基于串行数据的全并行FFT设计方案及性能评估报告

1. 设计方案

对于基为的FFT而言，可以利用蝶形电路实现计算，例如8点的FFT蝶形运算电路如图1所示。蝶形运算电路的好处是运算较快，结构简单，但随着的加大，硬件消耗增长非常快。为了使用蝶形电路计算点数较大的FFT，需要对蝶形运算电路做一定的改进以降低硬件资源的占用。

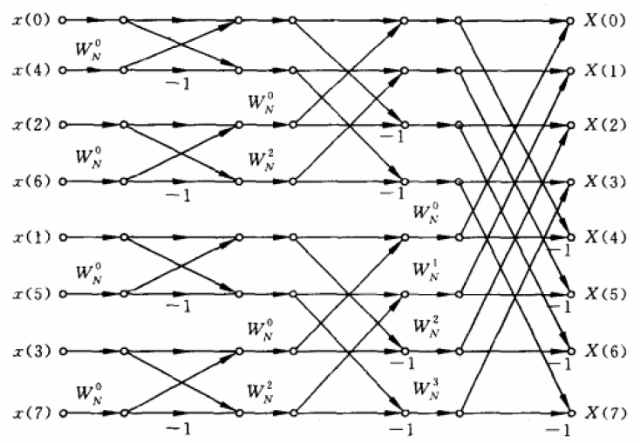


图1. 8点FFT蝶形运算电路

本设计基于蝶形运算电路，主要做出两点改进：

1. 数据改为串行输入。串行输入会带来两点好处，其一是电路只需要单比特全加器；其二是硬件电路只需要更改时序就可以适应不同位宽的输入数据。
2. 改进乘法器，降低乘法器资源消耗。由于蝶形运算电路中被乘数（旋转因子）都是固定的，因此对乘法器做特殊的优化可以极大的降低乘法器的资源开销。
3. 串行输入的优势



图2. 串行蝶形运算电路

如图2所示，串行的蝶形运算电路串行的输入和输出数据，可以在只更改时序的情况下适应不同位宽的数据输入。另外，由于只需要处理单比特运算，所以硬件电路中只需要1bit的加法器。

1. 乘法器的优化

在FFT运算中，由于被乘数（旋转因子）在设计电路时就已经固定，所以可以对乘法器作特殊优化以降低乘法器的硬件复杂度。

乘法器采用移位相加乘法器，移位乘法器的原理同手算乘法相同，如图3所示。可以知道，在被乘数固定的情况下，乘法器可以只实现部分的加法运算，即被乘数非零位对应的加法运算，达到降低乘法器硬件复杂度的效果。



图3. 移位加乘法器原理

乘法器的硬件复杂度与被乘数非零位的长度正相关，为了使被乘数的非零位长度达到最短，被乘数采用正则有符号数表示。

正则有符号数（canonic signed digit, CSD）与二进制补码相似，是一种权重码。不同的是二进制补码每个位或是0或是1，而CSD数扩充为{-1,0,1}。CSD数有一系列性质[1]：

1. CSD数不存在连续的两个非零位。
2. 一个数的CSD数表示中含有的非零位是最少的。
3. CSD数表示唯一。
4. 在区间[-1,1)中CSD数的非零位平均比二进制补码少33%。

例如在一个8点FFT运算中，涉及到对0.7071的常系数乘法：



0.707的二进制数为0.101101010000010，CSD数为1.0-0-01010000000（-表示-1），-0.707的CSD数为-.01010-0-00000-0。所以移位加的式子为：



电路实现上，信号进入之后分为原码和补码，取补码采用取反加1。假设第一拍数据到达，7拍到达时两个单比特全加器同时开始计算和，第8拍计算这两路的和，第9时刻开始计算与两路和的结果输出。



对每个CSD数统一截9位，其中一位是整数位，则乘法器的处理时延为9拍，以32点FFT为例，如果以二进制补码来表示，则有290个非零位，而CSD数表示时有208个非零位。