + k_4 r_1 r_2 r_3 + k_3 r_1 r_2 + k_2 r_1 + k_1)(n_1 r_2 r_3 r_4 r_5 + n_2 r_3 r_4 r_5 + n_3 r_4 r_5 + n_4 r_5 + n_5)}$$

$$= \sum_{n_1=0}^{r_1-1} \left\{ \sum_{n_2=0}^{r_2-1} \left\{ \sum_{n_3=0}^{r_3-1} \left\{ \sum_{n_4=0}^{r_4-1} \left\{ \sum_{n_5=0}^{r_5-1} x(n_1 r_2 r_3 r_4 r_5 + n_2 r_3 r_4 r_5 + n_3 r_4 r_5 + n_4 r_5 + n_5) W_N^{(k_5 r_1 r_2 r_3 r_4 + k_4 r_1 r_2 r_3 + k_3 r_1 r_2 + k_2 r_1 + k_1)(n_1 r_2 r_3 r_4 r_5 + n_2 r_3 r_4 r_5 + n_3 r_4 r_5 + n_4 r_5 + n_5)} \right\} \right\} \right\} \right\} \times W_N^{(k_5 r_1 r_2 r_3 r_4 + k_4 r_1 r_2 r_3 + k_3 r_1 r_2 + k_2 r_1 + k_1)(n_1 r_2 r_3 r_4 r_5)}$$

其中,

$$W_N^{(k_5 r_1 r_2 r_3 r_4 + k_4 r_1 r_2 r_3 + k_3 r_1 r_2 + k_2 r_1 + k_1)(n_1 r_2 r_3 r_4 r_5)} = W_{r_1}^{k_5 n_1} W_{r_2}^{k_4 n_2} W_{r_3}^{k_3 n_3} W_{r_4}^{k_2 n_4} W_{r_5}^{k_1 n_5}$$

由式(4)可知, 先做 N/r_1 个 r_1 点的 DFT, 乘以第一级和第二级之间的旋转因子 $W_{r_1 r_2}^{k_1 n_2}$, 然后做 N/r_2 个 r_2 点的 DFT, 乘以第二级和第三级之间的旋转因子 $W_{r_1 r_2 r_3}^{n_3(k_2 r_1 + k_1)}$, 再做 N/r_3 个 r_3 点的 DFT, 乘以第三级和第四级之间的旋转因子 $W_{r_1 r_2 r_3 r_4}^{n_4(k_3 r_1 r_2 + k_2 r_1 + k_1)}$, 再做 N/r_4 个 r_4 点的 DFT, 乘以第四级和第五级之间的旋转因子 $W_N^{n_5(k_4 r_1 r_2 r_3 + k_3 r_1 r_2 + k_2 r_1 + k_1)}$, 最后做 N/r_5 个 r_5 点的 DFT。对于 LTE 的运算方式如图 2 所示。

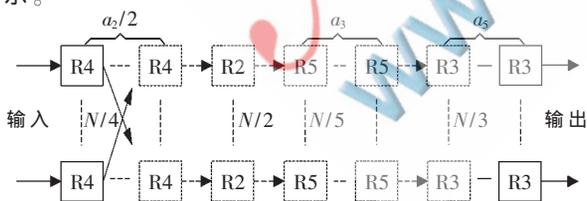


图 2 分而治之 DFT 阶段全图

2 总体结构及技术实现

2.1 整体结构框图

LTE DFT 的模块化总体结构如图 3 所示, 根据算法分析可以知道 LTE DFT 的分而治之需要几个阶段才能完成, 每个阶段需要做多次小因子点的 DFT, 所以图示是一个循环的形式。由状态机控制这些阶段的完成, 直到最后一个循环结束输出数据。

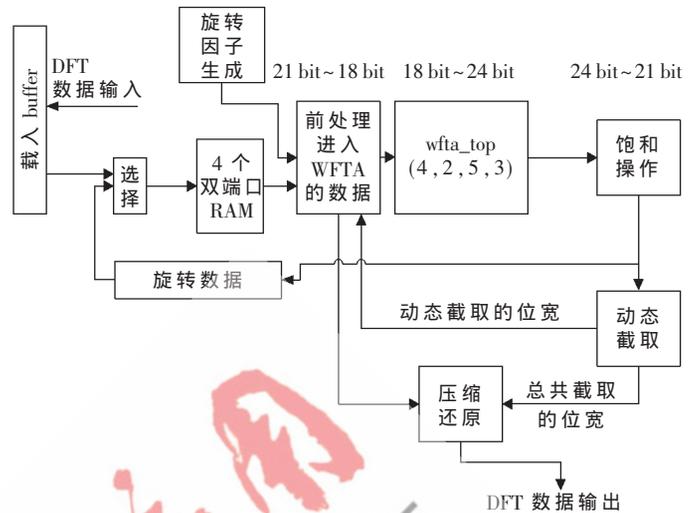


图 3 DFT 总体设计图

其中前处理进入 WFTA 模块的包括对 4 个双端口 RAM 的读取控制以及对旋转因子 ROM 的读取, 还有旋转因子地址的计算。饱和操作根据系统的最大 bit 数限定, 对经过 WFTA 计算后的数据进行饱和处理, 超过的 bit 数直接截取掉。

2.2 技术实现

2.2.1 4 个双端口 RAM 的数据存储

为保证 pipeline 地处理每次循环的数据, 这里采用 4 个双端口 RAM 对数据进行存取。对 4、2、5、3 四种小因子的 WFTA 计算来说, 选 4 个 RAM 最方便, 如果需要进行 4 点的 WFTA 计算, 则从每个 RAM 中读出一个数据, 这仅需要一个时钟就可读出 4 个数据。对 2 点的 WFTA 计算, 则可以一个时钟读出两组的 2 点 WFTA 进行计算。对 3 点的用一个时钟, 对于 5 点的用两个时钟读取。

在基于原位计算的基础上进行改进, 加入旋转数据模块, 是为了将本来是在一个 RAM 中的数据在填入 RAM 前进行旋转, 使其在不同的 RAM 中便于下一阶段 pipeline 读取。图 4 展示了一个最简单的 12 点的填写 RAM 实例, 在开始第一阶段前先将 12 点的输入数通过载入 buffer 模块用 12 个 clk 按图 3 顺序载入 4 个 RAM 中, 也就是将数据倒位序放入 4 个 RAM 中。将倒位序之后的数据重新标号, 即 1 对应载入 buffer 的 3, 2 对应 6 等。这样做的目的是为了更方便计算地址。例如, 在第一阶段读的过程中, 0、1、2、3 通过右移 2 bit, 即除以 4 可以算出地址为 0, 它们分别对应 4 个 RAM 的第 0 地址; 同理 4、5、6、7 除以 4 可以得到 1, 即对应 1 地址, 依此类推。

根据公式 4 的推导可知在第一阶段 DFT 的处理中不需要乘以旋转因子, 所以旋转因子为 0, 在第一阶段和第二阶段中需要先乘以旋转因子, 旋转因子按照公式推导处理列出在表中。在第一阶段先处理 0、1、2、3 四点的 WFTA, 然后按原位顺序填入 4 个 RAM, 接着处理 4、5、6、7 四点的 WFTA, 本来应该也按原位填入 RAM

网络与通信 Network and Communication

中,但是注意到在第二阶段需要处理 0、4、8 三点的 WFTA,如果还按照原位填入,则 0、4、8 三个数据在同一个 RAM 中,要读取这 3 个数需要 3 个 clk,显然不适应 pipeline 的处理。所以在做完 4、5、6、7 四点的 WFTA 之后将数据旋转再写入 4 个 RAM 中,同样将 8、9、10、11 四点的结果也旋转,如图 4 所示。这样的读写 RAM 操作可以保证 pipeline 的处理。



图 4 12 点的 DFT 处理

2.2.2 旋转因子的存取

根据式(4)的推导,每一级之间需要先乘以旋转因子,对于旋转因子的地址计算依据式(4)的推导。由于要实现 35 种可配置模式的 DFT 设计,所以在实现时要尽可能地考虑旋转因子的共享存储,从而尽可能地减少存储这些旋转因子的 ROM 大小。

旋转因子 $W_N^{nk} = e^{-j\frac{2\pi kn}{N}}$,一般做法是将 N 点的旋转因子全部存储,然后根据算出来的 nk 乘积来查找对应的旋转因子,这样 35 中模式需要很多的 ROM 地址来存储。这里将具有 2 的幂次方关系的旋转因子共用,如 12、24、48...768 点 DFT 的旋转因子共用,12 点的旋转因子是 24 点的一部分,24 点的是 48 点的一部分等,这样就只需要存储具有两的幂次方关系的 DFT 点数的最大那个点 768 点,又由于旋转因子自身的对称性,只存储最大点数的 1/8 就可以了,其他部分通过对称性来查找。具体实现步骤如下:

(1)根据 2 的幂次方关系特性,将 35 种模式的 DFT 旋转因子分成 10 组,并存储这 10 组中最大的点的八分之一构成一个 ROM。对于 N 点(对应组中最大的点),只存储 $\lceil N/8 \rceil$ 个地址数据;

(2)对于计算出的旋转因子地址 K ,根据它所处的 DFT 模式,选择它所属的组,10 组分别用 $\{R_0, R_1, R_2, \dots, R_9\}$ 表示;

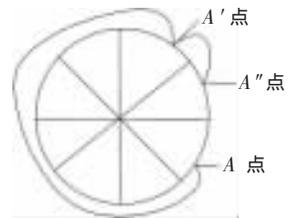
(3)如果 K 在 R_5 ,则 $R_0+R_1+R_2+R_3+R_4$ 为它的偏移地址 offset;

(4)12 点的 DFT 需要用此组中最大的 768 点 ROM 表来找数,则地址 K 有可能是 $[0, \dots, 11] \times 768/12$ 中的一个作为有效地址 eff_dft_addr ;

(5)对于算出的 eff_dft_addr ,根据对 $\lceil N \times 1/8 \rceil, \dots, \lceil N \times 7/8 \rceil$ 的比较找出它处于 768 点中的哪个位置(此处 N 为 768),即哪个 1/8 象限;

(6)找出所处的象限后,再找出其在第一个 1/8 对称的位置值 dft_8_addr ,计算出 $dft_addr = offset + dft_8_addr$,然后在 ROM 表中找出对应的值,再根据对称性还原其

原来的所属象限的值。如图 5 所示,展示一个点的查找方式。通过查找 A'' 的值来得到 A 的值。



2.2.3 WFTA 的运算单元

WFTA 算法对 2、3、4、5、7、8、9、16 等小 N 点有较快速处理能力,它将小 N 点 DFT 转换为循环卷积,利用多项式理论使卷积计算尽可能减少乘法。

一个 N 点的 DFT 可以表示为 $X=Wx$, W 为 $N \times N$ 的旋转因子矩阵,Winograd 证明那几个小 N 点的 W 矩阵可以表示为 $W=CGB$, C 表示为 $N \times J$ 的矩阵, G 表示为 $J \times J$ 的对角矩阵, B 表示为 $J \times N$ 的矩阵。 CB 矩阵的元素都是比较简单的数(0、 ± 1 、 ± 2 等), G 矩阵元素一般为实数或纯虚数。LTE 只用到 4、2、5、3 四点,下面给出 3 点的 WFTA 算法^[5]:

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & W_3^1 & W_3^2 \\ 1 & W_3^2 & W_3^1 \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ \frac{3}{2} \\ -j\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \end{bmatrix} \quad (7)$$

由图 6 所示 3 点的 WFTA 信号流程图可知,用了 6 个加减法,通过化简并移位,实际只用了一个乘法。

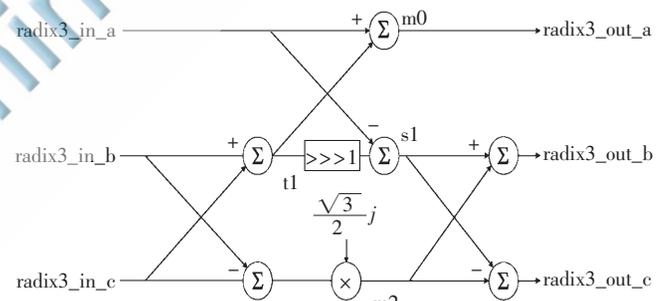


图 6 3 点 WFTA 流程图

2.2.4 块浮点的数据处理

定点运算的特点是速度快但动态范围小。浮点运算的特点则是动态范围大但占用资源大。块浮点具有两种运算的优点,是两种运算的折中,让一组数具有共同的阶码,这个阶码是同组数中最大的那个数的阶码,简化系统资源提高运算的精度^[6]。

如表 1 所示,因为每次 WFTA 运算后都有数据位宽的扩展,本结构具有 3 bit 的扩展。为保持输入 $wfta_top$ 的模

表 1 块浮点处理

高 4 位	扩展位宽
0000 1111	0
0001 1110	1
001X 110X	2
01XX 10XX	3

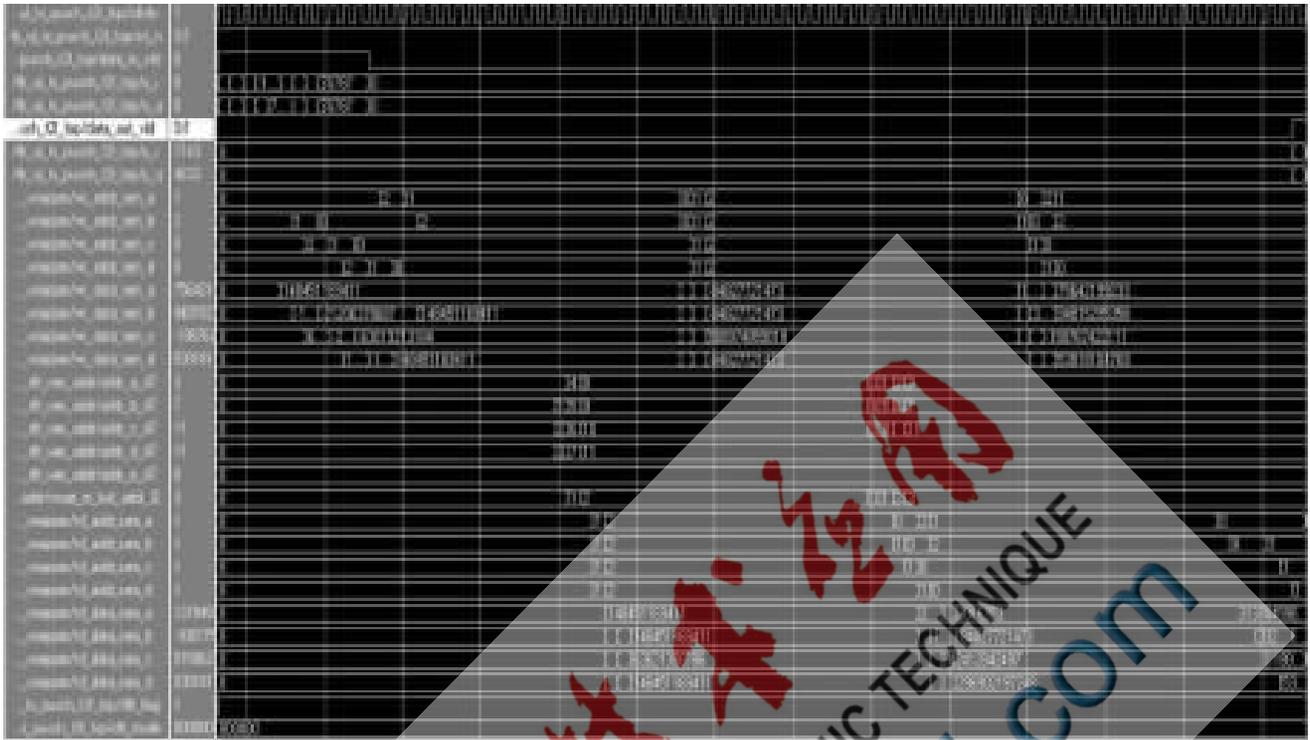


图7 12点DFT的仿真图

块数据始终为 18 bit, 这里用块浮点动态截取的方法对每一级的 WFTA 结果进行处理, 动态截取的位宽决定下一级的数据宽度, 同时循环累加每个阶段的阶码, 在数据输出时进行还原操作。

3 仿真综合

图 7 所示为 12 点 DFT 的仿真图形, dft 模式是第一种, 首先 data_in_vld 为高时开始数据输入, 然后用 12 个 clk 将数据读入 4 个 RAM, 之后计算第一级 RAM 读取地址将数据读出, 处理 3 次 4 点的 DFT, 处理后将数据写入 RAM, 需要 3 个 clk; 再后读出数据做 4 次 3 点的 DFT, 处理后将数据写入 RAM, 需 4 个 clk; 最后将数据读出做压缩还原处理, data_out_vld 为高后 pipeline 出数, 需要 12 个 clk。理论上需要 31 个 clk, 但是在处理中需要处理与其他模式的共享, 还要有打拍延时等操作, 实际用掉 98 个 clk。120 点的 DFT 实际用 502 个 clk, 理论上是 $120 \times 2 + 30 + 30 + 24 + 40 = 364$ 个 clk, 说明处理的点数越多冗余 clk 比例越小。

使用 stratrix III EP3SL340F1517I3 芯片, 运用 Quartus II 综合后的结果为: 7 824 个组合 ALUT, 0 个内存 ALUT, 8 699 个逻辑寄存器, 可达到时钟 124.64 MHz, 满足 LTE 系统时钟 122.88 MHz 的要求。

文章在介绍 LTE 上行 SC_FDMA 的基础上, 对 35 种模式的 DFT 预编码进行算法分析, 提出并用 FPGA 实现了一种高速可配置的方案。文中对数据存储、WFTA 运算单元和块浮点处理进行简单表述, 根据旋转因子特性, 详细介绍了旋转因子的优化, 大大降低了 35 种模式

旋转因子的存储大小。最后给出的仿真综合结果表明该方案具有较好的性能。

参考文献

- [1] DAHLMAN E. 3G evolution: HSPA and LTE for mobile broadband. Published by Elsevier Ltd. 2007; 75-81.
- [2] 3GPP TS 36.211. Evolved universal terrestrial radio access (E-UTRA). Physical channels and modulation.
- [3] Xilinx. LogiCORE IP discrete fourier transform v3.1. DS615. 2009.
- [4] 何小敏. LTE 系统中 DFT 快速算法研究 [DB/OL]. (2009-12-24). 中国科技论文在线. http://www.paper.edu.cn/paper.php?serial_number=2009/2-937.
- [5] 胡广书. 数字信号处理——理论、算法与实现 [M]. 北京: 清华大学出版社, 2003.
- [6] 陈丽安, 张培铭. 定点 DSP 块浮点算法及其实现技术 [J]. 福州大学学报, 2004, 32(6): 689-693.

(收稿日期: 2010-12-18)

作者简介:

刘少雄, 男, 1984 年生, 硕士研究生, 主要研究方向: LTE 无线通信。

林平分, 女, 1947 年生, 教授, 主要研究方向: 无线通信, 数字信号处理, 嵌入式实时系统。