

可配置高速高精度 FFT 的硬件实现

邓波, 戎蒙恬, 汤晓峰

(上海交通大学芯片与系统设计与中心, 上海 200030)

摘要: 提出了一种高速、可变长点、混合基 8/4/2、浮点的 FFT 硬件模块化设计方案。设计方案中, 改进了基 8/4/2 混合基算法, 能够处理可变长 $2^N(3 \leq N \leq 12)$ 采样点; 提出了一种乒乓 RAM 结构和数据地址的组织, 可以同时存、取和处理 16 个数据, 保证处理实时性; 采用了超长流水线浮点执行单元, 提高了处理结果的精度。目前, 该设计已在 FPGA 上实现, 采样点长 4k 时处理能力为 250MSPS。采用 0.18μm CMOS 工艺综合, 4k 点时处理能力可达到 800MSPS。

关键词: 混合基; 乒乓 RAM; 浮点执行单元; 流水线

Hardware Implementation of Configurable High-speed and High-precision FFT

DENG Bo, RONG Mengtian, TANG Xiaofeng

(Design Center of Core and System, Shanghai Jiaotong University, Shanghai 200030)

【Abstract】 A novel method is proposed to design a configurable, mixed-Radix 8/4/2 and floating-point high-speed FFT processor. The improved mixed-Radix 8/4/2 algorithm, ping-pong RAM architecture, data address generator and pipelining floating-point process unit which enhance the processor's performance to read, write, compute 16 data simultaneously and process unfixed point with high-precision are discussed in detail. The design has been verified on the FPGA platform and synthesized in 0.18μm CMOS technology. It is showed that the IO rate is 250MSPS at 4k point and will reach to 800MSPS after its migration to ASIC.

【Key words】 Mixed-Radix; Ping-pong RAM; Floating-point process unit; Pipelining

快速傅里叶变换(FFT)在信号处理中十分重要, 在音频处理、OFDM/DMT、雷达等领域有着广泛的应用。传统 FFT 硬件实现多为定长, 定点/块浮点, 在速度和硬件开销方面有着较好的考虑, 但不可配置, 采样点长度发生变化就不适用。而且, 定点/块浮点的结构有较大的误差, 精度较低。综合上述考虑, 本文提出了一种可配置(采样点数 $8 \sim 4096$)、浮点、高吞吐率的 FFT 硬件实现。

1 算法选取

1.1 各种 FFT 硬件算法比较

FFT 的算法包括卷积型和递归型算法。前者在运算量上占优, 用的乘法器较少, 但控制过于复杂, 不利于硬件实现; 后者则具有规整的结构, 每轮运算具有几乎相同的结构, 便于硬件的实现, 并且递归算法在进行浮点数运算时, 其精度要比 WFTA 和 PFA 算法高。本文的算法即为递归算法。

通常, FFT 硬件实现多为采样点数固定(2^{2N} 或 2^{3N}), 采用基 4、基 8 或者分裂基算法。例如在采样点数为 8k 时, 采用基 2/8 的分裂算法^[1], 对 n 为偶数部分采用基 2 算法, 而奇数部分采用基 8 算法, 减少运算中乘、加法次数, 但其通用性不强, 当采样点长度发生变化时不可配置。本文采用改进的基 8/4/2 混合基数的算法, 采样点数可配置。基本思想在于优先采用基 8 算法迭代, 如果经过若干次基 8 迭代后不能完成 FFT 运算, 再根据采样点长度选择一次基 4 或者基 2 运算, 合起来就可完成长度为 $2^N(3 \leq N \leq 12)$ 点的 FFT 运算。

1.2 基 8 时域抽取算法

考虑到基 8 算法迭代层数少, 在速度上有较大优势^[2],

本设计优先进行基 8 迭代。其算法局限性在于采样点长度必须为 2^{3m} 。下面介绍采样点长度为 $N=2^{3m}$ 点的基 8 FFT 变换:

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{kn}, k, n = 0, 1, 2, \dots, N-1 \quad (1)$$

将 N 分解, 得到

$$N = t_2 t_1 = 8 \times N/8, t_1 = N/8, t_2 = 8, \quad (2)$$

$$k = k_1 t_1 + k_0, k_1 = 0, 1, \dots, 7, k_0 = 0, 1, \dots, N/8$$

$$n = n_1 t_2 + n_0, n_0 = 0, 1, \dots, 7, n_1 = 0, 1, \dots, N/8 \quad (3)$$

将式(2)、式(3)带入式(1), 得

$$X(k) = \sum_{n_0=0}^7 \left[\sum_{n_1=0}^{N/8-1} x(8n_1+n_0)W_N^{8n_1k} W_N^{n_0k} \right] \quad (4)$$

不妨引入 $A(n_0)$, 令

$$A(n_0) = \sum_{n_1=0}^{N/8-1} x(8n_1+n_0)W_N^{8n_1k} = \sum_{n_1=0}^{N/8-1} x(8n_1+n_0)W_{N/8}^{n_1k} \quad (5)$$

将式(5)代入式(4), 得到

$$X(k) = \sum_{n_0=0}^7 A(n_0)W_N^{n_0k} = \sum_{n_0=0}^7 [A(n_0)W_{N/8}^{n_0k_0} W_8^{n_0k_1}] \quad (6)$$

观察式(5)和式(6)可知, N 点 FFT 运算即拆分为内层的

基金项目: 上海市科委科技发展基金资助项目(037062022)

作者简介: 邓波(1979—), 男, 硕士生, 主研方向: 通信系统集成电路设计, SOC 设计; 戎蒙恬, 教授、博导; 汤晓峰, 硕士生

收稿日期: 2005-09-14 E-mail: ski_db@sjtu.edu.cn

N/8 点和外层的 8 点 FFT，内层可以继续拆分下去，最终实现基 8 迭代算法。迭代的层数 $L = \log_8(N)$ 。

1.3 改进的混合基算法

本文采用改进的混合基算法，根据 FFT 的点数，选择迭代的次数和最后一次迭代的基数，可配置性好。以 1024 点为例，前 3 轮使用基 8 迭代，最后一轮使用基 2 迭代，即可实现 1024 点的 FFT 运算。

1024 点的 FFT 运算为

$$X(k) = \sum_{n=0}^{1023} x(n)W_N^{kn}, k, n = 0, 1, \dots, 1023 \quad (7)$$

参照 1.2 节中的推导，得到下式：

$$X(k_1, k_0) = \sum_{n_0=0}^1 \left[\sum_{n_1=0}^{511} x_0(n_1, n_0) W^{2n_1 k} W^{n_0 k_0} \right] \quad (8)$$

其中 $512=8^3$ ，内层进行 3 轮基 8 迭代，最后经过外层的基 2 迭代，即得到 1024 点的 FFT 运算。

图 1 给出了 16 点混合基(基 8 + 基 2)信号流程。

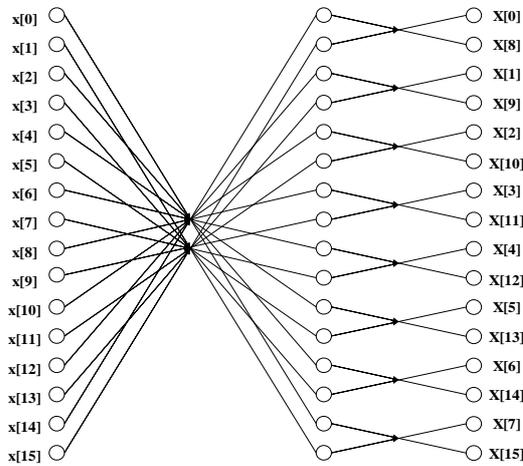


图 1 16 点混合基(基 8 + 基 2)信号流程

迭代次数以及最后一次迭代基的选择如表 1 所示。

表 1 迭代次数和迭代基的选择

FFT 采样点数 N	迭代次数	最后一次迭代选择的运算基
$2^{3k} (1 \leq k \leq 4)$	K	基 8
$2^{3k+1} (1 \leq k \leq 3)$	k+1	基 2
$2^{3k+2} (1 \leq k \leq 3)$	k+1	基 4

2 硬件结构

该 FFT 硬件结构如图 2 所示，包括 I/O 模块，运算模块 (Computing Module, CM) 和处理控制模块 (Process Control Module, PCM)。其中 CM 包括两个基 8/4/2 混合运算单元 (Radix8/4/2) 和旋转因子生成单元 (Twiddle Factor Generator)。

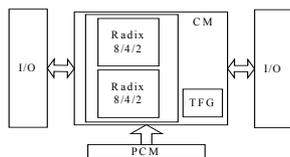


图 2 FFT 硬件结构

下面将分别介绍这几部分。

2.1 I/O

基于该设计高速、低功耗的考虑，I/O 部分采用低电压差分信号 (LVDS) 接口，共有 2 个输入通道和 2 个输出通道，数据格式为 DDR，保证了 IO 接口的高速率。

2.2 运算模块 CM

运算模块包括两个基 8/4/2 混合运算单元 (Radix8/4/2) 和旋转因子生成单元 (TFG)。

基于本文的设计目标，迭代次数最多为 4 次，为避免硬件上的过多耗费，并从降低功耗的角度出发，未采用具有完整蝶式展开的纯流水结构，而采用迭代结构。同时，在运算模块中放入两个 Radix8/4/2 运算单元，提高了处理能力。其中 Radix8/4/2 组织方式如图 3 所示。

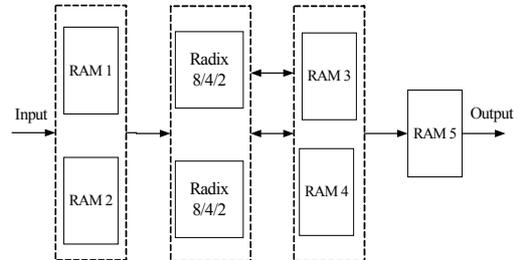


图 3 Radix8/4/2 组织方式

考虑到 I/O 速率和内核工作速率的区别，以及实时处理的需要，数据存取和运算必须

同时进行。因此，将 RAM 划分为 5 块。RAM 1 和 RAM 2 为乒乓模式，作为数据的输入缓冲，保证数据的无阻塞性。RAM 3 和 RAM 4 用来暂存每轮迭代运算的中间数据。运算完成后，经过调整顺序，最终数据输入到 RAM 5，经缓冲后输出。

基于 Radix8/4/2 的结构，每个基 8 运算需要读入和输出 8 个数据，运算后还需混序，如图 4 所示。若按照常规方式顺序存取，最坏情况需要 8 个周期，RAM 存取成为系统瓶颈。本设计精心组织了 RAM 地址，RAM 和内核工作在同频时即可满足要求，同时在每轮运算中 RAM 存取地址相对固定，减少了控制复杂度。

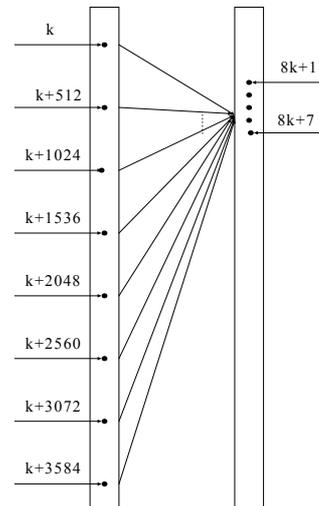


图 4 4k 点数据存储位置

正常工作中，优先进行基 8 迭代，根据表 1 决定迭代层数和最后一次迭代的基。每个 Radix8/4/2 单元可以做 1 个基 8 运算，还可以根据需要分解为 2 个基 4 或者 4 个基 2 结构。

Radix8/4/2 单元中，流程最长的即为基 8 运算，其流程如图 5 所示。可以看到，基 8 结构包括一系列的乘法和加法运算，其中： $w_8^2 = -j$ ，只需将加法运算作适当调整即可。 w_8^1, w_8^3 为常数，乘法可以用加法替换^[1]。

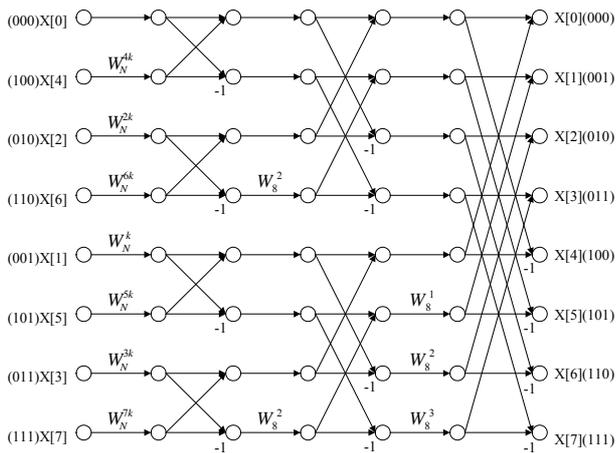


图 5 基 8 算法流程

每次运算所需的旋转因子由 TFG 生成，每个 Radix8/4/2 单元最多需要 7 个旋转因子。适当调整存取结构，保证两个 Radix8/4/2 单元的旋转因子相同，只需生成一组旋转因子即可，节省了一半的硬件开销。

2.3 处理控制模块 PCM

处理控制模块监控整个 FFT 过程，包括运算控制单元 (CCU)，用以计算迭代次数，选择最后一次迭代运算基，产生 Radix8/4/2 和 TFG 运算中所需控制信号；RAM 控制单元 (RCU)，控制数据的输入，输出和存取迭代运算的中间数据。由于该 FFT 采样点数的可配置性，在控制生成旋转因子和 RAM 存取地址上具有比较大的挑战性。

3 精度优化

硬件 FFT 多采用定点或块浮点结构^[3]，实现较为简单，但精度难以满足要求。因此，本文采用浮点结构，利用多级流水线实现浮点执行单元 (FPU)。

3.1 浮点数格式

浮点数存储格式如图 6 所示：第 22 位是符号域，表示该数据的符号；第 21 位到 17 位是指数域，表示该数据的指数 exp，偏置为 18；第 16 位到 0 位用来表示该数据的分数位 f。在这种表示方式中，尾数隐含了一个前 1，即实际的尾数为 1.f。

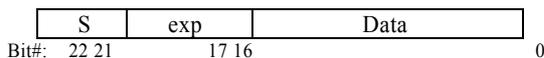


图 6 23 位浮点数格式

3.2 加法器

加法器采用回绕进位加法器 (ECA)^[4]，4 级流水线实现：

- (1) 比较两个操作数的指数位并计算出其差值 shifter，将较大的指数位作为运算结果的指数位；
- (2) 将指数位较小的操作数左移 shifter 位，对齐数据。利用桶型移位寄存器^[5]可以实现一个周期内的任意位平移；
- (3) 根据两个操作数的符号位和操作符对已经对齐的两个操作数进行运算；
- (4) 将运算结果规格化。

3.3 乘法器

乘法器采用改进的 BOOTH^[6]算法，两级流水线实现：

- (1) 根据两个操作数的指数位计算结果的指数位和符号位，并判断是否溢出。
- (2) 隐含的前 1 与分数组合，做乘法运算并将结果规格化。在结果规格化时，根据最低有效位的指数位决定多余位

的进位与否，降低了舍入误差。4K 采样点时块浮点运算和浮点运算的绝对误差如图 7、图 8 所示。

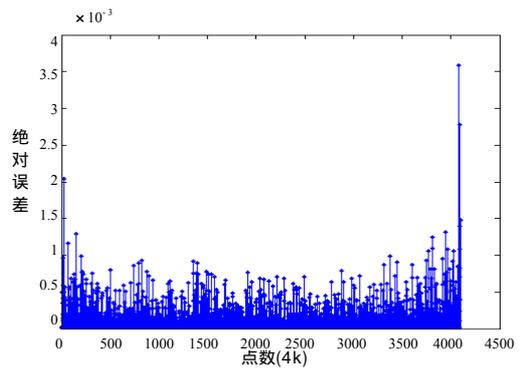


图 7 4K 采样点时块浮点绝对误差

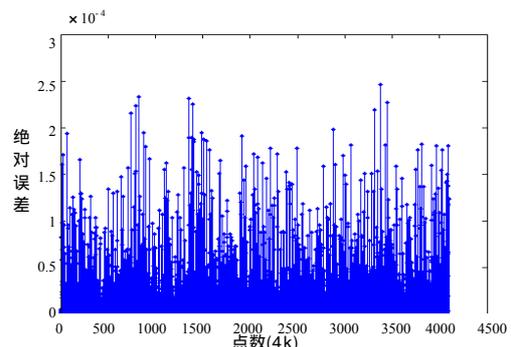


图 8 4K 采样点时浮点绝对误差

通过上述多级流水线的浮点执行单元，在大幅提高精度的同时保证了速度，工作频率可以达到 200MHz 以上。

4 VLSI 实现

4.1 ASIC 平台

该 FFT 设计采用了 Top-Down 流程。使用 Matlab 验证算法的正确性，设计了 RTL 级的 Verilog_HDL 代码，使用 Modelsim 进行了功能仿真。在 0.18μm 1P6M CMOS 标准工艺下，使用 DC 将设计综合为门级网表，用 Silicon Ensemble 完成版图设计，并通过了共 45 万门的后仿真，使用了 920kB RAM，该设计的系统工作主频为 200MHz。采样点长 4k 时处理能力为 800MSPS，当采样点长减小为 512 时，提高到 1.06GSPS。

4.2 FPGA 平台

考虑到 FPGA 内部布线资源的限制，只放入一个 Radix8/4/2 单元，IO 只使用了单通道的 LVDS，回绕进位加法器和 BOOTH 乘法器均调用内部 IP Core，桶型移位寄存器用自定时移位寄存器 (Self-timed Shift Register)^[7]替代。

本文写作时，FPGA 平台上的验证已经通过，在 XILINX VIRTEX-II PRO 50 上，占用了 26% 的乘法器，24% 的 RAM 和 55% 的逻辑资源。实际工作主频为 130Hz，功耗为 4.6W，采样点长度 4K 时处理能力为 250MSPS，计算一次 4K 点 FFT 需要 16μs。当采样长度为 512 时，处理能力提高到 330MSPS。

5 结束语

FFT 在信号处理领域有着广泛应用，其结构相对比较简单和固定，适合用 FPGA 或者 ASIC 硬件实现。本文介绍了一种可变长点 (8 ~ 4096)，混合基 8/4/2，浮点的模块化设计。设计具有可配置、高处理能力、高精度的优点，具有很强的应用价值。

(下转第 282 页)