

基于 FPGA 的 IFFT 处理器设计

郑华晋 陈友武

(南京邮电大学 通信与信息工程学院 南京 210046)

摘要: 本文首先通过分析 FFT 的 Cooley-Tukey 算法, 从而导出相应的 IFFT 算法。本文采用 Altera 公司的 Cyclone II 系列 FPGA 芯片中的 FFT megacore IP 核来定制 IFFT 功能, 最后分别使用 Quartus II 和 VHDL 开发工具验证实现。

关键词: FPGA IFFT Cooley-Tukey VHDL

FPGA-Based IFFT Processor Design

Zheng Huajin, Chen Youwu

(College of Communication & Information Engineering, Nanjing University of Posts & Telecommunications, Nanjing, 210046)

Abstract: Based on the analysis of the Cooley-Tukey algorithm of FFT, a IFFT algorithm is described. This paper presents an IP core named megacore to customize IFFT function by using the Cyclone II FPGA chip from Altera. The algorithm is verified in the Quartus II development tool and VHDL separately.

Keywords: FPGA IFFT Cooley-Tukey VHDL

1、引言:

数字信号处理(Digital Signal Processing, 简称 DSP)是现代工程学科分析的常用手段。它从两个方面解决信号处理的问题: 一为时域分析, 即数字滤波; 二为频域分析, 即从频谱角度分析信号特征。其中, 较常用的频域分析方法包括离散傅里叶变换 (DFT), 这是信号处理中最常用的处理方法。

作为最常用的分析手段, DFT 在各种数字信号处理中起着核心作用, 而它的快速算法—FFT (Fast Fourier Transform) 的提出, 从硬件方面解决了数字信号处理实现和应用的瓶颈。现代信号处理要求高速、大容量数据流的实时处理, 关键在于系统的输入、处理和输出等各个阶段都具有时间限制, 要求处理单元具有高速度、大容量和实时处理的性能。因此, 高速、灵活地实现 FFT 及其逆变换 IFFT 变得越来越重要。在 FPGA 技术发展迅速、器件的规模和品种不断增加、FPGA 器件及其应用成本不断降低的大背景下, 在越来越多的领域和产品中, 采用 FPGA(现场可编程门阵列)为代表的现场可编程硬件来实现数字信号处理, 已成为工业发展的趋势。

FPGA 技术有如下优点:

- 1、可利用 EDA 工具来缩短开发周期, 降低设计成本, 并能够并行处理数据;
- 2、易实现流水线结构, 且升级简便, 设计的灵活性较高。

在业界对基于 FPGA 的 FFT 算法研究较成熟的情况下, 本文尝试研究基于 FPGA 的 FFT 的反变换——IFFT。本文研究基于 FPGA 芯片的 IFFT, 将 IFFT 与 FPGA 芯片设计相结合, 进行基于 FPGA 的 IFFT 处理器研究。

2、算法设计思想:

2.1、FFT 算法原理:

假设有长度为 L 的时域信号 x, 以及经过 DFT 后得到的频域信号 X, 其形式表示如下:

$$x = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ \vdots \\ x_{L-1} \end{bmatrix} \xrightarrow{\text{DFT}} X = \begin{bmatrix} X_0 \\ X_1 \\ \vdots \\ \vdots \\ X_{N-1} \end{bmatrix}$$

具体等式如下：

$$X_k = \sum_{n=0}^{L-1} A_{kn} x_n, \quad k = 0, 1, \dots, N-1$$

其中， A_{kn} 意义如下：

$$A_{kn} = e^{-j2\pi kn/N} = W_N^{kn}, \quad 0 \leq k \leq N-1, 0 \leq n \leq L-1$$

由表达式可见，当 N 点数非常大的情况下，时域信号相应的 DFT 变换需要的 RAM 存储单元数量极大，使得运算的实现很不便。在此情况下，可用 FFT 变换来实现其从时域信号到离散频域信号的变换过程。常用的 FFT 方法包括 DIT (时域抽取法)、DIF (频域抽取法) 以及 Cooley-Tukey 算法等，其中前两种是基于标号分解的算法。本文重点讨论 Cooley-Tukey 算法，其基本要求是离散信号的维数是 2 的幂次方，即有如下关系：

$$N = 2^B \iff B = \log_2(N)$$

则在此基础上，FFT 的基本运算量为： $\frac{1}{2}NB = \frac{1}{2}N\log_2(N)$

对于离散时域信号 $x(n)$ ，令 $g(n) = x(2n)$ ， $h(n) = x(2n+1)$ ， $n = 0, 1, 2, \dots, \frac{N}{2}-1$

则有

$$G_k = \sum_{n=0}^{\frac{N}{2}-1} W_{\frac{N}{2}}^{kn} g(n) \quad k = 0, 1, 2, \dots, \frac{N}{2}-1$$

$$H_k = \sum_{n=0}^{\frac{N}{2}-1} W_{\frac{N}{2}}^{kn} h(n)$$

经过一系列复数变换，最终可得到如下的变换公式：

$$X(k) = G(k) + W_N^k H(k)$$

$$X(k + \frac{N}{2}) = G(k) - W_N^k H(k)$$

$$k = 0, 1, 2, \dots, \frac{N}{2}-1$$

由此可见，典型的 FFT 变换包括如下三个步骤：

- (1) 通过蝶形运算, 将输入的 N 点时域信号转换为 N 个单点的时域信号;
- (2) 进行 N 次单点的 DFT 运算;
- (3) 将 N 个单点的 DFT 信号合并, 得到 N 点的 DFT 信号。

以 $N=8$ 的信号为例, 给出具体的运算过程;

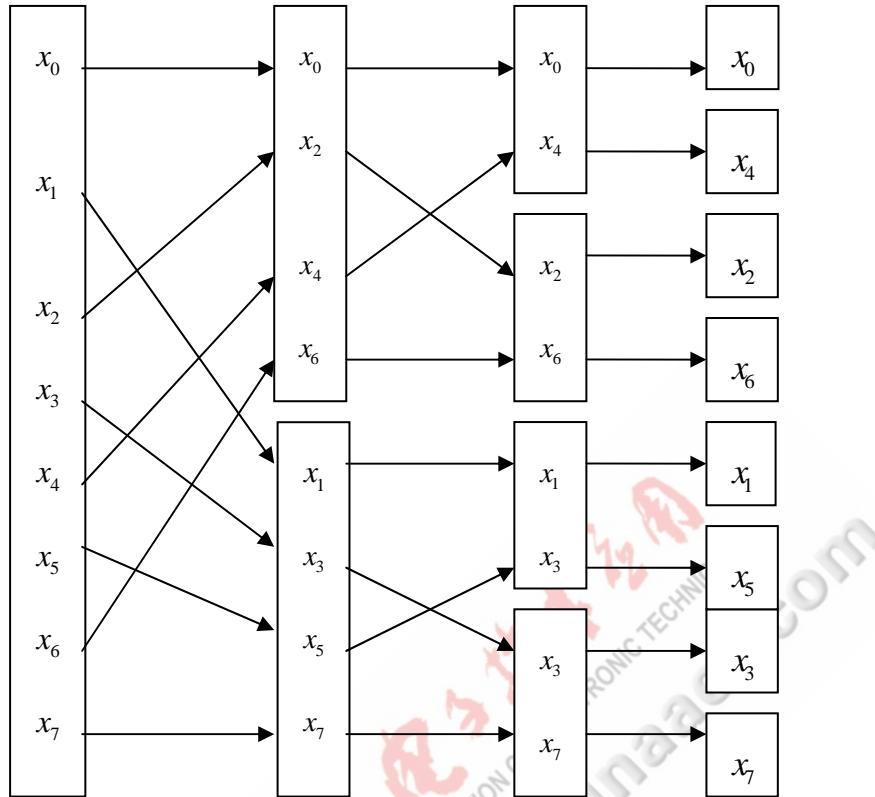


图 1 信号按奇偶分离过程

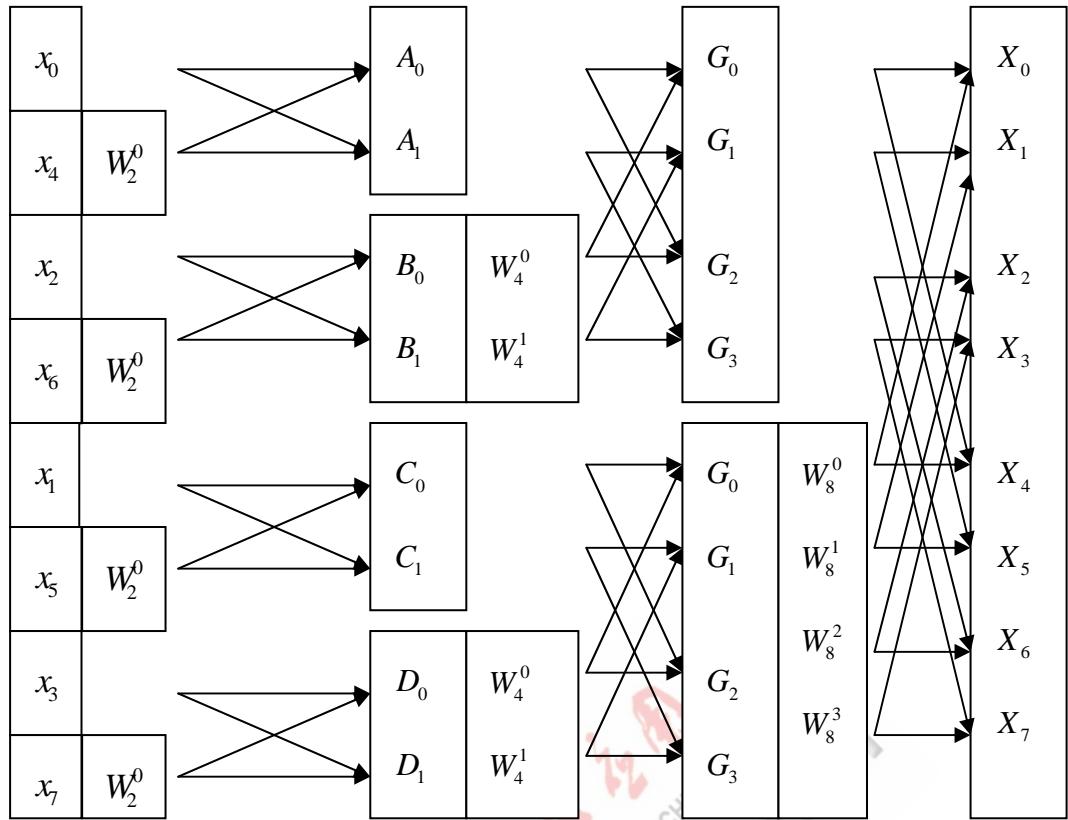


图 2 蝶形运算过程

根据上述的分析过程, 可知基 2 的 FFT 运算需要复数加法器、复数乘法器以及蝶形运算器来共同实现; 同时需要相应的 RAM 来存储运算结果, 以及 ROM 存储旋转因子 W_N^k 。上述硬件在 FPGA 中均可配置。

2. 2、IFFT 算法原理:

根据上述 DFT 及 FFT 的运算算法可知, 由于 $X = Ax = \tilde{A}\tilde{x}$

则由矩阵元算的逆变换, 可得

$$\tilde{x} = \tilde{A}^{-1}X, \text{ 其中 } \tilde{A}^{-1} \text{ 为 } \tilde{A} \text{ 的逆矩阵}$$

根据线性代数的定理, 有

$$\tilde{A}^{-1} = \frac{\tilde{A}^*}{N}, \text{ 其中 } \tilde{A}^* \text{ 为矩阵 } \tilde{A} \text{ 的伴随矩阵}$$

于是 FFT 运算的逆变换为

$$x = \frac{1}{N} [FFT(X^*)]^*$$

对比 FFT 运算的过程, 分析可知, 完全可用实现 FFT 的硬件电路来实现其 IFFT 功能。两者的区别仅在于在 IFFT 中, 需对输入信号先做共轭运算处理, 在输出端口前, 再增加进

行一次共轭运算，然后进行一个 $\frac{1}{N}$ (即总点数的倒数值) 的因子对信号进行衰减。最后得到的输出即为初始的时域信号 x 。

下面针对算法实现过程中的关键问题，进行讨论并给出解决方案。

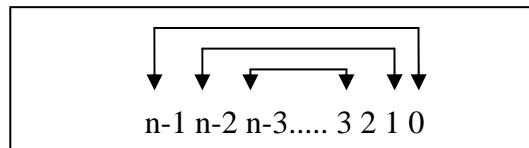
码位倒置算法：

由对输入的离散信号分奇偶分离过程中可看到，进入运算器时的 x_k 不再按输入的顺序排列。由于需寻址信号在 RAM 的存储地址，因此，对离散信号的寻址就尤为重要。可设计如下的算法来实现。以 $N=8$ 的 FFT 为例，分析 x_k ($k=0 \sim 7$) 的地址二进制数值，及分奇偶

后的 x_k 的排列值，得如下关系式：

次序	分奇偶前		分奇偶后	
	k	二进制值	k	二进制值
1	0	000	0	000
2	1	001	4	100
3	2	010	2	010
4	3	011	6	110
5	4	100	1	001
6	5	101	5	101
7	6	110	3	011
8	7	111	7	111

由此分析出，我们只要把输入的信号 x_k 的 k 的二进制值，进行变换，即首位和第二位调换，而第一位保持不变，即可得到相应的分奇偶后的结果。如对于 x_3 ，二进制值为 011。奇偶分离时第一位上的 1 不变，而首位 0 与第二位 1 互换，即得输出的地址值 110，即 x_6 。类似的，对于任意 n 位的地址值，变换规则为第 $(n-1)$ 位与第 0 位互换，第 $(n-2)$ 位与第一位互换，以此类推，即可得到输出序列地址。



倒置操作示意图

同址运算：

在进行运算处理时，我们考虑采用同址运算的思想，这是 FFT 算法在运算方式上的典型特点。研究蝶形运算的特点，可以看到计算结果只与相对应的两个输入数据相关，而且每个蝶形的输入输出节点均在同一水平线上；因此，不妨考虑，对于每个蝶形运算的输出数据，可将其存放在存储输入数据所对应的存储单元。即用输出的数据覆盖原来的输入数据所占的 RAM 存储区域。因此，只需要 N 个复数的存储单元。这种方法，不但可存放输入的原始数据，而且可存放中间结果，并且还可存放最终的运算结果。因此，同址运算方式节省了大量的存储单元，是 FFT 算法的一大优点。

3、硬件电路结构:

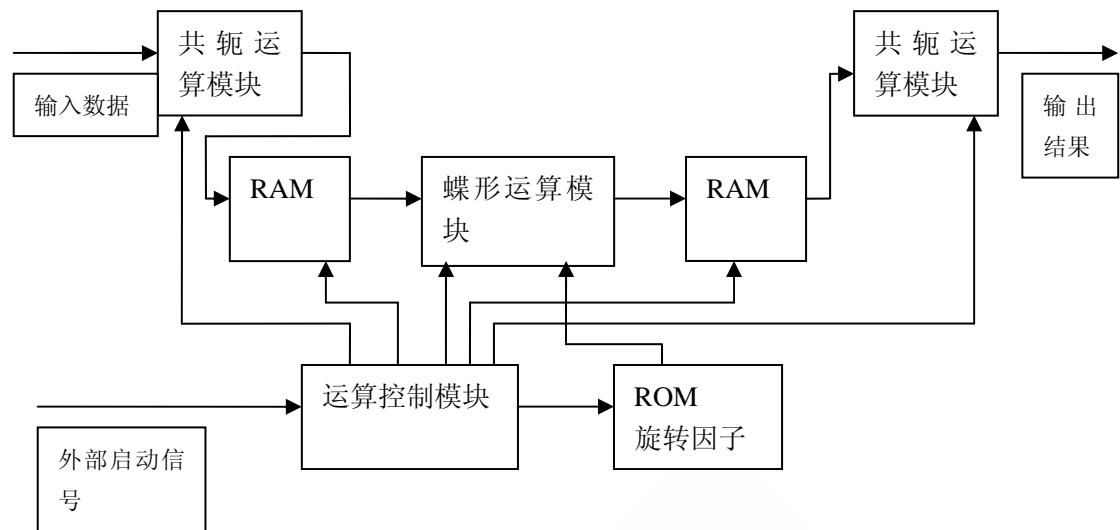


图 3 基本实现结构图

Altera 公司的 FPGA 产品内部具有高速数字信号处理(DSP)模块, 具有容量大、含高速 RAM 以及内部含有丰富的乘法器等优点。因此, 完全可利用 FPGA 来实现 IFFT 算法。

设计的电路结构主要包括以下模块: 作为缓冲器的双端口数据存储单元、蝶形运算单元、存放每级运算结果的双端口存储器; 进行共轭运算的共轭运算模块、存储旋转因子的 ROM、地址产生单元、对整个系统进行控制的控制模块。

下面介绍各模块的功能:

(1) 共轭运算模块——主要功能是对输入信号进行共轭运算, 这是实现 IFFT 的前提。运算后的信号返回 RAM, 进行基二运算。得到的结果再送共轭运算模块, 并乘上衰减因子后, 得到正确输出结果;

(2) RAM ——其功能是在运算控制模块的控制下, 将输入数据的实部和虚部存放到指定的地址单元; 并在控制模块的控制下, 将相应地址的数据送到蝶形运算模块, 作为蝶形运算模块的输入数据;

(3) 蝶形运算模块——蝶形运算由乘法器和加法器构成, 分别进行复乘和复加运算。由前述分析, N 点的 IFFT, 其运算可以分为 B 级 ($B = \log_2[N]$), 每一级需做 $N/2$ 次蝶形运算。

鉴于它们之间以及相邻级之间均为串行工作状态。因此, 每次运算只需调用同一个蝶形运算模块;

(4) ROM ——该模块用于存储并往蝶形运算单元送运算所需的旋转因子。IFFT 正是利用了系数的对称性和周期性实现把长点数变换转化为短点数变换, 从而提高运算效率。旋转因子由 Matlab 软件产生, 并转换为 32 位的定点数, 并保存为. hex 数据格式, 然后利用 Quartus 软件的 MegaWizard 工具设计 32 位的 ROM, 并将. hex 文件固化在其中, 这样就可以省去了计算旋转因子的时间。将旋转因子 W_N^k 写入到 ROM 中, 每次进行蝶形运算时, 由运算控制模块选择正确的旋转因子, 将其送入蝶形运算模块进行计算;

(5) 运算控制模块 ——控制模块是算法执行的核心单元, 负责产生 BAM 以及存放旋转因子的 ROM 的读、写使能信号和读、写地址, 并产生控制运算单元工作的信号。这一部分主

要由计数器、地址线交换器、寄存器和延时器组成。为节省资源，可对存储数据 RAM 采用原址读出原址写入的方法，即在进行下一级变换的同时，首先应将结果回写到读出数据的 RAM 存贮器中；而对于 ROM，则应采用与运算的数据相对应的方法来读出存储器中旋转因子的值。

4、仿真

4.1、Quartus II 仿真

设计过程采用 VHDL 语言描述,首先利用 Matlab 产生旋转因子,送 ROM 中存储。然后进行编程、布线及仿真,确保硬件结构达到设计要求。

我们利用 Quartus II 首先对加法单元、蝶算单元和地址产生单元分别综合，然后对整个设计综合优化，产生了一系列的输出文件，其中综合报告文件和 EDIF 网表文件时最重要的。实验结果证明系统吞吐量可以达到 $2M/ s$ 的变换速度，已经能够满足一般系统的设计要求，图 4 是系统仿真的结果。

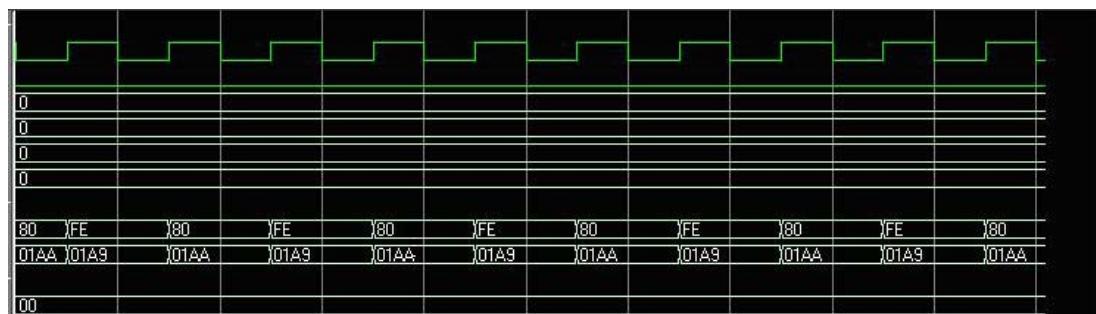


图4 系统仿真结果图

4.2、VHDL的编译与仿真

本设计中用MAX+PLUS II作为开发软件。MAX+PLUS II支持原理图、VHDL等多种语言文本文件。MAX+PLUS II可以进行功能仿真和时序仿真，能够产生精确的仿真结果。由MAX+PLUS II生成波形文件，波形如图5。

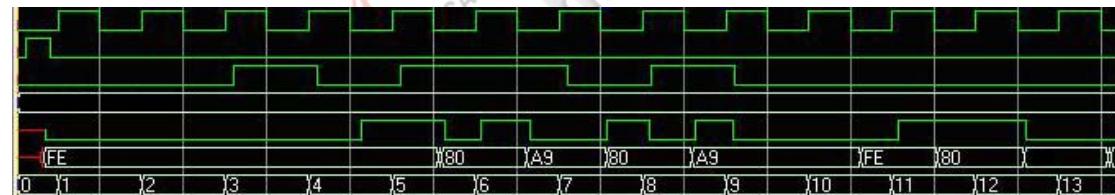


图5 蝶型运算单元波形

5、结束语

本文在分析快速傅里叶变换(FFT)算法的基础上,提出了一种IFFT算法的FPGA设计方法,描述了设计中的蝶形运算、地址生成及旋转因子的设计过程。根据IFFT算法中数据输入、旋转因子方面的规律,利用同址运算的思想,改进了结构设计,减少了蝶形运算模块和RAM的数量,提高了存储效率。硬件试验的结果表明,设计的IFFT处理器的运算结果基本满足要求,且具有较高的运算速度和效率。

参考文献:

- [1] Sophocles J.Orfanidis, 《Introduction To Signal Processing》, 北京: 清华大学出版社, 2007 年 7 月
 - [2] 王林泉, 皮亦鸣, 陈晓宁, 肖欣, 《基于 FPGA 的超高速 FFT 硬件实现》, 成都: 电子科技大学学报, 2005 年 4 月

- [3]高瞻, 庄圣贤, 王森林, 史琴, 《基 FPGA 的 FFT 处理器设计》, 设计新技术, 2005 年 10 月
- [4] 段玉波, 刘继新, 刘树庆, 《基于FPGA的快速傅立叶变换(FFT)处理器的设计》, 自动化技术与应用, 2006年

论文原创性声明

本人郑重声明: 此处所呈交的论文《基于 FPGA 的 IFFT 处理器设计》是本人在导师指导下进行研究工作所取得的研究成果。据我所知, 除了参考文献外, 论文中不包含其他人已经发表或撰写过的研究成果。本声明的法律结果将完全由本人承担。

作者签字: 郑华晋 陈友武

2008 年 5 月 25 日