

# 多DFT16模块并行的说明

刘润

2021 年 2 月 23 日

对于512点FFT，我们可以采用混合基分解的方式将其转变为两级DFT16，图1和图2展示了这种拆分方法。

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DFT16

从图1和图2中可以看出混合基分解之后的DFT16计算具有并行性，所以在存内计算的阵列中多个DFT16模块可以并行计算，用以加快计算速度。在图3中，每一个方块代表一个DFT16模块，而同一种颜色的DFT16模块将会同时运行，得到16个并行输出的结果。在图4中，给出了512点FFT分解

$n_2, n_1, n_0$	$n=32*n_2+2*n_1+n_0$						
$0 \sim 15, 0, 0$	$x_1'(0)$	$x_1'(32)$	$x_1'(64)$	$x_1'(96)$	...	$x_1'(448)$	$x_1'(480)$
$0 \sim 15, 0, 1$	$x_1'(1)$	$x_1'(33)$	$x_1'(65)$	$x_1'(97)$	...	$x_1'(449)$	$x_1'(481)$
$0 \sim 15, 1, 0$	$x_1'(2)$	$x_1'(34)$	$x_1'(66)$	$x_1'(98)$	...	$x_1'(450)$	$x_1'(482)$
$0 \sim 15, 1, 1$	$x_1'(3)$	$x_1'(35)$	$x_1'(67)$	$x_1'(99)$	...	$x_1'(451)$	$x_1'(483)$
$0 \sim 15, 2, 0$	$x_1'(4)$	$x_1'(36)$	$x_1'(68)$	$x_1'(100)$	...	$x_1'(452)$	$x_1'(484)$
$0 \sim 15, 2, 1$	$x_1'(5)$	$x_1'(37)$	$x_1'(69)$	$x_1'(101)$	...	$x_1'(453)$	$x_1'(485)$
$\vdots$		$\vdots$				$\vdots$	
$0 \sim 15, 14, 1$	$x_1'(29)$	$x_1'(61)$	$x_1'(93)$	$x_1'(125)$	...	$x_1'(477)$	$x_1'(509)$
$0 \sim 15, 15, 0$	$x_1'(30)$	$x_1'(62)$	$x_1'(94)$	$x_1'(126)$	...	$x_1'(478)$	$x_1'(510)$
$0 \sim 15, 15, 1$	$x_1'(31)$	$x_1'(63)$	$x_1'(95)$	$x_1'(127)$	...	$x_1'(479)$	$x_1'(511)$

图 2: 第二级DFT16

为两级DFT16之后计算的时序图。在图4中，每一个小方格代表一个DFT16模块，只有带有颜色的小方格才会处于计算状态，其余DFT16模块处于关闭状态。在前四个小图中，每张图代表着同一时刻有4个DFT16模块在进行计算，有64个数据在同时输入到CIM计算阵列中。在中间四个小图中，每张图代表着同一时刻有8个DFT16模块在进行计算，有128个数据在同时输入到CIM计算阵列中，这是因为这四张小图中不仅有第一级DFT16的计算模块，也包括第二级DFT16的计算模块，在此时，第一级和第二级实现了流水效果。在后四个小图中，每张图代表着同一时刻有4个DFT16模块在进行计算，有64个数据在同时输入到CIM计算阵列中，这是第二级DFT16的后四个计算时序。图4中小方格的颜色和图3中DFT16模块的颜色和图1图2 中数据被框起的虚线框颜色一一对应。在这种方式下，可以达到提升计算速度的功能，并且实现一定的流水线效果。

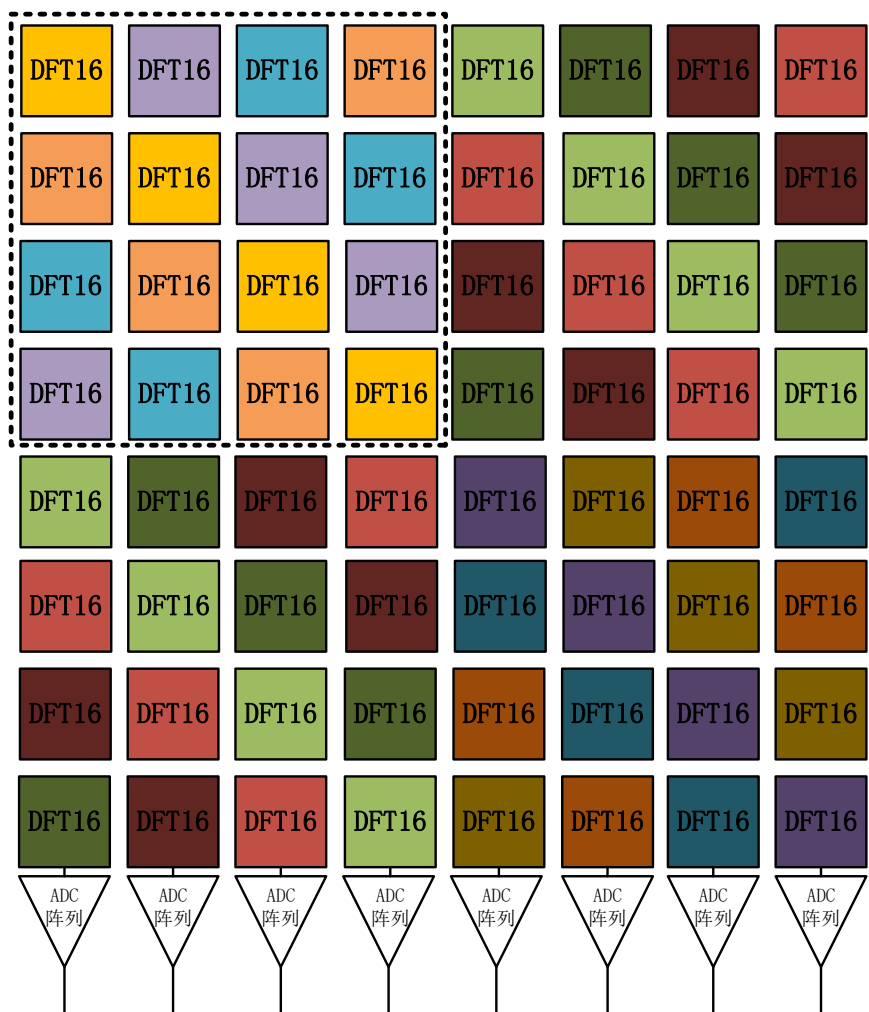


图 3: 在CIM块中的计算过程

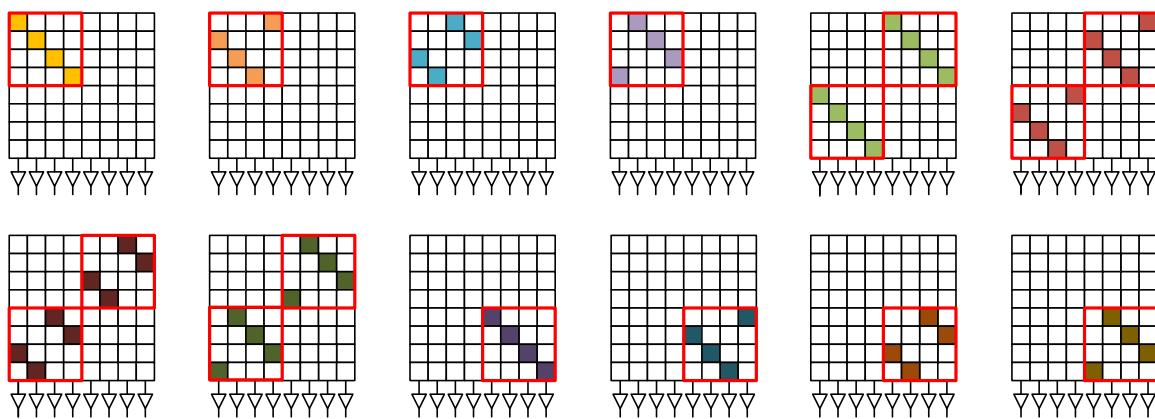


图 4: 计算时序