

## 用 FPGA 实现浮点 FFT 处理器的研究\*

王远模 赵宏钟 张 军 付 强

( 国防科技大学电子科学与工程学院 湖南 长沙 410073 )

**摘 要** 针对定点 FFT 处理器精度不高的缺点,提出了浮点格式 FFT 处理器的 FPGA 硬件实现方案。详细阐述了 FFT 处理器的自定义浮点格式确定、算法选择和浮点加法实现等关键技术。该处理器已投入使用,工作性能稳定,系统时钟 80MHz,完成 1024 点 FFT/IFFT 运算只需 64 $\mu$ s,误差小于 -80dB。

关键词 :FPGA ;FFT ;蝶形运算

中图分类号 :TN47 文献标识码 :A

## The Realization of Floating-point FFT Processor with FPGA Chip

WANG Yuan-mo ZHAO Hong-zhong ZHANG Jun FU Qiang

( College of Electronic Science and Engineering , National Univ. of Defense Technology , Changsha 410073 , China )

**Abstract** :The FPGA realization of a floating-point FFT processor is proposed to get over the poor precision of the fixed-point FFT processor. The definition of the floating-point format ,the selection of arithmetic and the key techniques of the FPGA realization are discussed. Such a processor has been put into service and has stable performance. Its operating frequency is 80MHz and it can finish 1024 point FFT/IFFT in 64 $\mu$ s with an error less than -80dB.

**Key words** :FPGA ;FFT ;butterfly computation

在现代雷达、通信、图像处理等领域中,数字信号处理系统经常要进行高速、高精度的 FFT 运算。实现 FFT 有两种方式:通用 DSP 实现和专用集成电路实现。通用 DSP 实现的优点是硬件开发和软件编程技术成熟、开发时间短,缺点是硬件电路复杂、功耗大。专用集成电路实现的优点是速度快、功耗低,整个 FFT 运算由一块芯片完成,缺点是开发时间长、成本高。现场可编程逻辑阵列(FPGA:Field Programmable Gate Array)是一种半定制集成电路,具有面向数字信号处理算法的物理结构。用 FPGA 实现 FFT 处理器具有硬件系统简单、功耗低的优点,同时具有开发时间较短、成本较低的优势。国内外学者在用 FPGA 实现 FFT 处理器方面做了大量的工作<sup>[1~3]</sup>,取得了许多成果。但是在这些处理器中,由于数据格式采用定点方式,处理精度不高,不能满足高性能信号处理系统的精度要求,为此,本文提出了浮点数据格式的 FFT 处理器的硬件实现,并用 FPGA 器件(XC2V1000)实现了一个可变长度的 FFT/IFFT 处理器。

## 1 关键技术

## 1.1 自定义浮点数据格式

在标准 IEEE 单精度浮点格式中,其数值用 32bit 表示。其中,bit31 是符号位,用  $S$  表示,bit30 ~ bit23 是指数位,用  $e$  表示,bit22 ~ bit0 是尾数位,用  $f$  表示。则数值  $v$  有如下表示:

$$\begin{cases} v = (-1)^S \times 2^{e-127} \times (1.f) & (e \neq 0) \\ v = 0 & (e = 0) \end{cases} \quad (1)$$

用 FPGA 实现标准浮点格式的 FFT 处理器时,占用资源严重。为了克服这个缺点,该处理器内部采用自定义的浮点数据格式。为了确定合适的自定义浮点格式,下面分析蝶形运算对处理器中的数据动态范围的影响<sup>[4]</sup>。基 2 蝶形运算用数学式表示有

\* 收稿日期 2004 - 05 - 20

作者简介:王远模(1973—)男,博士生。

$$\begin{cases} X_{m+1}(p) = X_m(p) + X_m(q) \cdot W_N^r \\ X_{m+1}(q) = X_m(p) - X_m(q) \cdot W_N^r \end{cases} \quad (2)$$

其中  $X_m(p)$  和  $X_m(q)$  为输入数据,  $X_{m+1}(p)$  和  $X_{m+1}(q)$  为输出数据,  $W_N^r$  是 1 的一个复根, 称为旋转因子。

由式(2)可得<sup>[4]</sup>:

$$\left[ \frac{|X_{m+1}(p)|^2 + |X_{m+1}(q)|^2}{2} \right]^{\frac{1}{2}} = \sqrt{2} \left[ \frac{|X_m(p)|^2 + |X_m(q)|^2}{2} \right]^{\frac{1}{2}} \quad (3)$$

$$\max\{|X_m(p)|, |X_m(q)|\} \leq \max\{|X_{m+1}(p)|, |X_{m+1}(q)|\} \leq 2 \max\{|X_m(p)|, |X_m(q)|\} \quad (4)$$

因此, 蝶形运算使输出数据的模值扩大到输入数据的 2 倍。

在实际工程应用中, FFT 运算的输入数据通常是中频采样、变频、滤波后的 12-bit、16-bit、32-bit 定点数据。浮点数据的动态范围由指数确定, 该处理器中用 6-bit 表示, 能表示 380dB 动态范围内的数据。

浮点数尾数的位数决定 FFT 运算的处理精度, 在高性能的信号处理系统中, 要求其运算误差足够小, 一般在 -80dB 以下。根据理论分析和仿真, 该处理器用 17-bit 表示浮点数的尾数。

根据以上分析, 自定义浮点数据格式用 24-bit 表示: bit23 是符号位, 用  $S$  表示; bit22 ~ bit17 是指数位, 用  $e'$  表示; bit16 ~ bit0 是尾数位, 用  $f'$  表示。其数值  $v$  表示为:

$$\begin{cases} v = (-1)^S \times 2^{e'-127} \times (1.f') & (e' \neq 0) \\ v = 0 & (e' = 0) \end{cases} \quad (5)$$

用 FPGA (XCV1000) 实现两个标准格式的浮点实数相加需要 584 个 slices; 实现两个自定义格式的浮点实数相加只需要 277 个 slices; 实现两个标准格式的浮点实数相乘需要 285 个 slices 和 4 个 MULT18X18s; 实现两个自定义格式的浮点实数相乘只需要 50 个 slices 和 1 个 MULT18X18s, 所以, 在处理器内部采用自定义浮点数据格式能有效减少实现处理器的硬件资源占用。

## 1.2 FFT 算法选择

提高 FFT 处理速度的四个主要技术途径是采用流水线结构、并行运算、增加蝶形处理单元数目和高基数结构<sup>[4]</sup>。确定算法中的基数  $r$  时, 必须综合考虑算法的处理速度、算法的运算量以及硬件实现算法的资源占用等因素。

基 2 算法由 1 个复数乘法和 2 个复数加法组成, 而一个基 4 算法由 3 个复数乘法和 8 个复数加法组成。虽然基 4 算法的运算速度是基 2 算法的 2 倍, 但是基 4 蝶形硬件实现的资源占用是实现 2 蝶形的 2 倍多。另外, 基 4 算法要求 FFT 的处理长度  $N = 4^m$  ( $m$  为自然数), 在工程实际中有时不能满足该条件(如  $N = 512$ )。综合考虑, 该处理器选用基 2 算法。

## 1.3 浮点加法器的实现

实现基 2 蝶形运算时, 由于并行、流水实现的实数加法多达 14 个, 实现浮点加法的硬件电路复杂, 因此, 其设计技术直接决定着处理器的性能。两浮点数<sup>[5]</sup>相加通过以下步骤来完成:

- (1) 浮点加法的对阶处理;
- (2) 对阶处理后的尾数相加;
- (3) 对相加后的结果进行规格化处理。

FPGA 器件由大量结构规整的 slice 组成, 每个 slice 结构如图 2 所示, 包括两个 4 输入函数(LUT/RAM16/SRL)、2 个进位逻辑(CY)、2 个算术逻辑(Arithmetic Logic)、2 个数据选择器逻辑(MUXFx)和 2-bit 缓冲存储(Register)等资源。

对阶处理是实现浮点加法的最复杂、最关键的步骤。传统的对阶处理用桶形移位器来实现, 该方法占用硬件资源多, 功耗大。本处理中, 结合 slice 结构特点, 采用基于数据选择器的方法实现。

实现对阶处理, 就是将指数小的浮点数的尾数向右移  $n$  位 ( $n$  为两浮点数的指数差) 移出后的高位补上符号位。对阶处理实现可以由一系列数据选择器(2 选 1、4 选 1、8 选 1、16 选 1 等)组合完成, 选择器的输入是二进制补码表示的较小指数的浮点数尾数  $DA18 \sim DA0$  ( $DA18$  为符号位) 和 0, 选择器

输出就是对阶处理结果  $DB_{18} \sim DB_0$  数据选择器控制码  $S_5S_4S_3S_2S_1S_0$  是指数差  $n$  的二进制表示。为了便于理解,以  $DB_1$  的实现为例,当  $n=0$  时,  $DB_1=DA_1$ ; 当  $n=1$  时,  $DB_1=DA_2$ ; ..... ; 当  $n=17$  时,  $DB_1=DA_{18}$ ; 当  $n>17$  时,  $DB_1=0$ 。也就是说,  $DB_1$  是一个以  $DA_{18} \sim DA_1$  和 0 为输入数据,以  $n$  为控制码的 19 选 1 的选择器输出。一个 slice 能实现 4 选 1 逻辑,两个 slice 能实现 8 选 1 逻辑,4 个 slice 能实现 16 选 1 逻辑。

对浮点加法的结果进行规格化处理,需要知道前导(0/1)的位数(有效位数前面的符号位数),从而确定规格化时位移的位数。结合 FPGA 结构特点,利用其专门高速进位逻辑通道,实现前导(0/1)的快速判定。需要规格化数用  $S m_k m_{k-1} \dots m_1 m_0$  表示,其中,  $S$  表示符号位,输出前导(0/1)判定结果  $LZA_i (0 \leq i \leq k)$ ,输出位为 1 表示对应位为符号位。

$$\begin{cases} LZA_0 = m_k \oplus \bar{S} & (k=0) \\ LZA_i = (m_{k-i} \oplus \bar{S}) LZA_{i-1} & (1 \leq i \leq k) \end{cases} \quad (6)$$

其中,符号  $\oplus$  表示异或逻辑操作,  $\bar{S}$  表示  $S$  的逻辑非。

采用桶形移位方法实现两个浮点数相加需要 663 个 slices,采用基于数据选择器的方法实现两个浮点数相加需要 277 个 slices。所以,采用基于数据选择器的方法实现加法能有效节约资源、降低功耗。

## 2 FPGA 实现

### 2.1 处理器总体设计

基于 FPGA 实现的浮点 FFT/IFFT 处理器结构如图 1 所示。

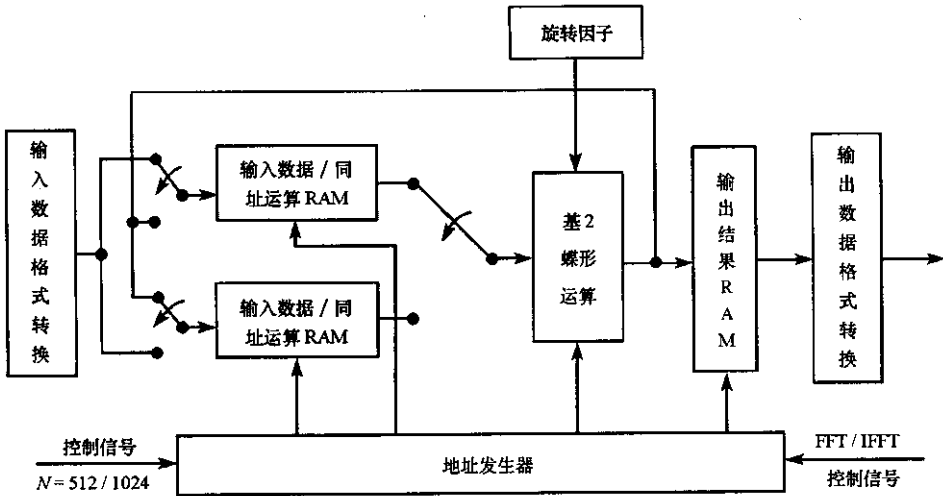


图 1 处理器结构

Fig. 1 Architecture of processor

该处理器有以下特点：

- (1) 处理器内部采用 24-bit 自定义浮点格式,能够兼顾处理器的处理性能和资源占用;
- (2) 采用基 2 蝶形运算单元结构,减少 FPGA 资源占用;
- (3) 内置三组数据存储器,保证蝶形运算全速工作,提高处理器处理能力;
- (4) 处理器进行 FFT 运算或 IFFT 运算以及运算长度  $N=512$  或  $N=1024$  由外部控制信号决定;
- (5) 自定义浮点格式的旋转因子以只读 RAM 的方式存储在 FPGA 片上资源 Block RAMs 内。

### 2.2 FFT 处理器性能

FPGA 选择 Xilinx 公司的产品 XC2V1000,其片内资源丰富,包括:5120 个可配置逻辑单元(slices) 40 个 18bit  $\times$  18bit 有符号乘法器,720Kb RAM 存储器等,能够满足 FFT 处理器硬件实现对资源占用的要求。

该处理器已完成并投入实用,运行结果正确,性能稳定。处理器蝶形运算时钟 80MHz,完成 512

点 FFT/IFFT 时间只需  $29\mu\text{s}$  ,完成 1024 点 FFT/IFFT 时间只需  $64\mu\text{s}$  ,处理误差小于  $-80\text{dB}$  ,功耗小于  $1\text{W}$  。整个处理器设计占用 FPGA 资源情况如下 :

Number of Slices	2406	out of	5120	47%
Number of Block RAMs	16	out of	40	40%
Number of MULT18X18s	8	out of	40	20%

### 3 结束语

针对定点 FFT 处理器精度低的不足 ,提出并实现了一个自定义浮点格式的 FFT 处理器。在设计与实现的过程中 ,关键是结合 FFT 算法和 FPGA 结构特点 ,减少资源开销、提高处理速度、降低功耗。

虽然 ,该处理器运算速度较通用 DSP 提高了很多 ,但这还不能满足某些高速数字信号处理系统的要求 ,所以还应该在提高处理器处理能力方面开展更深入的研究。

### 参考文献 :

- [ 1 ] Shaditalab. Self-sorting Radix-2 FFT on FPGAs Using Parallel Pipelined Distributed Arithmetic Blocks[ A ]. IEEE Symposium on FPGAs , Montreal( Canada ) ,1998 337 - 338.
- [ 2 ] Perez-Pascual. FPGA-based Radix-4 Butterflies for Hiperlan/2[ A ]. IEEE International Symposium on Circuits and Systems , Valencia ( Spain ) 2002 277 - 280.
- [ 3 ] 刘朝辉 ,韩月秋 . 用 FPGA 实现 FFT 的研究 [ J ]. 北京理工大学学报 ,1999 ,12( 2 ) 234 - 238.
- [ 4 ] 清华大学无线电系快速傅立叶变换组 . 快速傅立叶变换 [ M ]. 北京 :人民邮电出版社 ,1980.
- [ 5 ] 胡振华 . VHDL 与 FPGA 设计 [ M ]. 北京 :中国铁道出版社 2003.

( 上接第 60 页 )

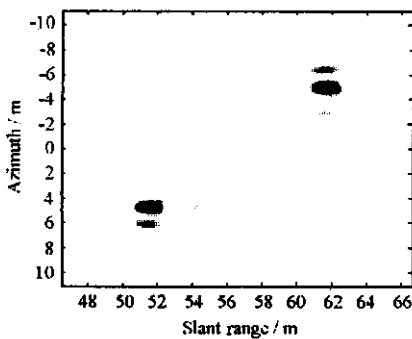


图 5 采用角反射器辐射校准后的图像

ig. 5 SAR image calibrated using a triangle reflector

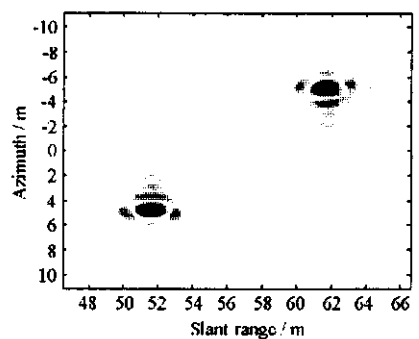


图 6 采用理想点目标辐射校准后的图像

Fig. 6 SAR image calibrated using an ideal point target

### 3 结论

低频超宽带雷达的工作带宽和工作波长与常规高频窄带雷达有很大区别 ,因而不能适用常规雷达的辐射校准技术。低频特性要求目标的电磁模型为低频散射模型 ,宽带特性要求该散射模型必须是频率的函数。在构造补偿滤波器的时候必须考虑目标的宽带散射特性 ,才可以得到较好的辐射校准效果。

### 参考文献 :

- [ 1 ] Freeman A. SAR Calibration : An Overview[ J ]. IEEE Trans. GRS ,1992 1107 - 1121.
- [ 2 ] Rao S M , Wilton D R , Glisson A W. Electromagnetic Scattering by Surfaces of Arbitrary Shap[ J ]. IEEE Trans. AP. 1982 ,AP - 30 : 409 - 418.
- [ 3 ] Cockrell C R , Beck F B. Asymptotic Waveform Evaluation ( AWE ) Technique for Frequency Domain Electromagnetic Analysis[ R ]. NASA Technical Memorandum 110292 ,1996.
- [ 4 ] 黎海涛 ,徐继麟 . 超宽带雷达方程 [ J ]. 电子科技大学学报 2000 ( 2 ) .



