

Retina算法定位单粒子闪烁效应 应用研究

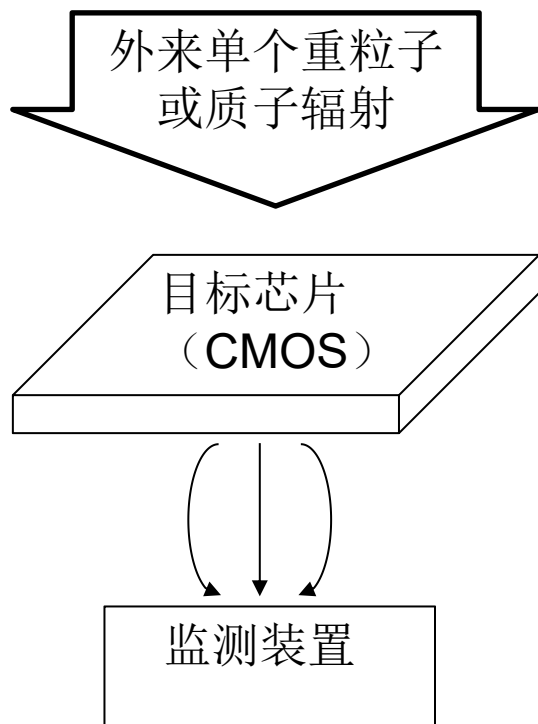
主讲人：邓文迪
单位：武汉纺织大学

2024.05.25

- 基于pixel芯片的单粒子闪烁效应定位策略
- Retina算法介绍
- 研究技术路线
- 研究进展

基于pixel芯片的单粒子闪烁效应 定位策略

芯片的单粒子闩锁效应



- 单粒子闩锁效应是辐射对电子系统产生的主要效应之一

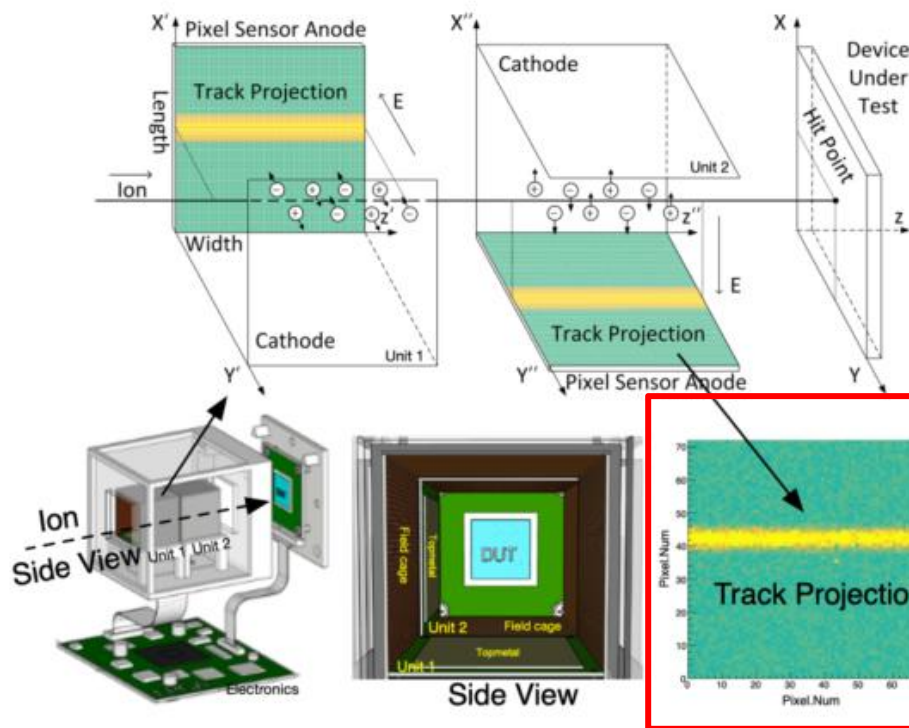
- 单个重离子或质子击中芯片内部MOS管会引起瞬时的逻辑翻转，导致芯片逻辑功能紊乱

- 人们通过在辐射过程中对目标芯片的工作状态（一般是功率）进行监测，从而判断该芯片

单粒子闩锁效应定位技术的意义：

找到目标芯片的辐射敏感区域，从而对针对这些区域做对应的抗辐射加固设计和处理。

基于pixel芯片的单粒子闪烁效应定位策略



轨迹寻迹算法设计目标:

- ✱ 二维平面空间中直线轨迹在线寻迹
- ✱ 更好的轨迹寻迹效率和位置分辨率
- ✱ 更快的实时轨迹寻迹速率 5-8us/track

TABLE 2. Table of various conditions for the formation of a single track on the pixel detector.

Parameter	Value	Parameter	Value
Particle types	Kr^{26+}	Height of incidence (mm)	1.5~5
Particle energy (MeV/u)	25	Electric field (V/cm)	300
Gas	Ar : CO ₂ (70:30)	Drift speed (cm/ μ s)	0.7
Gas density (mg/cm ³)	1.714	Diffusion coefficient (μ m/ \sqrt{cm})	150
Ionized energy W_I (eV)	28.4	Pixel pitch (μ m)	83
Energy loss (keV/ μ m)	3.06	pixel array	72 × 72
Range of ions in gas (mm)	513		

Retina算法介绍

Retina 算法

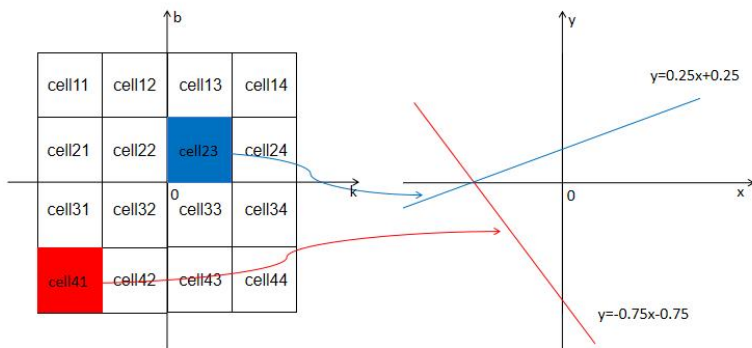
-Retina算法是一种模式识别算法。

-算法通过提取目标模式的关键参数建立对应的参数空间，并根据需求将其离散化。

-输入待识别样本，扫描离散化参数空间，找出吻合度（权重值）高的参数空间位从而识别出样本中可能存在的目标模式。

算法应用示例-二维平面直线轨迹重建：

步骤1:参数空间定义与离散化参数空间建立

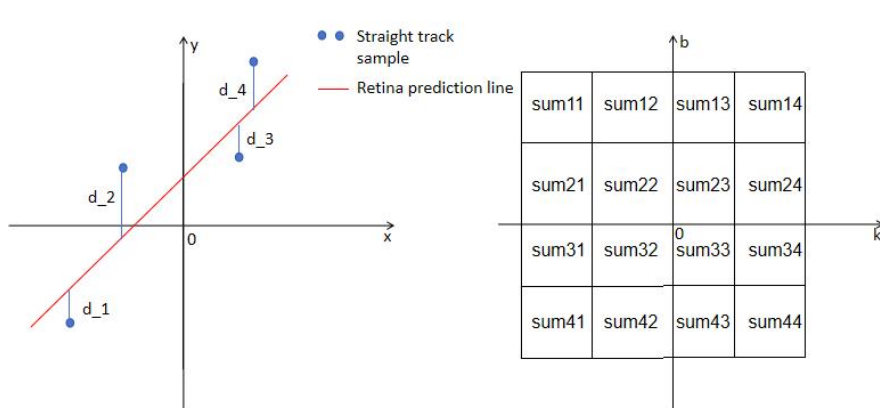


关键参数设置：

✳ 参数空间范围

✳ 参数空间粒度

步骤2:输入样本扫描输出样本在离散化空间中权重和Sum分布结果



$$Sum_{ij} = \sum_{n=1}^m W_{ij,n} = \sum_{n=1}^m e^{-\frac{d_{ij,n}^2}{2\sigma^2}}$$

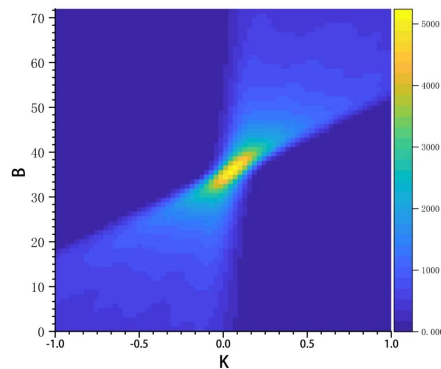
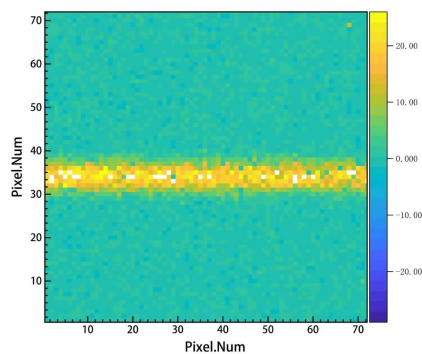
加入阈值后的Sum计算：

$$Sum_{ij} = \sum_{n=1}^m W_{ij,n}$$

$$= \sum_{n=1}^m A_n * \begin{cases} e^{-\frac{d_{ij,n}^2}{2\sigma^2}} & d_{ij,n} \leq d_{th} \wedge A_n \geq A_{th} \\ 0 & d_{ij,n} > d_{th} \vee A_n < A_{th} \end{cases}$$

Retina 算法

步骤3:找出Sum分布中的最大值/极大值, 作为识别到的粒子径迹输出

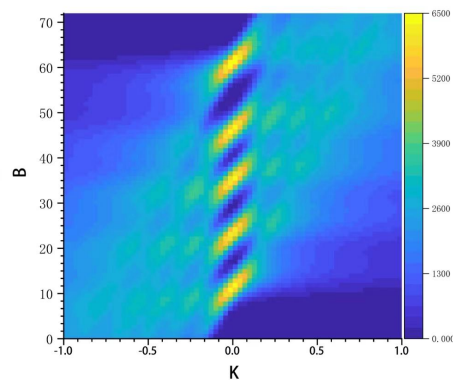
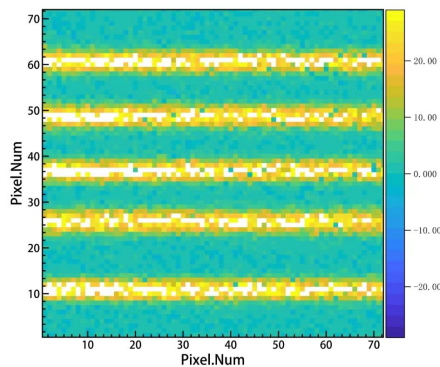


✱ 单轨迹样本寻找最大值

✱ 多轨迹样本寻找极大值

✱ 算法优势:

- 1.Retina cell扫描可全并行计算
- 2.Retina cell单元可重复配置调用



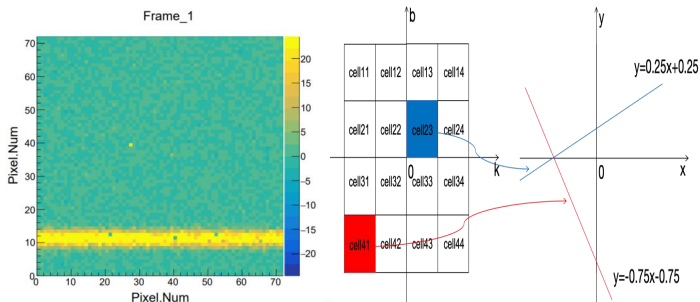
研究技术路线

单轨迹样本事件的轨迹识别:

Retina 算法移植

借鉴之前本人CMS实验中Retina算法

- 样本事件相比较简单
- 模拟Topmetal建模样本+ Single-Kr样本
- 单个样本中仅有单个粒子轨迹



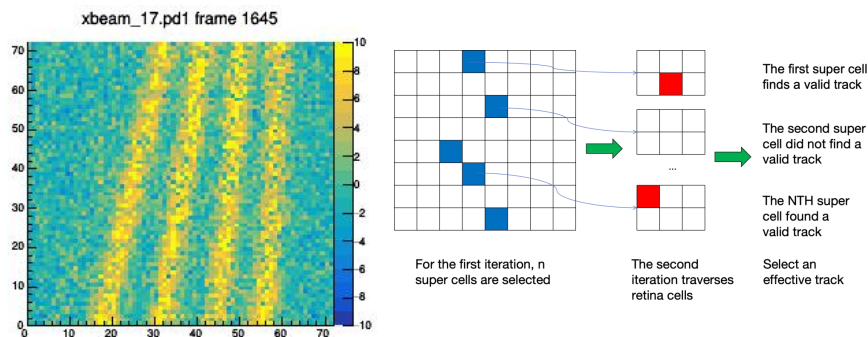
研究目的:

- 探索Retina算法识别重离子轨迹的可行性
- 算法识别轨迹的准确度
- 评估算法开销
- 算法重建参数的精度

多轨迹样本事件的轨迹识别:

迭代视网膜算法 (Iterative Retina, IR算法)

- 样本事件相比较复杂
- 单个样本中包含多个粒子轨迹



研究目的:

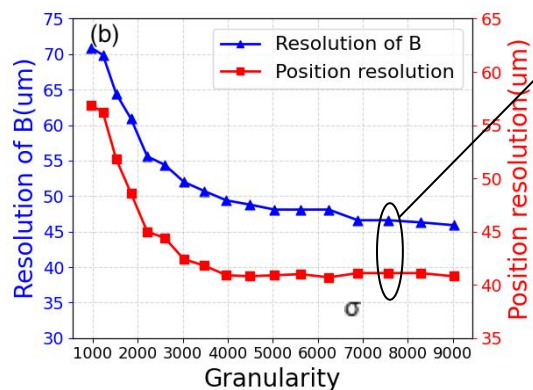
- 设计具体的算法满足SEE要求的在线重离子寻迹算法
- 算法并行性
- 算法轨迹识别效率
- 节省算法计算量易于固件实现
- 算法重建参数的精度

研究进展

✳️在PC上完成Retina算法设计和仿真测试（持续更新）

-Retina 与 IR算法的仿真测试均已完成（样本统计量约为5000左右）

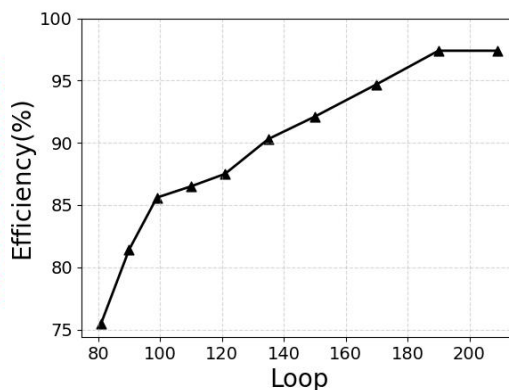
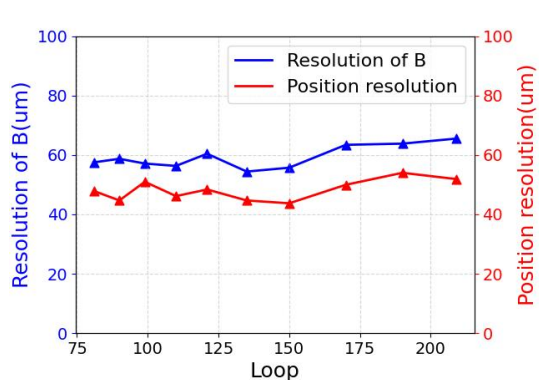
●单粒子轨迹事件中Retina算法测试结果：



granularity	σ	A_th	sample size	d_th	σ_k	σ_b	position resolution
76*100	0.8	10pixel	5000	0.4pixel	0.573 ²	46.9um	39.4um

- 算法的位置分辨率随着算法粒度增大而逐步提高
- 算法粒度高于3000后位置分辨率结果稳定在40um左右，约为单个pixel尺寸（83um*83um）的一半。

●多粒子轨迹事件中IR算法测试结果：

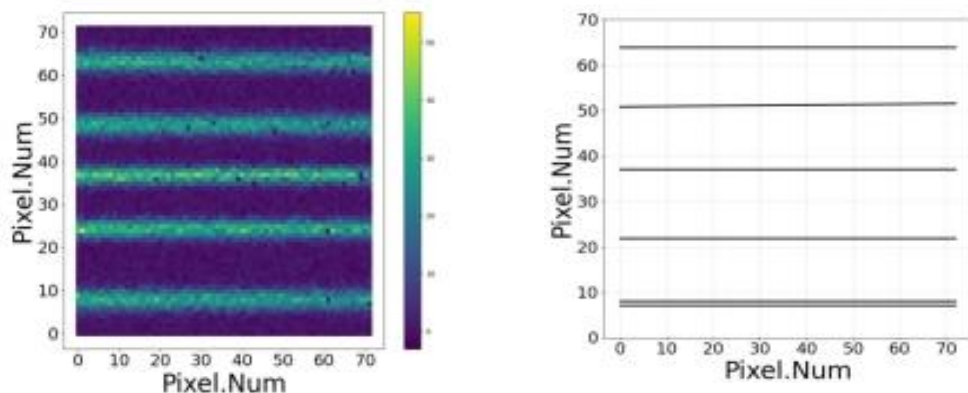


- 算法的位置分辨率在45um左右，精度与单轨迹事件比有所下降。
- IR算法的算法效率在粒度Loop=140以上时达到了百分之90以上。（Loop为IR算法每次扫描所用的粒度）

研究结论与结果汇总：

- Retina算法可以实现重离子轨迹寻迹和参数重建工作
- pixel 探测器+Retina算法的系统有实现SEE定位的可能性

●多粒子轨迹事件中IR算法重复识别问题:



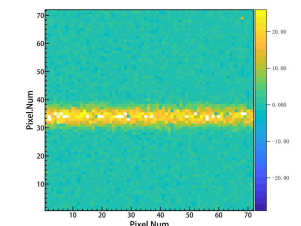
-重复识别的轨迹占比
8%左右

TABLE 4. The performance results of IR algorithm under the granularity of loop = 19×10 .

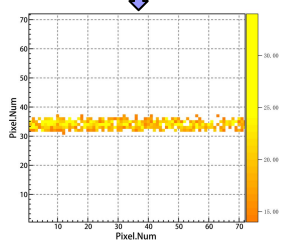
Parameter	Value	Parameter	Value
$Z \times Z$	19×19	N_{all}	30000
$Q \times Q$	10×10	N_{true}	26806
σ	9.5	N_{repeat}	2415
σ_I	0.4	N_{miss}	779
Pixel threshold	10 pixel	Efficiency	97.4%
d_threshold	3.4 pixel	Variance k	0.009 rad
d_threshold_I	0.25 pixel	Variance b	$63 \mu m$
Position resolution	$53 \mu m$		

✳在FPGA上完成Retina固件设计（正在进行中）

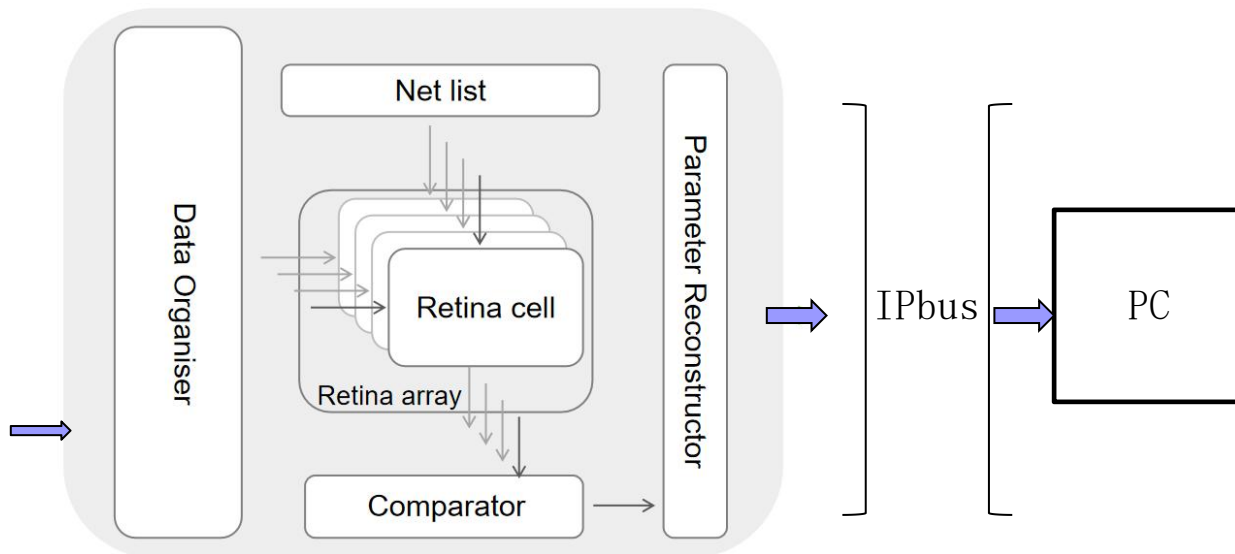
1.针对重粒子轨迹样本的Retina算法原型固件设计:



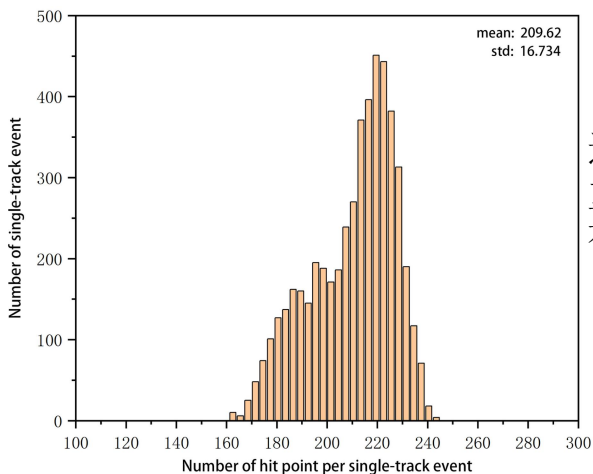
(Color line) The pixel distribution graph of single-track event sample.



The heavy ion trajectory samples after background noise filtering.



背景噪声过滤



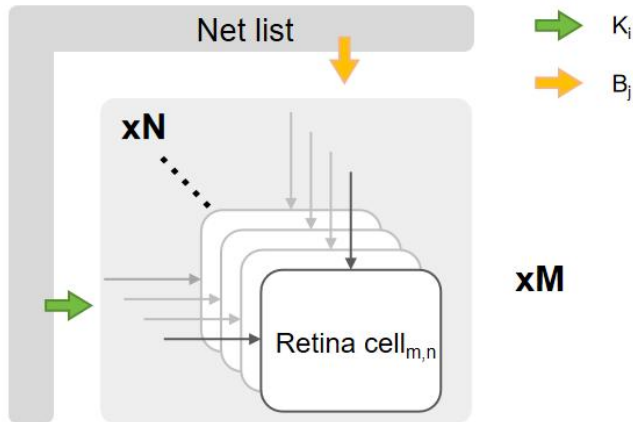
过滤后平均每条重离子径迹包含有效像素点数量为210

固件设计关键特征与目标:

- 保持参数与寻迹精度与仿真结果一致或相近
- 参数可配置的弹性计算单元，保持高移植性
- 全定点计算，尽可能提升FPGA资源利用效率
- 全流水线计算处理，提高固件计算速度

✳ Retina array 模块模块设计:

● Retina array 模块由 网表+N*M个retina cell 组成

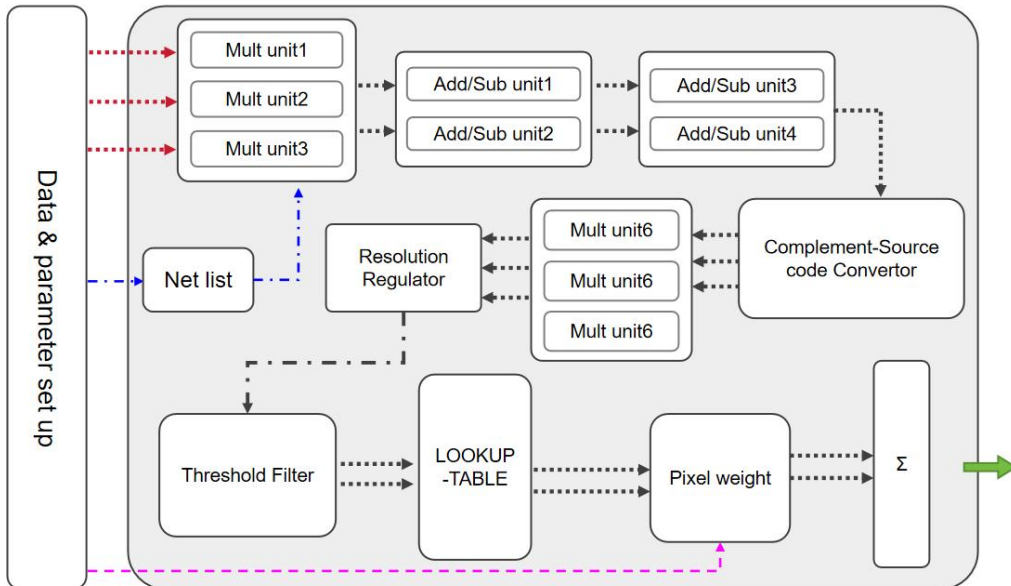


● 网表模块通过输入算法粒度 (M, N) 输出用于扫描计算的对应M*N个参数队列 (k_i, b_j)

$$k_i = -1 + \frac{2}{M}(i - \frac{1}{2}) \quad i = 1, 2, 3, \dots, M$$

$$b_j = \frac{72}{N}(j - \frac{1}{2}) \quad j = 1, 2, 3, \dots, N$$

单个retina cell 模块结构: 主要利用分步骤流水线的结构完成对应参数空间中权重值Sum的计算



Calculation step	Use of calculation unit	Latency(CLKs)
d _{ij} v	mult x3 add/sub x4	16(320ns)
d _{ij} ² v	mult x3	7(140ns)
param	mult x3	7(140ns)
exp	lut table x2	5(100ns)
pixel	mult x2	5(100ns)
sum	add x1	1(20ns)
TOTAL	addx3 subx2 lutx2 multx11	41(0.82us)

✳️ Retina 固件设计在线性能评估:

1. 算法固件资源消耗

单个 Retina cell

Retina Cell Number	1
Luts	1.22(0.60)
Logic Registers	1.73(0.43)
Block Memory	4(0.90)
DSP48	5(0.60)

100个 Retina cell 规模组成的完整固件

	Luts k (%)	Logic Registers k (%)	Block Memory # (%)	DSP48 # (%)
Net list	27.86(13.67)	0(0.00)	0(0.00)	200(23.80)
Retina array	122.72(60.21)	167.60(41.11)	400(89.88)	500(59.52)
Comparator	1.46(0.71)	2.85(0.70)	0(0.00)	0(0.00)
data originator & etc	0.46(0.22)	0.09(0.02)	171.5(38.54)	0(0.00)
TOTAL w/o Data originator	152.11(74.63)	170.45(41.81)	400(89.88)	700(83.32)

-整个 Retina 固件模块在 Xilinx 7 系列 KC325T FPGA 上消耗大约 80% 的逻辑和计算资源。

2. 算法固件实时性能初测试:

Step	Step latency (clock cycles)	Step latency (us)
(1) Distance caculation	16	0.32
(2) Weight caculation	24	0.48
(3) Sum _{ij} caculation	1	0.02
(4) Compare	8	0.16
Total	49	0.98

-算法处理从单个有效像素点输入至计算出权重值所消耗的时间为 49clks@50MHz -> 0.98us

-算法处理单粒子径迹样本的平均时间为 (49+N) clks@50MHz -> 5.08us

✳未来计划:

1.继续利用Retina算法进行寻迹方面研究

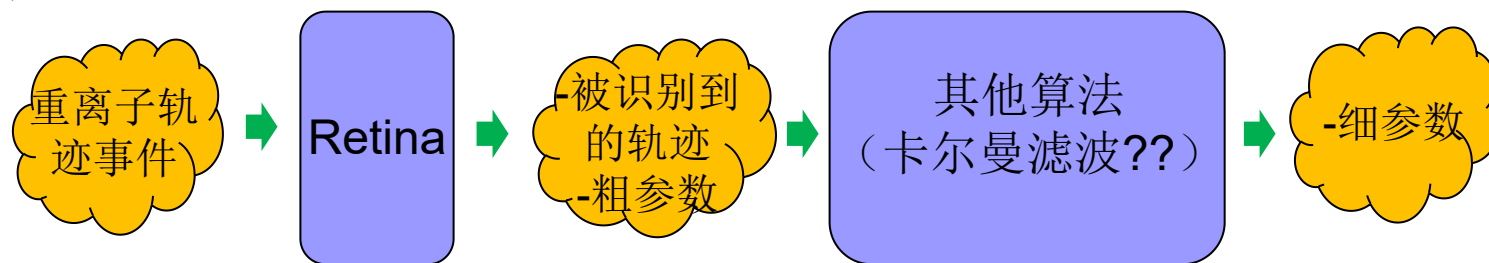
●方法研究（仿真）

- 其他粒子轨迹模式识别
- 计算步骤简化优化
- 二维到三维的协同定位
-

●FPGA固件完善于实装&系统搭建

- 传输协议开发
- IR算法固件逻辑开发
- 提高全局时钟速率50Mhz→200Mhz
-

2.其他算法融合



- 主要作为补充解决目前算法在参数重建精度上的问题。
- 不需要承担寻迹任务，只需对找到的轨迹做进一步参数重建，提高整体位置分辨率。
- 在同样重建分辨率情况下降低Retina粒度，降低算法资源消耗?? ?（愿望）

谢谢