

1.问题来源

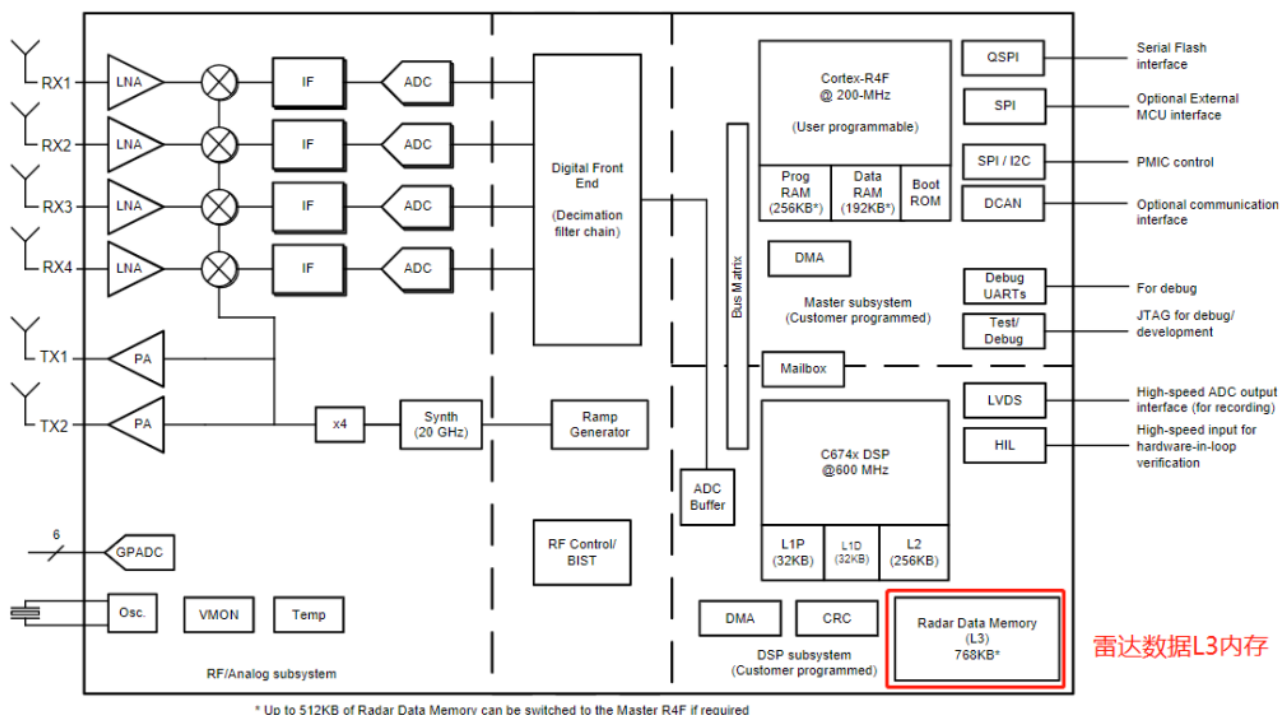
IWR1642有些不用DCA1000，通过USB串口也能把数据导出来？

提前注意事项：（1）串口数据采集不能做到实时采集，间隔时间还是比较大的，大约有3s左右；（2）串口数据采集每次只能采集最多768KB的数据，太多了内存装不下，具体为什么，后面会分析到。本文提供的采集方案是只采集128*128的一帧数据。

2.内存资源估计

首先，采集数据需要对芯片内部的资源进行估计，需要预算一下我们最多能够采集多少数据，这点至关重要。

串口采集的数据首先是存储在IWR1642的内存里的，IWR1642雷达数据内存（Radar Data Memory）大概是768KB,也就是DSP核内部存储块 L3资源总大小为768KB，主要是用来存储雷达数据。



进入TI官方提供的例程里面去看，L3内存的起始地址是0x200A0000，长度是0x000A0000，存储位宽是32位。

```

3 #if (L1P_CACHE_SIZE < 0x8000)
1   L1PSRAM:      o = 0x00E00000, l = (0x00008000 - L1P_CACHE_SIZE)
2 #endif
3 #if (L1D_CACHE_SIZE < 0x8000)
4   L1DSRAM:      o = 0x00F00000, l = (0x00008000 - L1D_CACHE_SIZE)
5 #endif
5   L2SRAM_UMAP1:  o = 0x007E0000, l = 0x00020000
7   L2SRAM_UMAP0:  o = 0x00800000, l = 0x00020000
3   L3SRAM:        o = 0x20000000, l = 0x000A0000
3   HSRAM:          o = 0x21080000, l = 0x8000
3

```

假设雷达参数配置为：发射天线：2个；接收天线：4个；一帧数据的chirp数：128个
一个chirp的ADC采样点数：128个 ADC位数：16bit 采样形式：
IQ正交采样（两个十进制数组成一个复数，I+jQ），因此计算数据量大小要乘以2。

ADC采样率为：2.5Msps。

那么最终采集到的数据大小为：

$$\frac{2 * 4 * 128 * 128 * 16 * 2 (bit)}{8 * 1024} = 512 (KB)$$

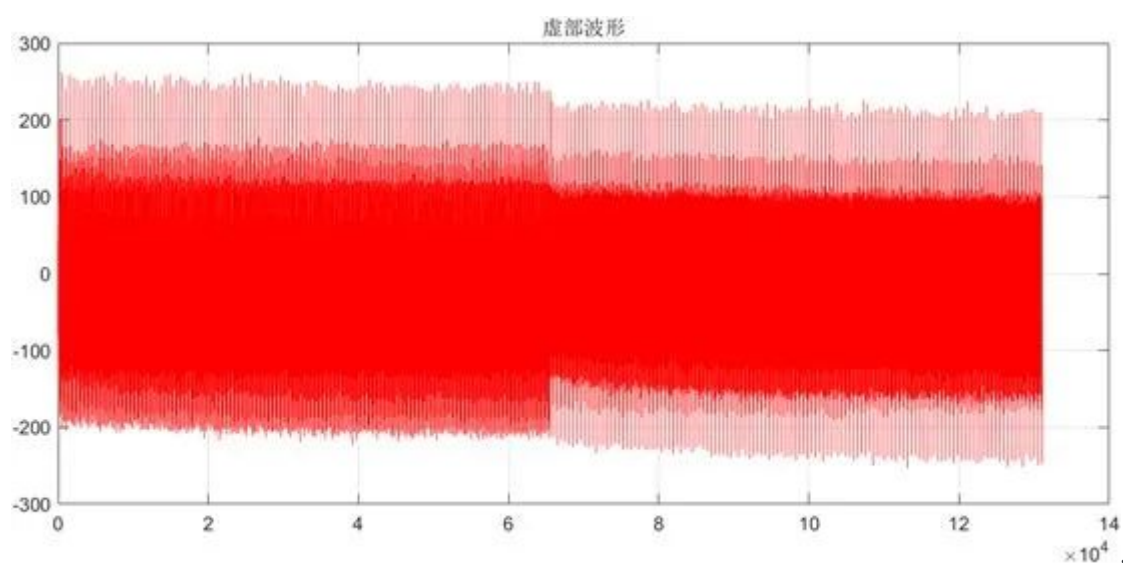
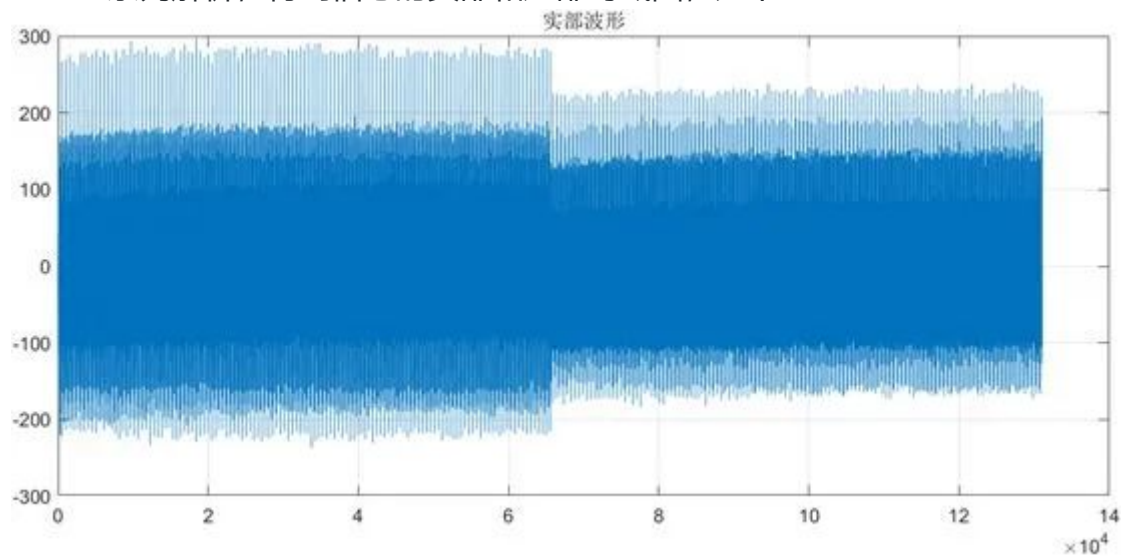
因此经过对内存的资源估计，如果按照上述参数，发现最多也就只能存储一帧的数据。如果将原始数据的chirp数和一个chirp的ADC采样点数降低，降低几倍就能够多采集几帧，这样也能够得到多帧的数据，只是相应的雷达性能会下降。另外，如果不按照2的次幂采集chirp数或者ADC采样点数，只要总的文件大小没有超过768KB，也能得到更多的数据，这点有待读者自行探索。虽然一帧的数据不多，也不能像DCA1000一样实时地长时间采集，但是对于初学者来说用来做毫米波雷达的学习或者实验教学是绰绰有余，基本上可学习到、测距、测速、测角、CFAR以及聚类算法。

3.雷达参数配置与数据格式说明

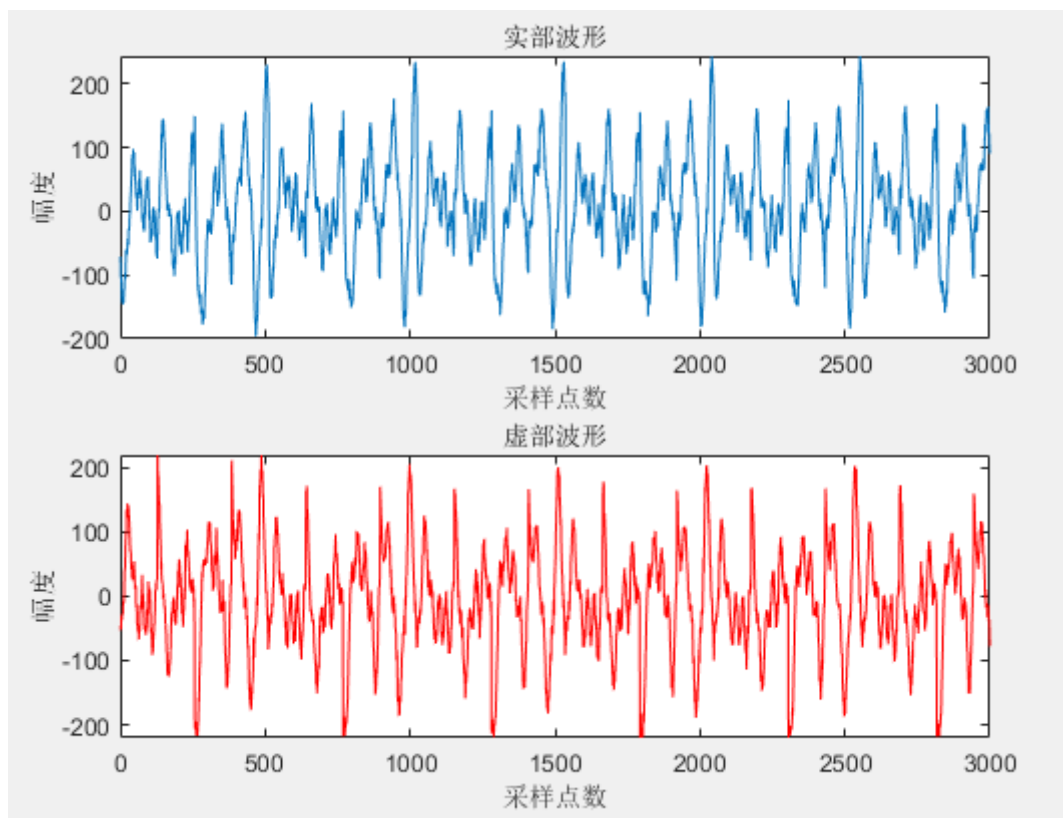
复信号个数为：128×128×2×4= 131072（个），即表示两个发射天线，四个接收天线，一共128个交替chirp脉冲，每个chirp脉冲一共128个采样点。

- ☐ 当采用虚拟天线模式时，复信号的个数为：128×128×8= 131072（个），表示一发8收，有8个接收通道的数据。
- ☒ 当没有采用虚拟天线模式时，复信号的个数为：~~2561284= 131072（个）~~，表示2发4收，只有4个接收通道，而256表示有256个chirp脉冲。

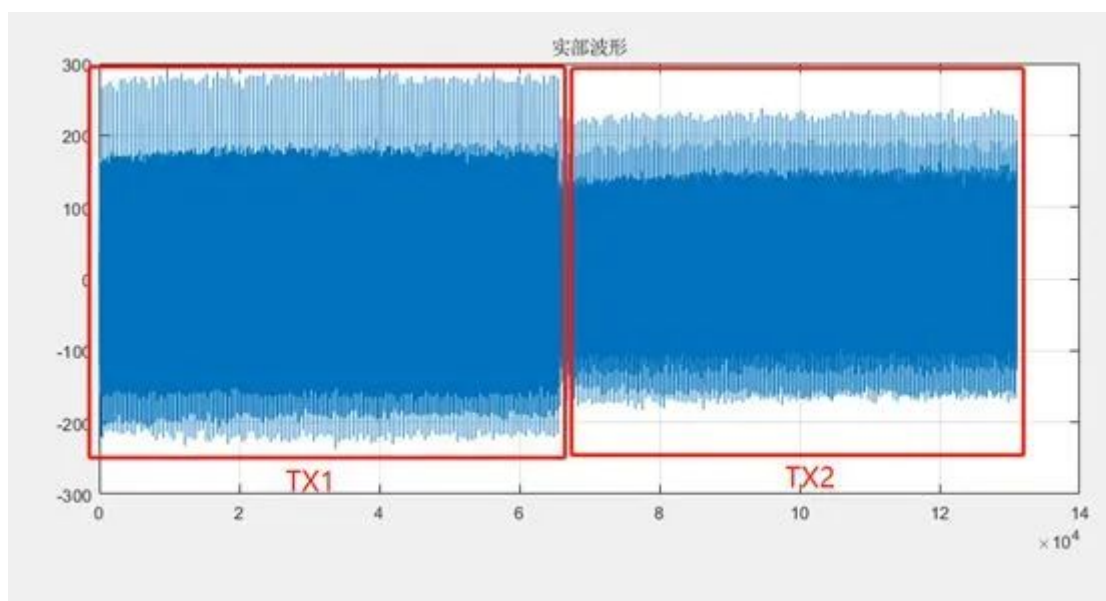
经过一系列解析，得到信号的实部和虚部时域图，如下：



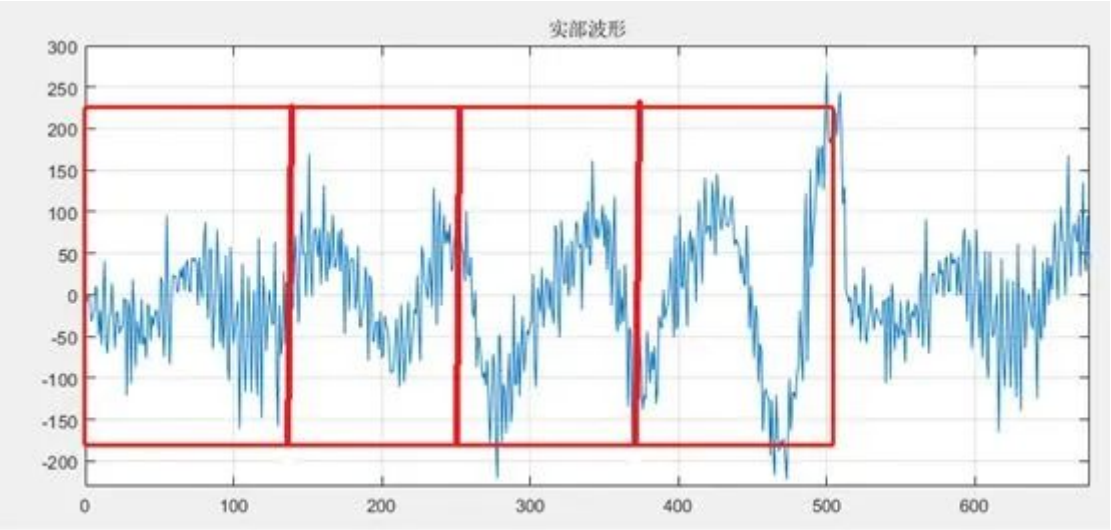
放大第一个chirp时域波形，正交性良好，如下图所示：



其中，可以看出，信号的前面一半是TX1（第一个发射天线）的128个chirp脉冲在四个接收通道上的数据，一共为 $128 \times 128 \times 4 = 65536$ 。同理，后半是TX2的数据。

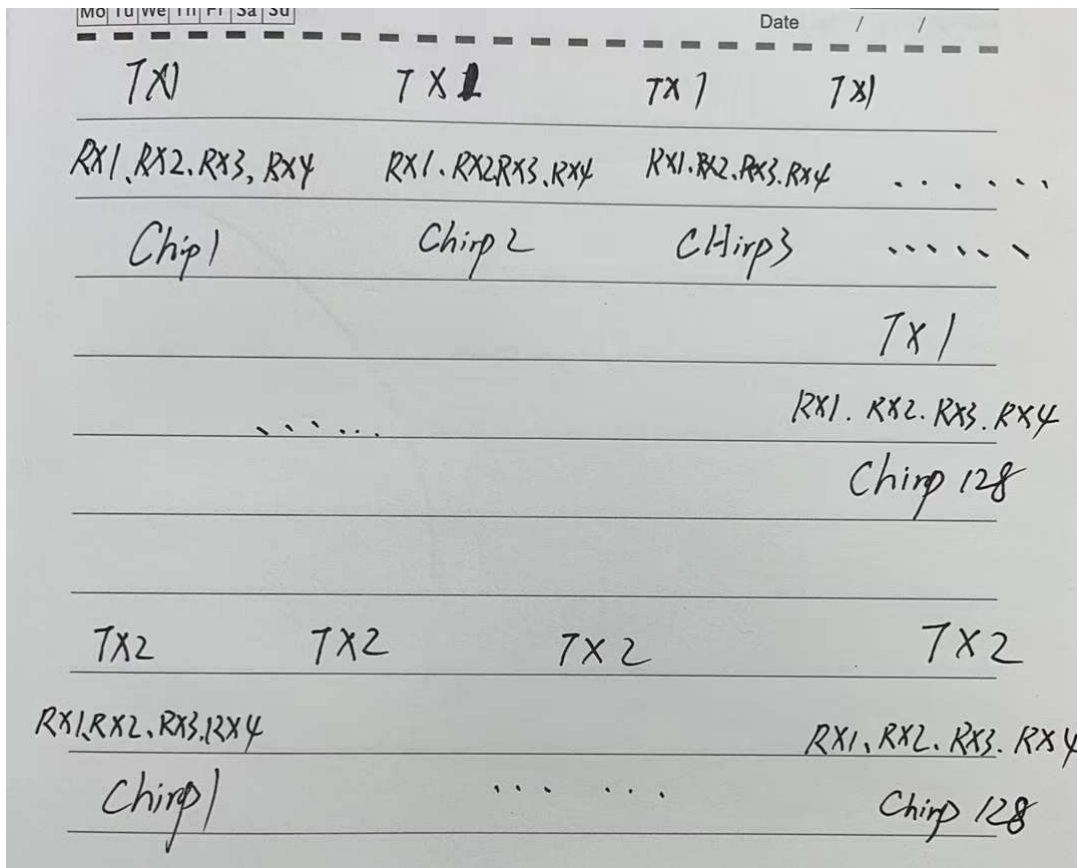


把上面的实部信号放大了看，只看前5个脉冲，红框的部分就是第一个发射天线（TX1）下，RX1、RX2、RX3、RX4的接收数据，每个RX一共128个点，如下图所示：



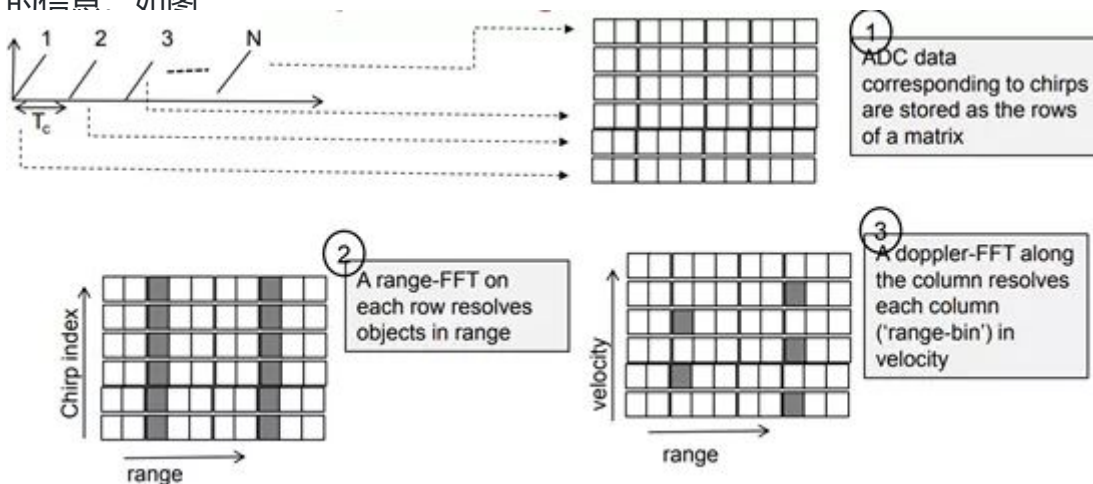
关于雷达串口输出的数据组成格式，可以参照下图进行理解

TX1				TX1				TX1				TX1				...
RX1	RX2	RX3	RX4	RX1	RX2	RX3	RX4	RX1	RX2	RX3	RX4	RX1	RX2	RX3	RX4	...
chirp1				Chirp2				Chirp3				Chirp4				...
.																
.																
.																
...	TX1				TX1				TX1				TX1			
...	RX1	RX2	RX3	RX4	RX1	RX2	RX3	RX4	RX1	RX2	RX3	RX4	RX1	RX2	RX3	RX4
...	Chirp29				Chirp30				Chirp31				Chirp32			
.																
.																
.																
TX3				TX3				TX3				TX3				...
RX1	RX2	RX3	RX4	RX1	RX2	RX3	RX4	RX1	RX2	RX3	RX4	RX1	RX2	RX3	RX4	...
chirp1				Chirp2				Chirp3				Chirp4				...
.																
.																
.																
...	TX3				TX3				TX3				TX3			
...	RX1	RX2	RX3	RX4	RX1	RX2	RX3	RX4	RX1	RX2	RX3	RX4	RX1	RX2	RX3	RX4
...	Chirp29				Chirp30				Chirp31				Chirp32			
.																
.																
.																
TX2				TX2				TX2				TX2				...
RX1	RX2	RX3	RX4	RX1	RX2	RX3	RX4	RX1	RX2	RX3	RX4	RX1	RX2	RX3	RX4	...
chirp1				Chirp2				Chirp3				Chirp4				...
.																
.																
.																
...	TX2				TX2				TX2				TX2			
...	RX1	RX2	RX3	RX4	RX1	RX2	RX3	RX4	RX1	RX2	RX3	RX4	RX1	RX2	RX3	RX4
...	Chirp29				Chirp30				Chirp31				Chirp32			



见《chirp、profile、frame、advframe与3FFT》

因此可以构成一个 $4 \times N$ (range bin) 的矩阵，这个就是2D数据包，包含距离和速度两个自由度的信息。如图



例子中，128个chirp在距离维做FFT之后，可以看到有两个距离门有目标，在速度维做FFT之后有5个速度门有目标。

右边这个radarCube其实就是一个二维矩阵，列数为range bin的总数（本工程中数值为128）。上图中重复出现的绿1-黄2-蓝3-粉4为4个接收天线分别接收到的数据，上图中显示了TX1和2全部chirp，所以行数为 $4 \times N$ 。那么Doppler Bin的总数怎么算呢，每次发射都有 2×4 个回波产生，我们要把这8个回波分开来算多普勒频率，所以Doppler Bin的总数为 $4 \times N / 8 = N / 2$ 。综上，距离维就是这个二维矩阵的行方向，多普勒维就是这个二维矩阵的列方向！