存内计算下的FFT分级计算

刘润

2022 年 6 月 5 日

1 FFT混合基分解

1.1 FFT混合基分解一般形式

对于N点DFT，如果N是一个复合数，它可以分解成一些因子的乘积，则可以用FTT的一般算法，即混合基FFT算法，而基-2算法只是这种一般算法的特例。

若N可以表示为复合数$N = r_1 r_2 \cdots r_L$，则对于$n < r_1 r_2 \cdots r_L$，的任何一个正整数n，可以按照 L基r_1，r_2，$\cdots$，r_L 表示为多基多进制形式$(n_{L-1} n_{L-2} \cdots n_1 n_0)_{r_1 r_2 \cdots r_L}$，这一多基多进制所代表的数值为：

$$(n)_{10} = n_{L-1}(r_2 r_3 \cdots r_L) + n_{L-2}(r_3 r_4 \cdots r_L) + \cdots + n_1 n_L + n_0 \tag{1}$$

其倒位序形式为$[\rho(n)]_{r_L r_{L-1} \cdots r_2 r_1} = (n_0 n_1 \cdots n_{L-2} n_{L-1})_{r_L r_{L-1} \cdots r_2 r_1}$，它所代表的数值为：

$$[\rho(n)]_{10} = n_0(r_1 r_2 \cdots r_{L-1}) + n_1(r_1 r_2 \cdots r_{L-2}) + \cdots + n_{L-2} r_1 + n_{L-1} \tag{2}$$

在这一多基多进制的表示中

$$\begin{aligned} & n_0 = 0, 1, \cdots, r_L - 1 \\ & n_1 = 0, 1, \cdots, r_{L-1} - 1 \\ & \cdots \\ & n_{L-1} = 0, 1, \cdots, r_1 - 1 \end{aligned} \tag{3}$$

1.2 512点FFT示例

下面以512点FFT作为示例讲解，将512点FFT分解为$16*16*2$三级，即$N=16*16*2$，则$r_1=16$，$r_2=16$，$r_3=2$，$L=3$，则有：

$$n = n_2(r_2r_3) + n_1r_3 + n_0 = 32n_2 + 2n_1 + n_0 \tag{4}$$

$$k = k_0(r_1r_2) + k_1r_1 + k_2 = 256k_0 + 16k_1 + k_2 \tag{5}$$

$$\begin{aligned} &n_0, k_0 : 0 \sim 1(r_3) \\ &n_1, k_1 : 0 \sim 15(r_2) \\ &n_2, k_2 : 0 \sim 15(r_1) \end{aligned} \tag{6}$$

所以根据DFT公式$X(k)=\sum\limits_{n=0}^{511} x(n)W_{512}^{nk}$，则将公式(4) $\sim$ (6)带入可得：

$$\begin{aligned} X(k) &= \sum_{n=0}^{511} x(n)W_{512}^{nk} \\ &= \sum_{n_0=0}^{1}\sum_{n_1=0}^{15}\sum_{n_2=0}^{15} x(n_2,n_1,n_0)W_{512}^{(32n_2+2n_1+n_0)(256k_0+16k_1+k_2)} \end{aligned} \tag{7}$$

可以得到

$$\begin{aligned} X(k) = \sum_{n_0=0}^{1}\sum_{n_1=0}^{15}\sum_{n_2=0}^{15} x(n_2,n_1,n_0)&W_{512}^{8192n_2k_0}W_{512}^{512n_2k_1}W_{512}^{32n_2k_2} \\ &W_{512}^{512n_1k_0}W_{512}^{32n_1k_1}W_{512}^{2n_1k_2} \\ &W_{512}^{256n_0k_0}W_{512}^{16n_0k_1}W_{512}^{n_0k_2} \end{aligned} \tag{8}$$

$$
\begin{aligned}
X(k) &= \sum_{n_0=0}^{1}\sum_{n_1=0}^{15}\sum_{n_2=0}^{15} x(n_2,n_1,n_0) W_{512}^{32n_2k_2} W_{512}^{32n_1k_1} W_{512}^{2n_1k_2} W_{512}^{256n_0k_0} W_{512}^{16n_0k_1} W_{512}^{n_0k_2} \\
&= \sum_{n_0=0}^{1}\sum_{n_1=0}^{15}[\sum_{n_2=0}^{15} x(n_2,n_1,n_0) W_{16}^{n_2k_2}] W_{512}^{32n_1k_1} W_{512}^{2n_1k_2} W_{512}^{256n_0k_0} W_{512}^{16n_0k_1} W_{512}^{n_0k_2} \\
&= \sum_{n_0=0}^{1}\sum_{n_1=0}^{15}[X_1(k_2,n_1,n_0) W_{512}^{(2n_1+n_0)k_2}] W_{512}^{32n_1k_1} W_{512}^{16n_0k_1} W_{512}^{256n_0k_0} \\
&= \sum_{n_0=0}^{1}[\sum_{n_1=0}^{15} X_1'(k_2,n_1,n_0) W_{16}^{n_1k_1}] W_{512}^{16n_0k_1} W_{512}^{256n_0k_0} \\
&= \sum_{n_0=0}^{1}[X_2(k_2,k_1,n_0) W_{512}^{16n_0k_1}] W_{512}^{256n_0k_0} \\
&= \sum_{n_0=0}^{1} X_2'(k_2,k_1,n_0) W_{2}^{n_0k_0} \\
&= X(k_2,k_1,k_0)
\end{aligned}
\tag{9}
$$

并且有

$$
\begin{aligned}
X_1(k_2,n_1,n_0) &= \sum_{n_2=0}^{15} x(n_2,n_1,n_0) W_{16}^{n_2k_2} \\
X_1'(k_2,n_1,n_0) &= X_1(k_2,n_1,n_0) W_{512}^{(2n_1+n_0)k_2}
\end{aligned}
\tag{10}
$$

$$
\begin{aligned}
X_2(k_2,k_1,n_0) &= \sum_{n_1=0}^{15} X_1'(k_2,n_1,n_0) W_{16}^{n_1k_1} \\
X_2'(k_2,k_1,n_0) &= X_2(k_2,k_1,n_0) W_{512}^{16n_0k_1}
\end{aligned}
\tag{11}
$$

$$
X(k_2,k_1,k_0) = \sum_{n_0=0}^{1} X_2'(k_2,k_1,n_0) W_{2}^{n_0k_0} \tag{12}
$$

可以很明显看出，512点的FFT可以拆解为16*16*2的三级FFT的级联，公式(10)为第1级16点FFT，公式(11)为第2级16点FFT，公式(12)为第3级2点FFT。

2 计算过程

2.1 理论计算流程

根据以上公式，可以得出结论，512点FFT计算可以分解为三级较小规模的FFT计算，并且通过公式(4) $\sim$ (6)可以得知原始信号的输入次序和计算得到信号的输出次序。原始信号为$x(0), x(1), x(2), ..., x(511), x(512)$，输入时读入的次序为$n = 32n_2 + 2n_1 + n_0$。

n2, n1, n0	n=32*n2+2*n1+n0						
0~15, 0, 0	x(0)	x(32)	x(64)	x(96)	⋯	x(448)	x(480)
0~15, 0, 1	x(1)	x(33)	x(65)	x(97)	⋯	x(449)	x(481)
0~15, 1, 0	x(2)	x(34)	x(66)	x(98)	⋯	x(450)	x(482)
0~15, 1, 1	x(3)	x(35)	x(67)	x(99)	⋯	x(451)	x(483)
0~15, 2, 0	x(4)	x(36)	x(68)	x(100)	⋯	x(452)	x(484)
0~15, 2, 1	x(5)	x(37)	x(69)	x(101)	⋯	x(453)	x(485)
⋮		⋮				⋮	
0~15, 14, 1	x(29)	x(61)	x(93)	x(125)	⋯	x(477)	x(509)
0~15, 15, 0	x(30)	x(62)	x(94)	x(126)	⋯	x(478)	x(510)
0~15, 15, 1	x(31)	x(63)	x(95)	x(127)	⋯	x(479)	x(511)

图 1: 第一级FFT16输入次序

原始信号以行为单位并行输入16个数据，作为第一级FFT计算。根据式(10)可以得知，第一级FFT16个数据并行输入，和存储在 FLASH模块中的旋转因子相乘，经过ADC采样之后，得到X_1，再与$W_{512}^{(2n_1+n_0)k_2}$相乘完成第一级FFT。所以对于一个 FFT16计算模块来说，以第一行数据为例，其矩阵运算模式为：

$$\begin{bmatrix} X_1(0) \\ X_1(32) \\ \vdots \\ X_1(480) \end{bmatrix} \begin{bmatrix} W_{16}^0 & W_{16}^0 & W_{16}^0 & \dots & W_{16}^0 \\ W_{16}^0 & W_{16}^{1\times1} & W_{16}^{1\times2} & \dots & W_{16}^{1\times15} \\ \vdots & \vdots & \vdots & & \vdots \\ W_{16}^0 & W_{16}^{15\times1} & W_{16}^{15\times2} & \dots & W_{16}^{15\times15} \end{bmatrix} \begin{bmatrix} x(0) \\ x(32) \\ \vdots \\ x(480) \end{bmatrix} \tag{13}$$

再将得到的X_1向量与$W_{512}^{(2n_1+n_0)k_2}$向量元素对应相乘即可。

$$\begin{bmatrix} X_1(0) \\ X_1(32) \\ \vdots \\ X_1(480) \end{bmatrix} * \begin{bmatrix} W_{512}^{(2*0+0)*0} \\ W_{512}^{(2*0+0)*1} \\ \vdots \\ W_{512}^{(2*0+0)*15} \end{bmatrix} \rightarrow \begin{bmatrix} X_1'(0) \\ X_1'(32) \\ \vdots \\ X_1'(480) \end{bmatrix} \tag{14}$$

重复以上过程，完成第一级FFT16计算。完成之后可以进行第二级FFT16计算，还是按照$n = 32n_2 + 2n_1 + n_0$公式，只不过此时信号输入数据为第一级FFT16计算得到结果，并且次序按照n_1进行。将红框内的数据并行输入，进行第二级FFT16的计算。

n2, n1, n0	n=32*n2+2*n1+n0						
0~15, 0, 0	$x_1'(0)$	$x_1'(32)$	$x_1'(64)$	$x_1'(96)$	···	$x_1'(448)$	$x_1'(480)$
0~15, 0, 1	$x_1'(1)$	$x_1'(33)$	$x_1'(65)$	$x_1'(97)$	···	$x_1'(449)$	$x_1'(481)$
0~15, 1, 0	$x_1'(2)$	$x_1'(34)$	$x_1'(66)$	$x_1'(98)$	···	$x_1'(450)$	$x_1'(482)$
0~15, 1, 1	$x_1'(3)$	$x_1'(35)$	$x_1'(67)$	$x_1'(99)$	···	$x_1'(451)$	$x_1'(483)$
0~15, 2, 0	$x_1'(4)$	$x_1'(36)$	$x_1'(68)$	$x_1'(100)$	···	$x_1'(452)$	$x_1'(484)$
0~15, 2, 1	$x_1'(5)$	$x_1'(37)$	$x_1'(69)$	$x_1'(101)$	···	$x_1'(453)$	$x_1'(485)$
⋮			⋮			⋮	
0~15, 14, 1	$x_1'(29)$	$x_1'(61)$	$x_1'(93)$	$x_1'(125)$	···	$x_1'(477)$	$x_1'(509)$
0~15, 15, 0	$x_1'(30)$	$x_1'(62)$	$x_1'(94)$	$x_1'(126)$	···	$x_1'(478)$	$x_1'(510)$
0~15, 15, 1	$x_1'(31)$	$x_1'(63)$	$x_1'(95)$	$x_1'(127)$	···	$x_1'(479)$	$x_1'(511)$

图 2: 第二级FFT16输入次序

$$\begin{bmatrix} X_2(0) \\ X_2(2) \\ \vdots \\ X_2(30) \end{bmatrix} \begin{bmatrix} W_{16}^0 & W_{16}^0 & W_{16}^0 & \ldots & W_{16}^0 \\ W_{16}^0 & W_{16}^{1\times1} & W_{16}^{1\times2} & \ldots & W_{16}^{1\times15} \\ \vdots & \vdots & \vdots & & \vdots \\ W_{16}^0 & W_{16}^{15\times1} & W_{16}^{15\times2} & \ldots & W_{16}^{15\times15} \end{bmatrix} \begin{bmatrix} X_1'(0) \\ X_1'(2) \\ \vdots \\ X_1'(30) \end{bmatrix} \tag{15}$$

$$\begin{bmatrix} X_2(0) \\ X_2(2) \\ \vdots \\ X_2(30) \end{bmatrix} * \begin{bmatrix} W_{512}^{16*0*0} \\ W_{512}^{16*0*1} \\ \vdots \\ W_{512}^{16*0*15} \end{bmatrix} \rightarrow \begin{bmatrix} X_2'(0) \\ X_2'(2) \\ \vdots \\ X_2'(30) \end{bmatrix} \tag{16}$$

重复以上过程，可以完成第二级FFT16的计算

2.2 FFT16的存内计算方法

存内计算模式对于矩阵乘加计算比较友好，适合FFT这样的矩阵乘法运算。任然以第一级FFT16为例，其第一行原始信号数据输入后与旋转因子矩阵第一行元素对应相乘相加，得到第一个FFT16计算结果$X_1(0)$，其过程在FLASH块中的计算过程为下图所示。

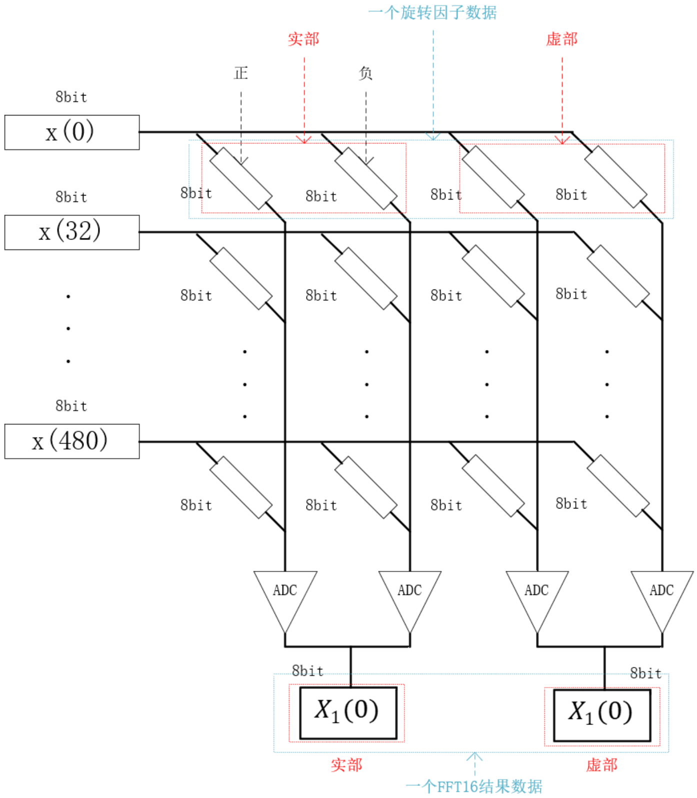

图 3: 一个FFT16数据在CIM中计算示意图

则对于一个完整的FFT16模块来说，其在CIM中的计算过程可以如下图所示。

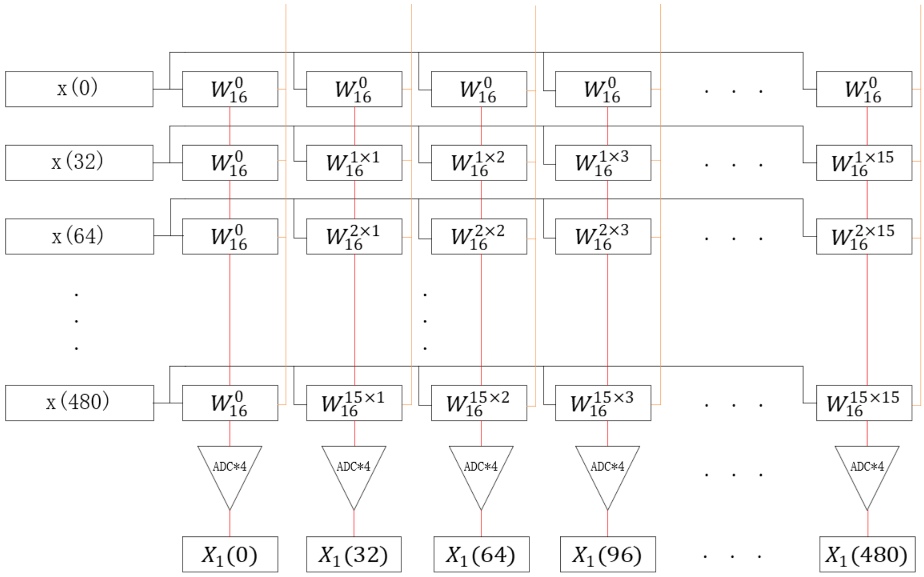

图 4: 一个FFT16模块在CIM中计算示意图

所以当有多个FFT16模块在FLASH Core中存在时，其结构图如下图所示。其中存储1中存储的是需要进行FFT计算的原始数据和在分级计算中产生的中间数据。从存储1模块中每次并行输出16个数据送入到一个FFT16模块中进行计算，在FLASH Core 中完成计算之后并行输出16个模拟结果经过ADC进行转换成为数字结果。随后存储2中根据公式(10)(11)(12)所产生的旋转因子和 FLASH core中产生的数字结果进行相乘，得到第一级的输出结果并且回存到存储1中，这样就完成了第一级FFT16的计算。在下图中的乘法器矩阵就是为了和旋转因子相乘而设置的。

由此，我们也可以得到一级FFT16的时序输出。 当我们重复上述时序之后，就会得到第二级FFT16时序，从而得到在CIM中FFT计算的完整时序。这是什么东西。这是什么鬼

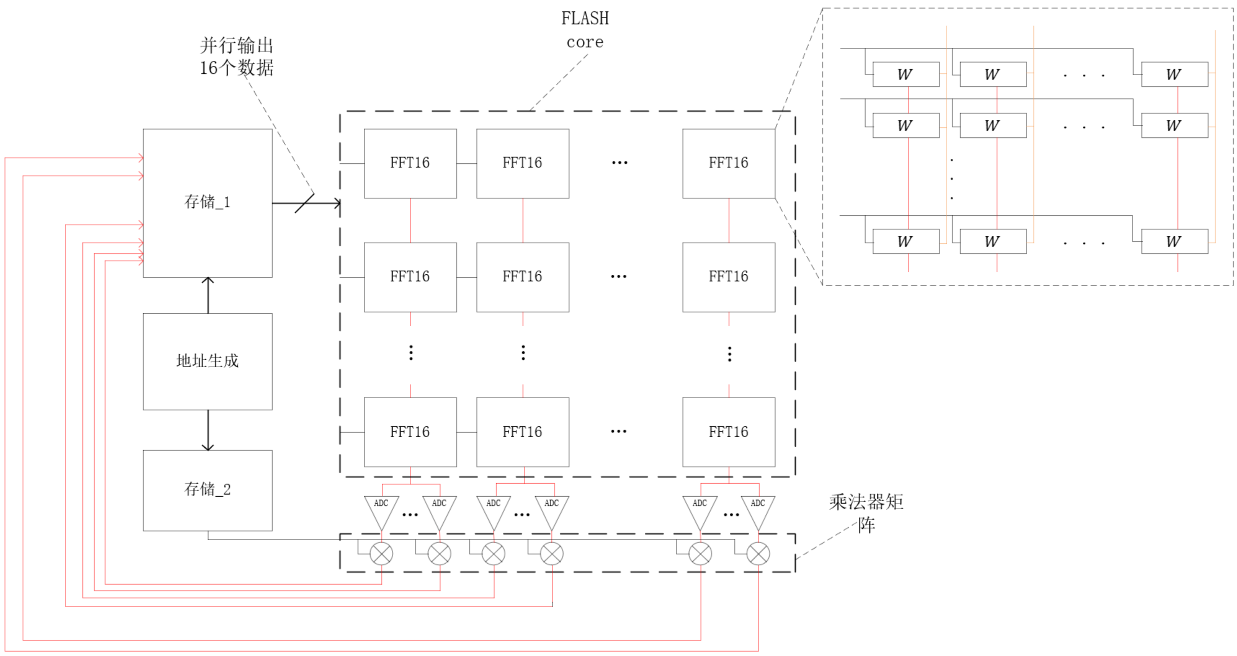

图 5: 多个FFT16模块在CIM中计算示意图

表 1: title

Specification	Proposed	[1](F)	[2](P)	[2]	[3]
FFT Size(N)	512	1024	1024	1024	4096
Technology	45nm	65nm	65nm	65nm	65nm
Vdd	1V	0.45V	0.45V	0.6V	1.2V
Word-length	8-bit	12-bit	12-bit	32-bit*	14-bit
Power(mW)	21.66	12	123	60.3	68.6

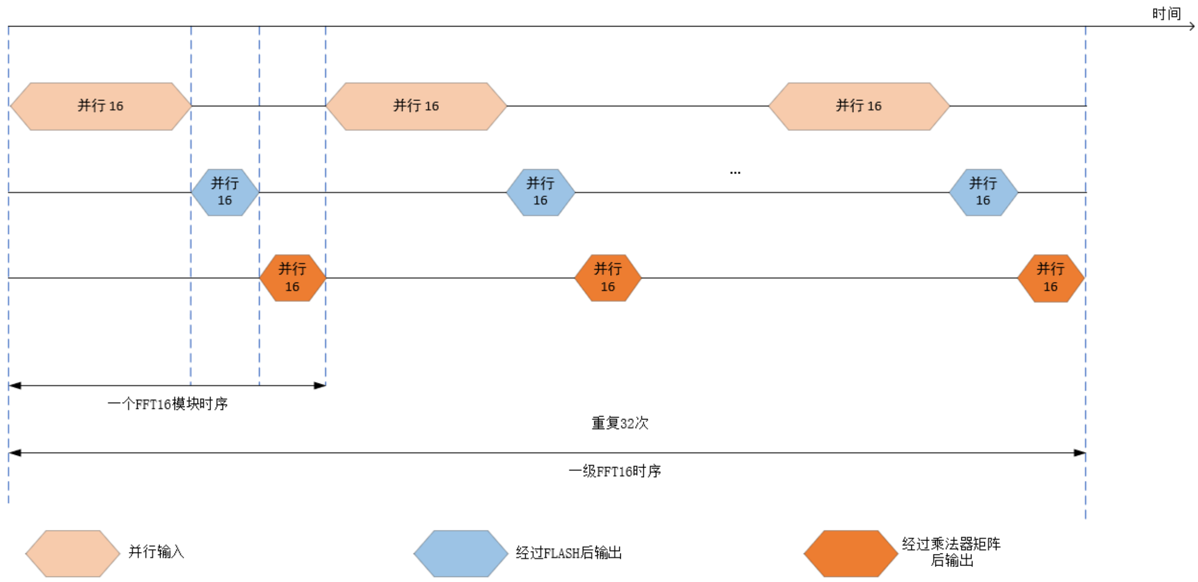

图 6: 一级FFT16计算时序示意图