基于串行数据的全并行FFT设计方案及性能评估报告

袁健生

2015/9/23

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。

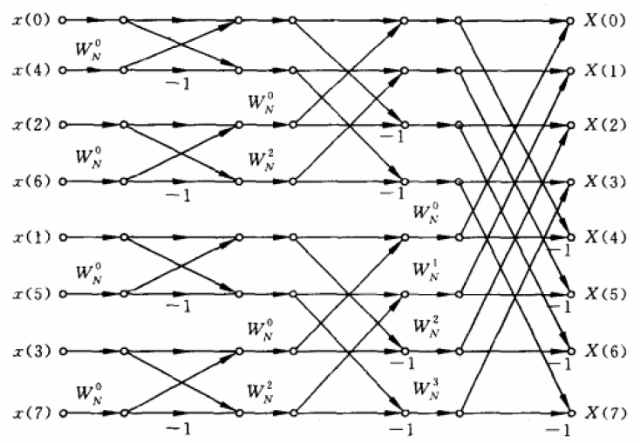


图 1. 8点FFT蝶形运算电路

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. 数据输入



图 2. 串行蝶形运算电路

本设计输入数据为定点数，位宽为16位，输入方式为串行输入。如图 2所示，串行的蝶形运算电路串行地输入和输出数据，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，所以串行的蝶形运算电路中只需要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，也是一种权重码。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}。在区间[-1,1)中的CSD数的非零位平均比二进制补码少33%。在本设计中采用CSD数来表示旋转因子以降低乘法器的硬件复杂度。



图 4. 移位加乘法器硬件结构

本设计的乘法器的硬件结构如图 4所示，从图 3中可以看出，被乘数固定后每一列相邻数的下标的差值固定，因此可以用一系列的D触发器缓存输入数据，并在对应的位置拿出缓存的数据相加得到各列的输出。

以如图 3所示的乘法运算为例，因为只有三个非零行，因此n=2；第一行与第三行下标差值为2，因此d1=2；第三行与第四行差值为1，因此d2=1。

1. 蝶形运算电路

16点全并行FFT蝶形运算电路如图 5所示。可见FFT算法将16点的FFT运算分解为多个4点的FFT运算。旋转因子定义为，为了便于描述，下面都使用的值来描述，有旋转因子矩阵：



可以看到矩阵第一行和第一列都为0，因此点FFT蝶形运算电路总共需要个复数乘法器。，分别为M点和N点FFT蝶形运算电路需要的复数乘法器数量。

本设计FFT运算点数为2048点，拆分为结构。



1. 全并行运算电路



(b) 串行运算电路

图 5. 16点FFT蝶形运算电路

1. 性能评估

经过改进后的乘法器每31拍输出16位数据，并且根据被乘数的不同，乘法器的性能很可能略高于该值（略去尾部的零部分的计算）。忽略加法器以及各级运算电路之间的延时，则n级FFT电路每31n拍输出所有的16位计算结果。

根据被乘数的不同，乘法器消耗的硬件资源也不相同。设被乘数的CSD表示中有m个非零位且分别位于第x1x2…xq位，则该乘法器需要m-1个全加器和1个半加器，xq-x1+m个寄存器(其中xq-x1个寄存器用于缓存输入数据，m个寄存器用于缓存进位位)。记乘法器的平均硬件消耗为10个1bit全加器，1个1bit半加器，30个1bit寄存器。

略去各级电路的寄存器消耗，的FFT运算电路大约需要8451个乘法器。总共需要110k个1bit全加器，300k个1bit寄存器。