



中国科学院重庆绿色智能技术研究所

Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences



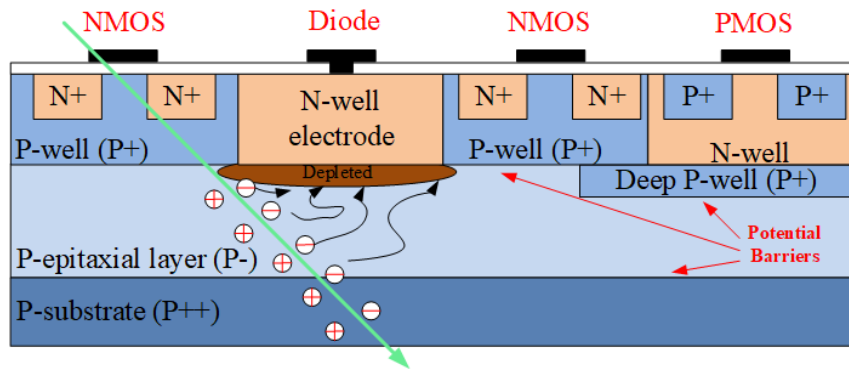
A 2-D clustering algorithm for data reconstruction in vertex detector of ILC



R. ZHAO; A. BESSON; C. HU-GUO; Y. HU

全国辐射探测微电子学交流 (NME'2021)

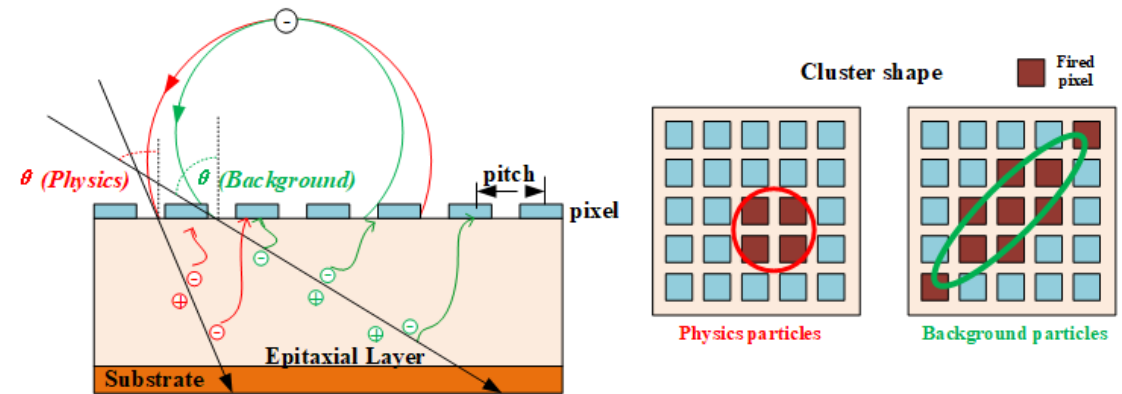
研究背景



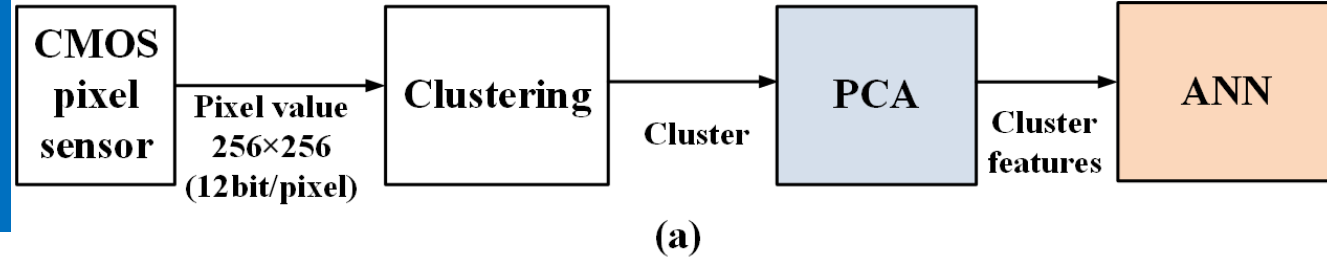
CMOS像素探测器将传感器和读出电路集成在同一块衬底，在大型高能物理实验设备中已经得到了应用，包括STAR实验和ALICE实验

ILC中存在大量来自背景的命中，由于这些带电粒子具有较小动量（ $10\sim 100\text{MeV}/c$ ），在探测器表面形成具有细长形状的簇（elongated cluster）。

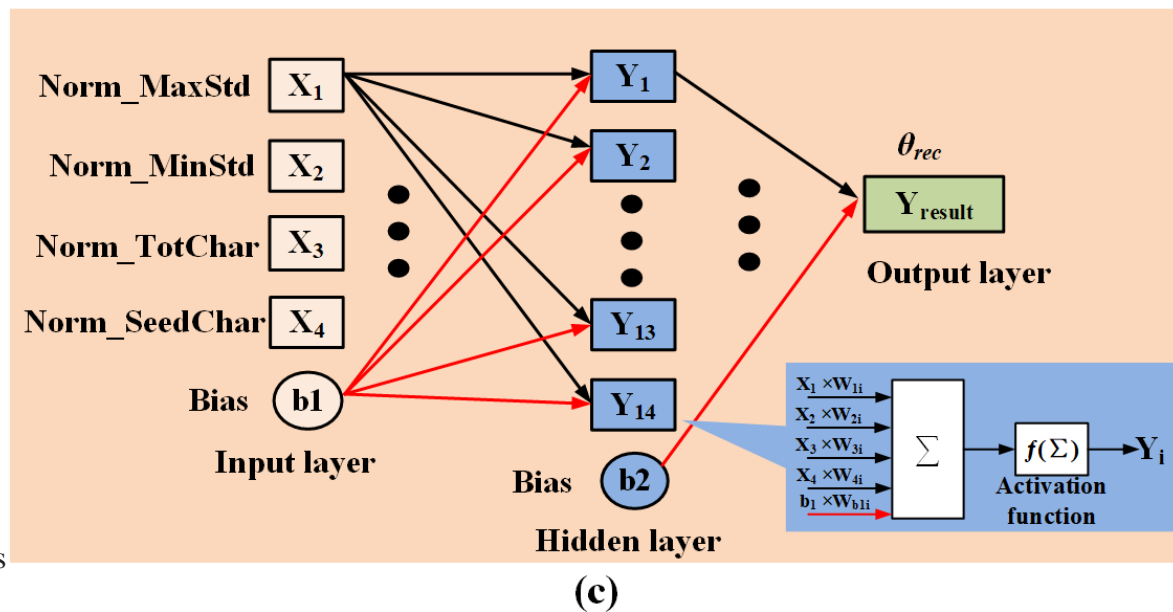
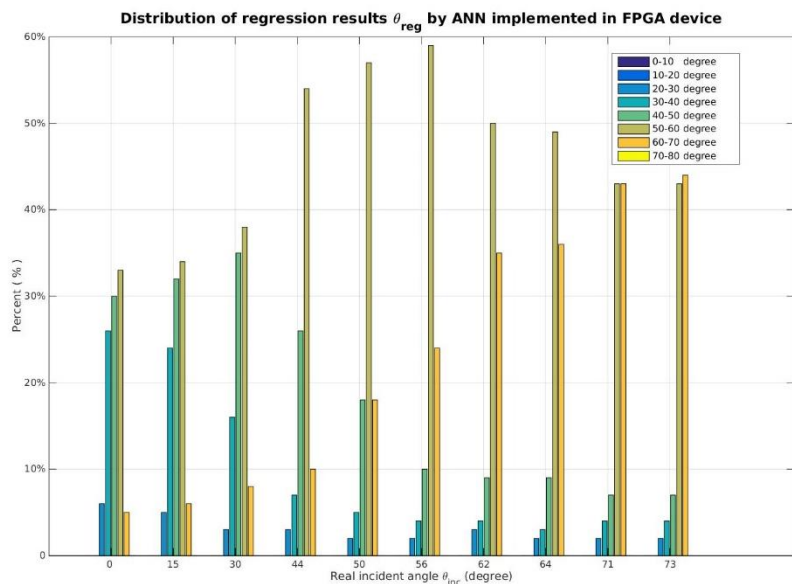
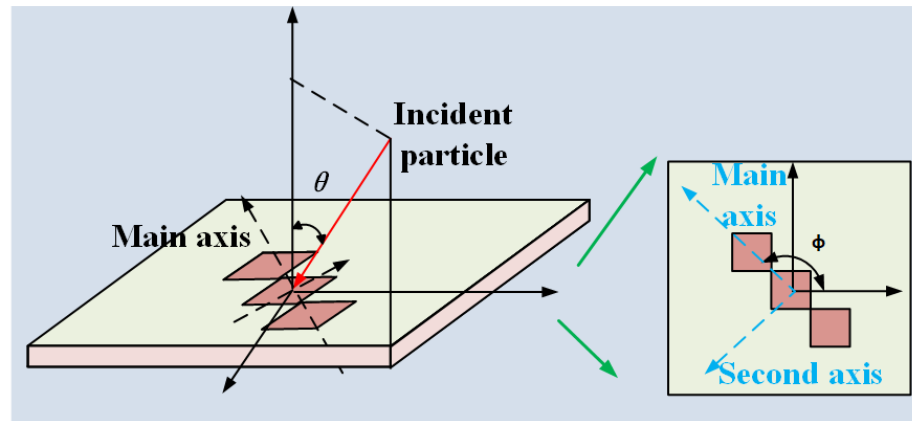
工艺的快速进步使得集成更多功能的CMOS像素探测器成为可能。我们提出集成片上人工神经网络的CPS。



研究背景

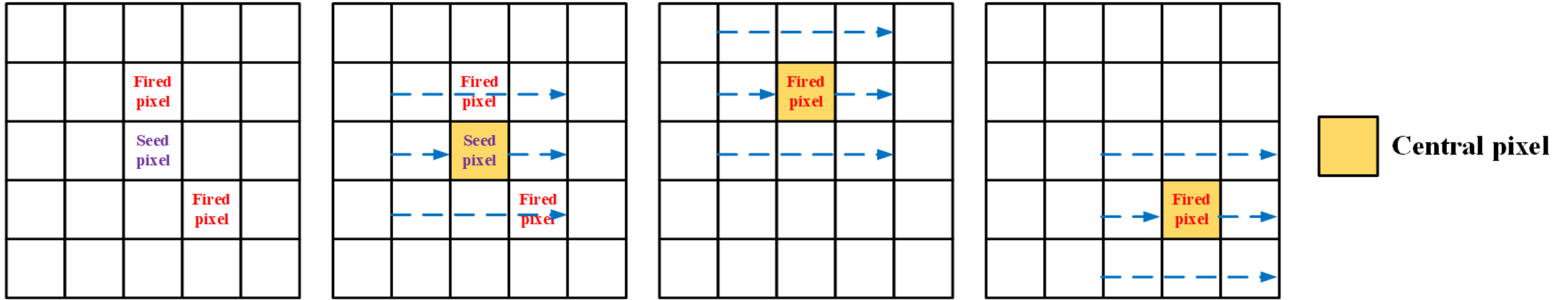


前期的可行性验证主要在FPGA中完成，包括簇找寻 (clustering) 模块、主成份分析 (PCA) 模块、人工神经网络 (ANN) 模块。FPGA实现的测试结果与TMVA软件重建结果一致，基本符合入射角度的变化规律。



"FPGA Implementation of an Artificial Neural Network for Subatomic Physics Experiment Particles Recognition." *TWEP2018*

传统簇找寻算法



FPGA中实现的簇找寻模块（clustering）基于传统的相邻像素的搜索算法。

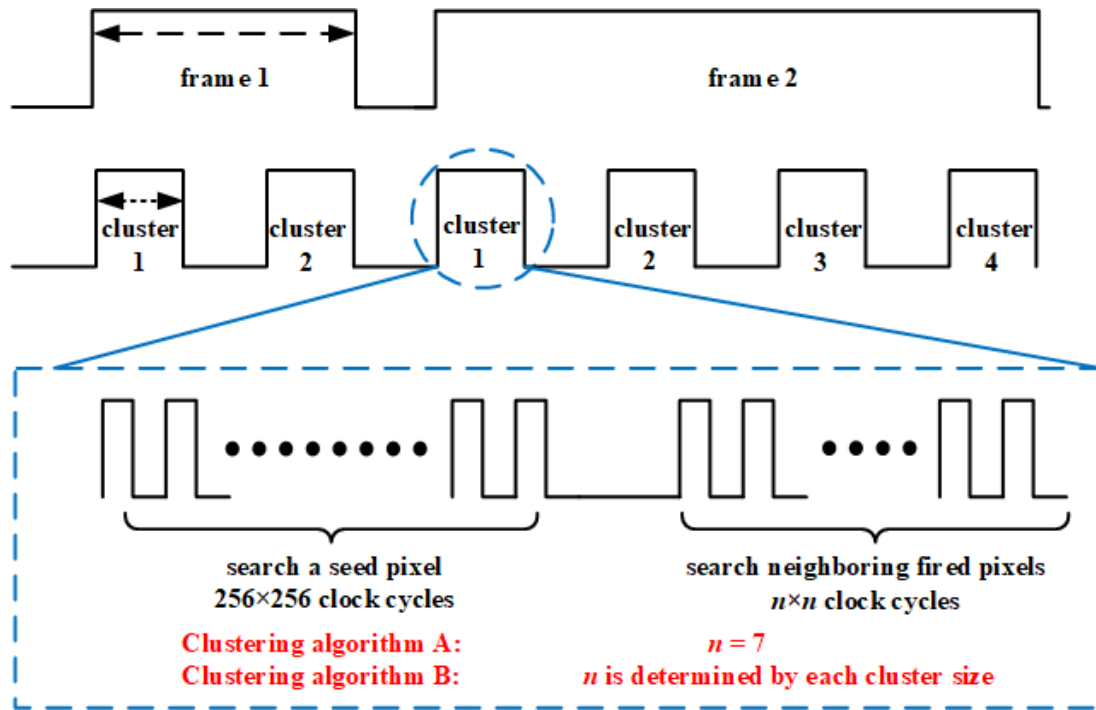
clustering algorithm A:

1.找寻种子像素； 2.搜索种子像素的8个相邻像素中的激活像素； 3.以相邻像素为中心，重复搜索其8个相邻像素，直到其相邻像素中无激活像素。

clustering algorithm B:

FPGA中实现的找寻算法额外增加了 7×7 的簇尺寸限制（根据数据采集的最大入射角度确定）

传统簇找寻算法时序



传统算法在时序中存在不确定

clustering algorithm A:

1. 找寻每个cluster时间**不确定**（取决于cluster尺寸）
2. 找寻一帧中所有的cluster时间**不确定**（取决于cluster数量）

clustering algorithm B:

1. 找寻每个cluster时间**基本一致**（7×7个时钟周期）
2. 找寻一帧中所有的cluster时间**不确定**（取决于cluster数量）

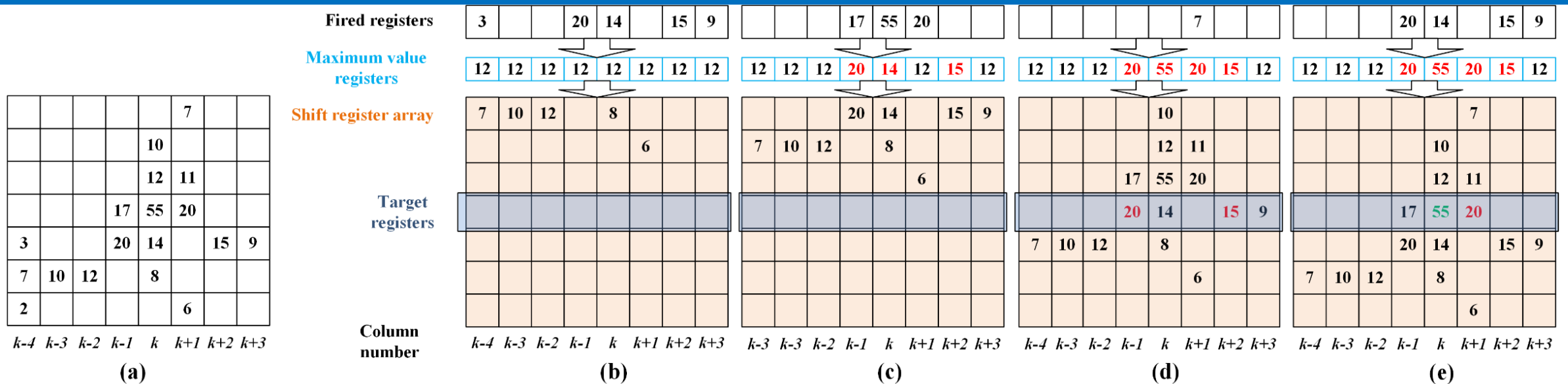
传统簇找寻算法面临挑战

1. 资源需求较大：存储整个一帧原始数据完成簇的找寻
2. 时序复杂，不利于实时处理。
3. 特殊簇处理存在问题：包括较大簇以及重叠簇

团队提出一种能够片上集成的面向列级ADC的实时簇找寻算法

("A 2-D Clustering Algorithm for Data Reconstruction in Vertex Detector of ILC," doi:10.1109/TNS.2021.3115146.) 。

2-D clustering algorithm



(a) 所示为一个 7×8 像素阵列的簇 (cluster) ；

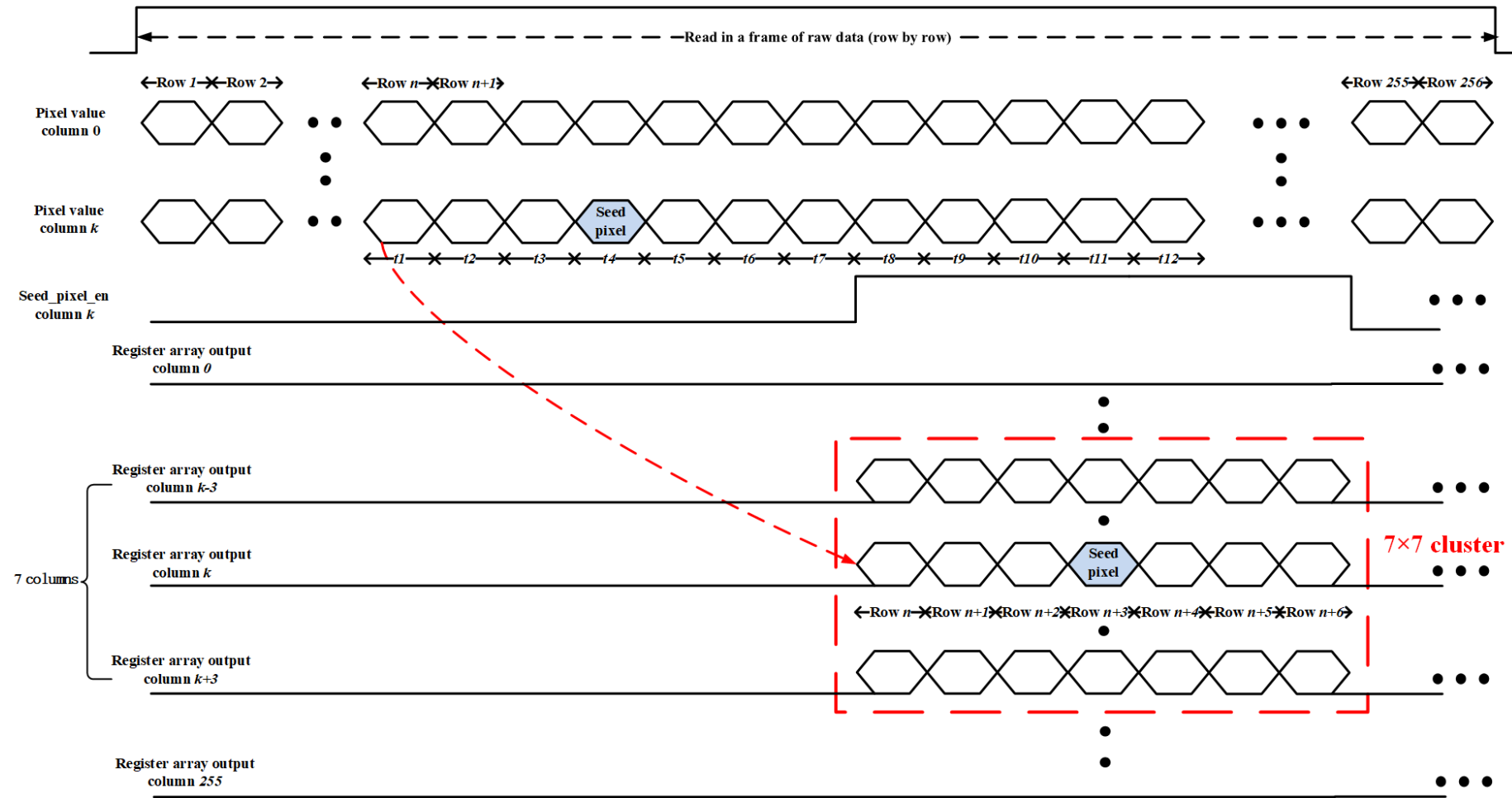
(b) 在fired registers中选择激活像素；

(c) $k-1$, k , 和 $k+2$ 列的最大值寄存器被更新；

(d) $k-1$ 和 $k+2$ 列目标寄存器分别捕获到列种子像素 (在一列7个像素中具有最大值)

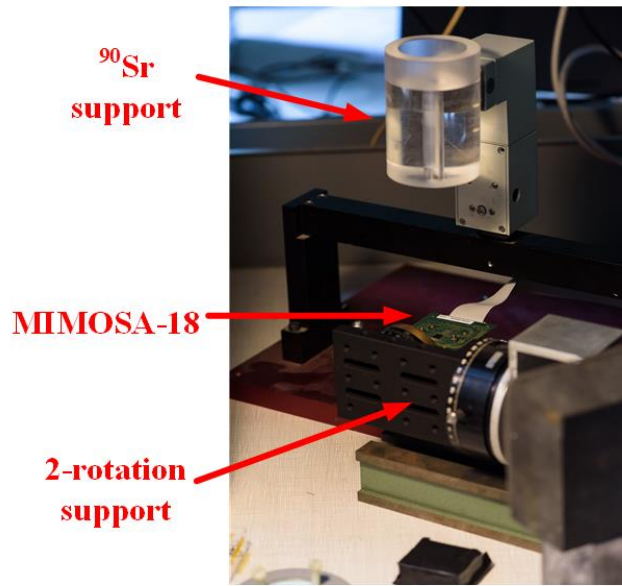
(e) k 列目标寄存器捕获到一个种子像素, $k+1$ 列目标寄存器捕获到一个列种子像素. 从 $k-3$ 到 $k+3$ 列的移位寄存器中的激活像素组成了一个簇。

提出算法的时序

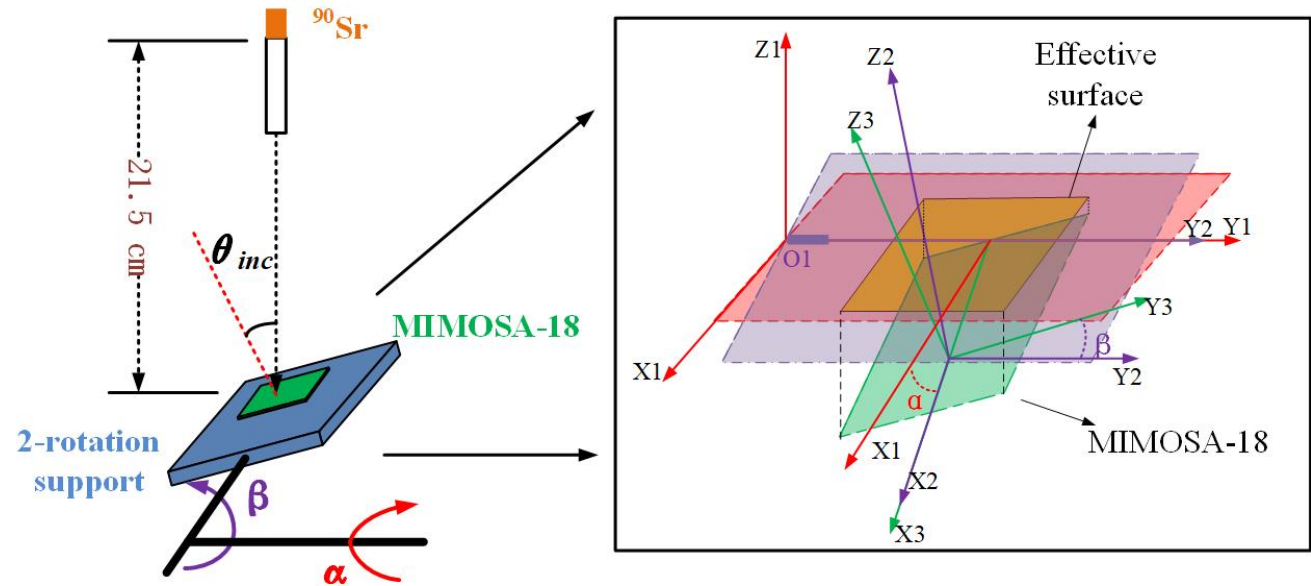


- (a) 像素值逐行读入处理
(基于列级ADC)
- (b) 种子像素从读入到识别
读出需要7个延时 (取决于
窗口尺寸)
- (c) 位于第k列 第n+3行的
种子像素被检测并读出。
移位寄存器从第k-3到 k+3
列的像素组成一个7x7簇。

数据采集系统



(a)



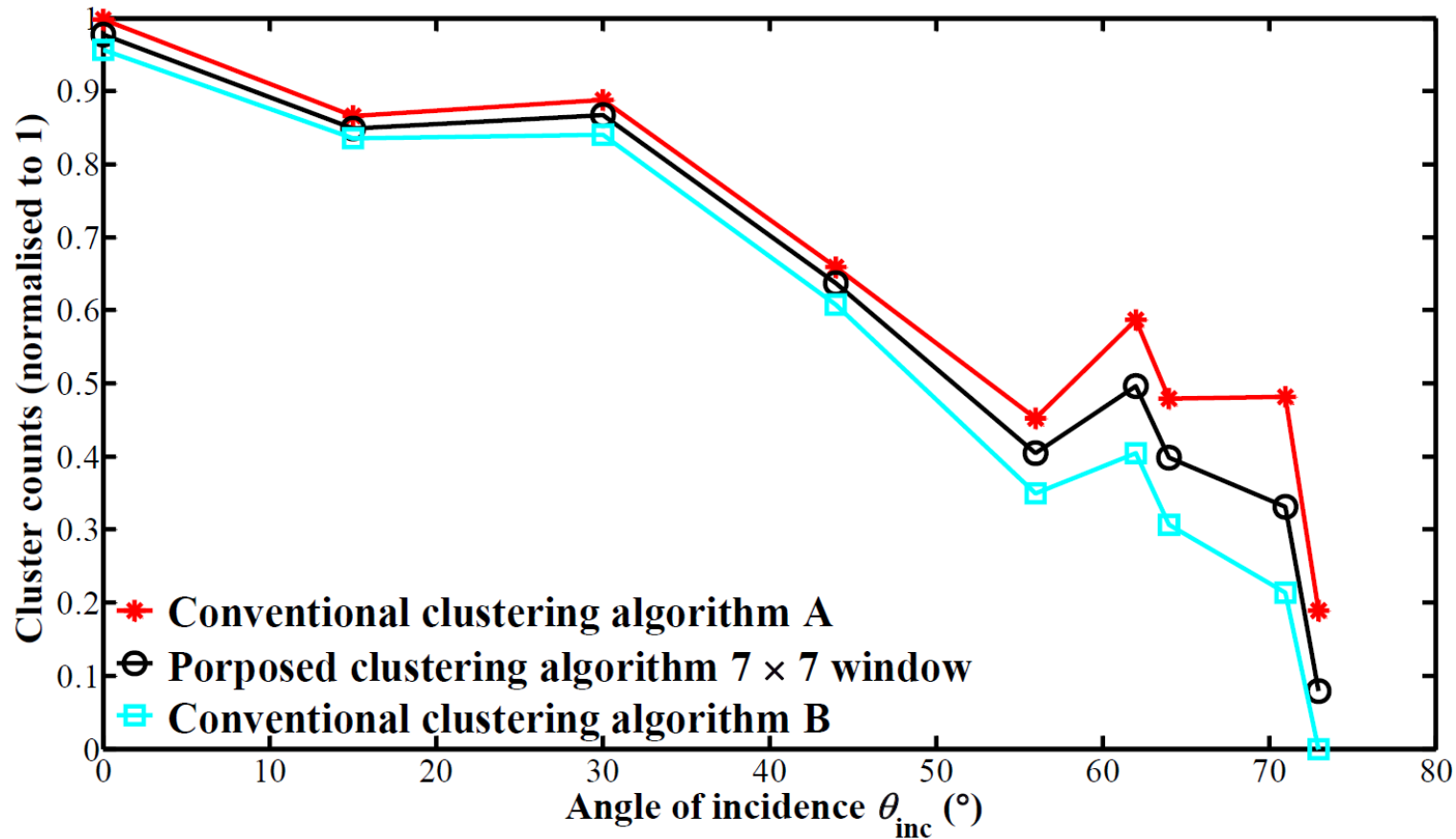
(b)

(a) 采集不同入射角度下的簇信息，共计九个入射角度 ($\theta_{inc} = 0^\circ, 15^\circ, 30^\circ, 44^\circ, 56^\circ, 62^\circ, 64^\circ, 71^\circ$, 和 73°)

(b) MIMOSA-18专门为高能物理实验设计的CMOS像素探测器。采用AMS 0.35 mm OPTO 工艺制造，芯片厚度 $50\mu\text{m}$ ，其中外延层 $14\mu\text{m}$ ，电阻率 $10\text{-}15\ \Omega\cdot\text{cm}$ 。

(c) 分辨率为 256×256 像素， $\text{pitch}=10\ \mu\text{m}$., 经过相关双采样后数据为12bit。

算法计数比较



包括我们提出的算法，共计三种算法仿真结果。

(a) 随着入射角度的增加，实际的有效芯片面积（感应面积）下降。三种算法的簇计数变化符合有效面积的减小趋势。

(b) 提出的算法可以达到相同水平的簇计数。

(c) 随着入射角度变大，出现更多细长的簇，可能被算法A识别为两个簇。不同算法的簇计数差距变大。

特殊簇的处理

| | | | | | | | | |
|----|----|----|----|-----|-----|----|----|----|
| | | | | | | | | |
| 10 | 9 | 15 | 14 | 14 | 10 | 7 | | |
| 31 | 36 | 43 | 45 | 80 | 71 | 25 | 7 | |
| 20 | 20 | 32 | 85 | 246 | 239 | 77 | 34 | 20 |
| | 9 | 9 | 29 | 85 | 92 | 69 | 67 | 56 |
| | | | | 11 | 18 | 13 | 17 | 22 |
| | | | | | | | | 7 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

| | | | | | | | | | | | | | |
|---|----|----|----|----|----|----|----|----|----|-----|----|----|----|
| | | | 6 | | | | | | | | | | |
| | | | | 7 | 10 | 6 | 8 | 6 | 11 | 27 | 17 | | |
| | 58 | 67 | 66 | 48 | 59 | 57 | 73 | 72 | 50 | 127 | 42 | 10 | |
| | 13 | 39 | 70 | 43 | 39 | 44 | 43 | 47 | 75 | 130 | 39 | 6 | |
| | | 7 | 8 | | 8 | | 8 | 8 | 7 | | | | |
| | | | | | | | | | | | | | |
| | | | | | | | | | | | | | |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |

出现在入射角度=71° 时的具有大尺寸的簇:

clustering algorithm A: 三个种子像素, 分别位于第4列 (246 ADU), 第0列 (31 ADU), 和第 8列 (56 ADU);

clustering algorithm B: 位于第4列的1个种子像素;

2-D clustering algorithm: 位于第4列的1个种子像素。

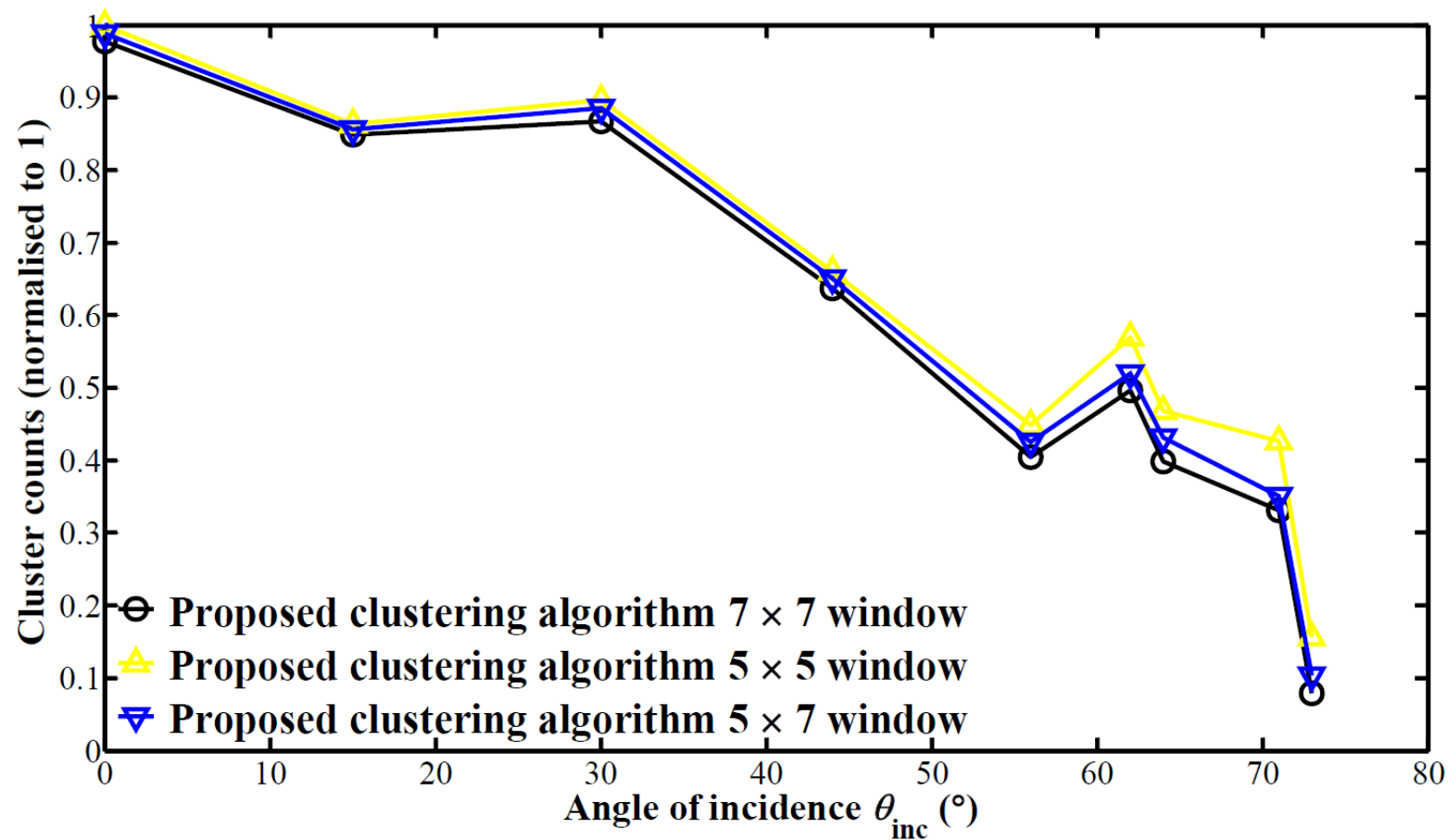
出现在入射角度=71° 时的具有重叠部分的簇:

clustering algorithm A: 两个种子像素, 位于第10列的种子像素 (130 ADU) 以及位于第 3列 (70 ADU);

clustering algorithm B: 位于第10列的1个种子像素;

2-D clustering algorithm: 两个种子像素, 位于第10列 (130 ADU) 以及位于第 3列 (70 ADU)。

不同窗口比较

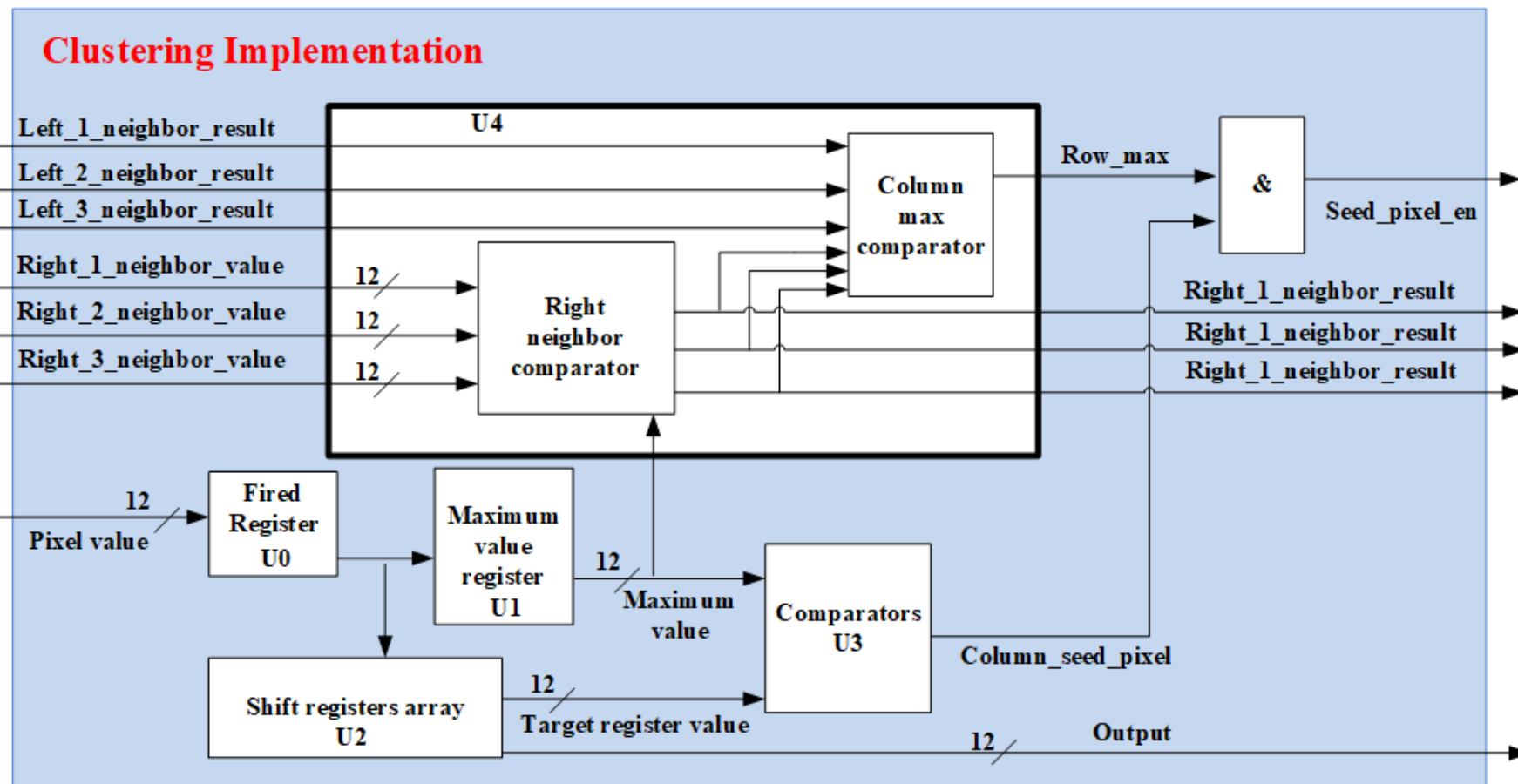


算法中窗口尺寸取决于CMOS像素探测器芯片的外延层参数。

采用具有更高外延层电阻率的MIMOSA-26芯片 ($\sim 400 \Omega \cdot \text{cm}$) 或者更薄外延层厚度的芯片, 窗口尺寸可以减小。

如图所示, 采用 5×5 窗口可能将一个簇识别为两个簇, 簇计数有一定的增加, 同时, 更小的窗口尺寸意味着资源更少, 执行时间更短。

提出算法的单元结构

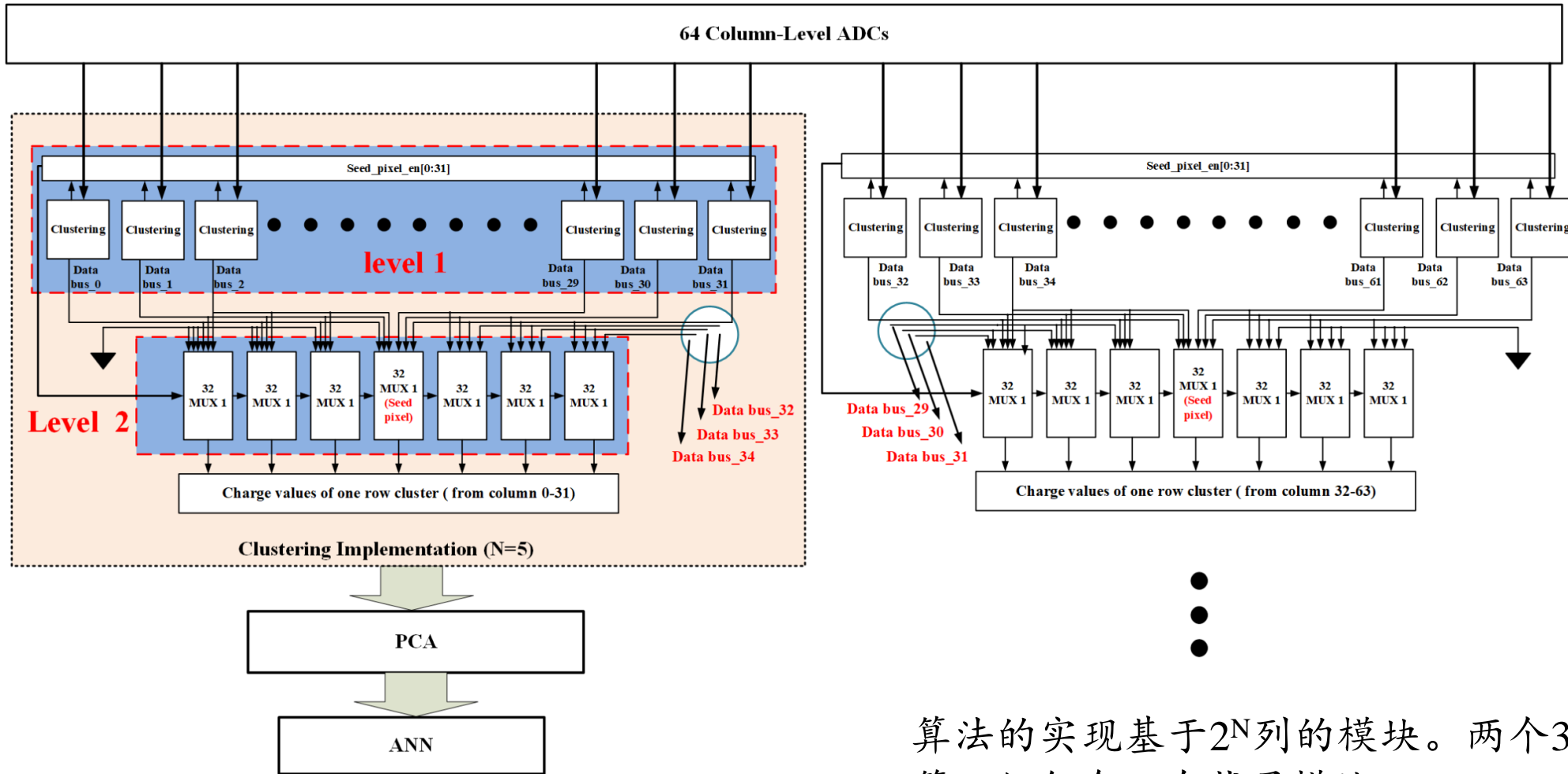


seed_pixel_en 信号激活表示捕获一个种子像素，是row_max和column_seed_pixel 与操作的结果。

column_seed_pixel信号激活如果目标寄存器等于最大值寄存器。

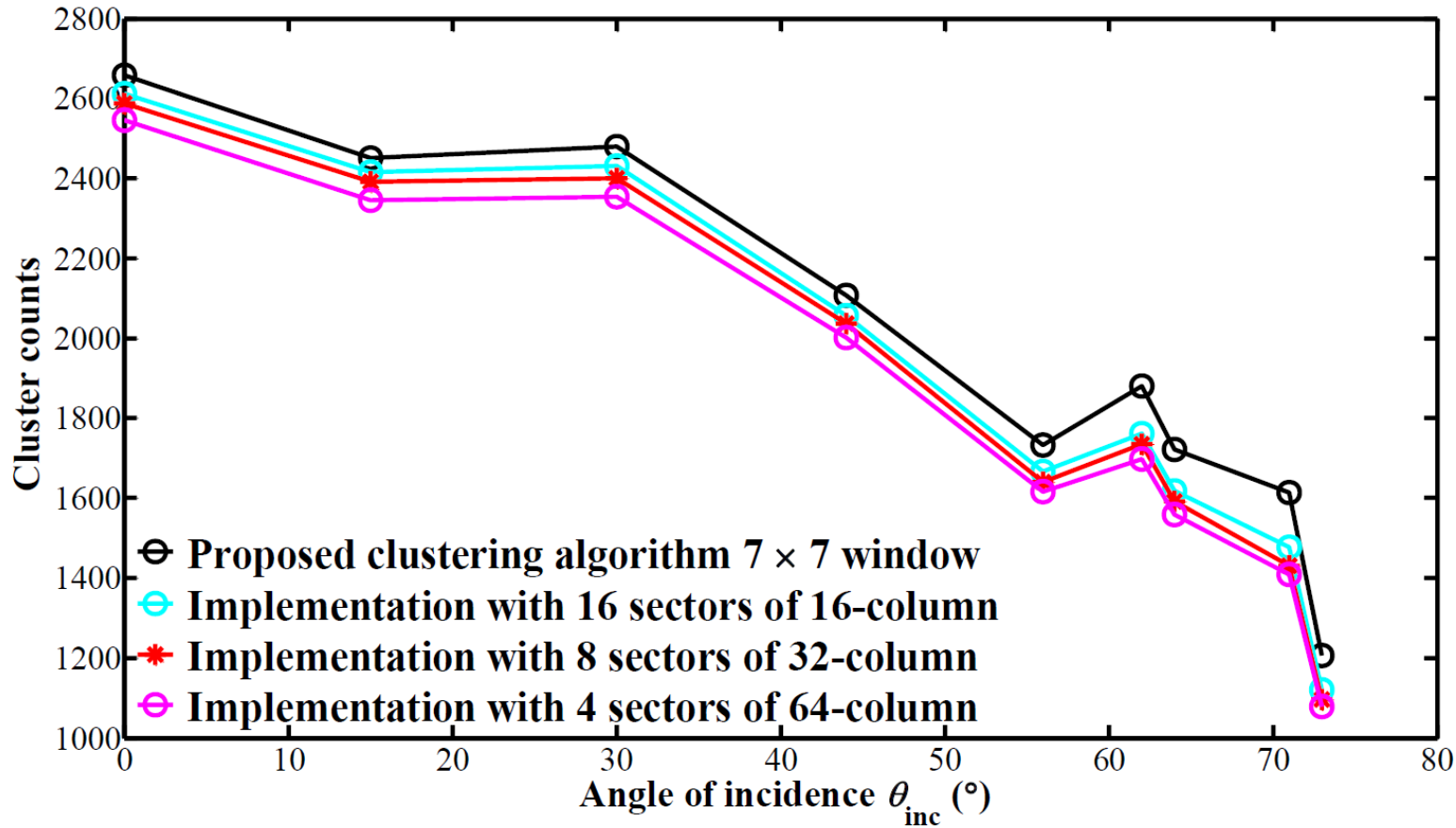
row_max signal信号激活如果当前像素大于左侧三个相邻列的最大值并且不小于右侧三个相邻列的最大值。

提出算法的阵列结构



算法的实现基于 2^N 列的模块。两个32列的模块（ $N=5$ ）。
第一级包含32个找寻模块。
第二级包含7个 32-1 多路选择模块。

算法实现的计数结果仿真



(a) 第二级的引入会引起簇计数的变化。

例如：如果在同一个块中的同一行出现两个种子像素，则位于右侧的种子像素会被丢弃。

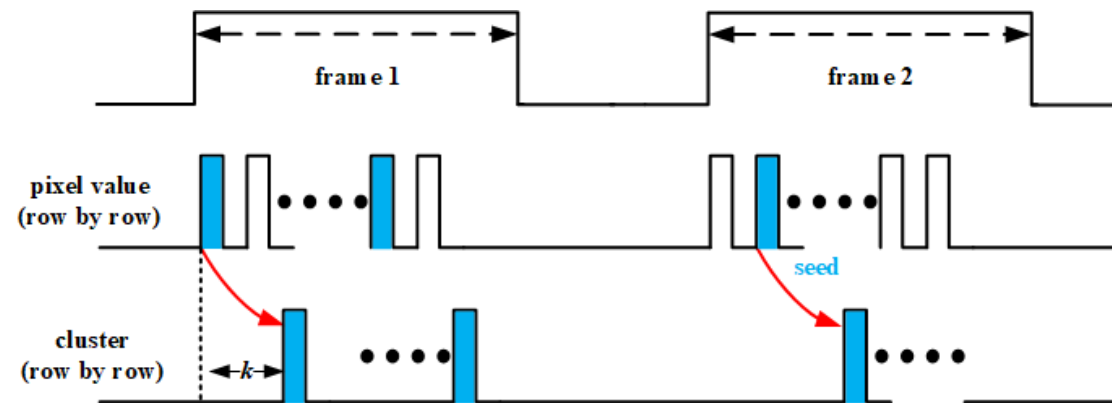
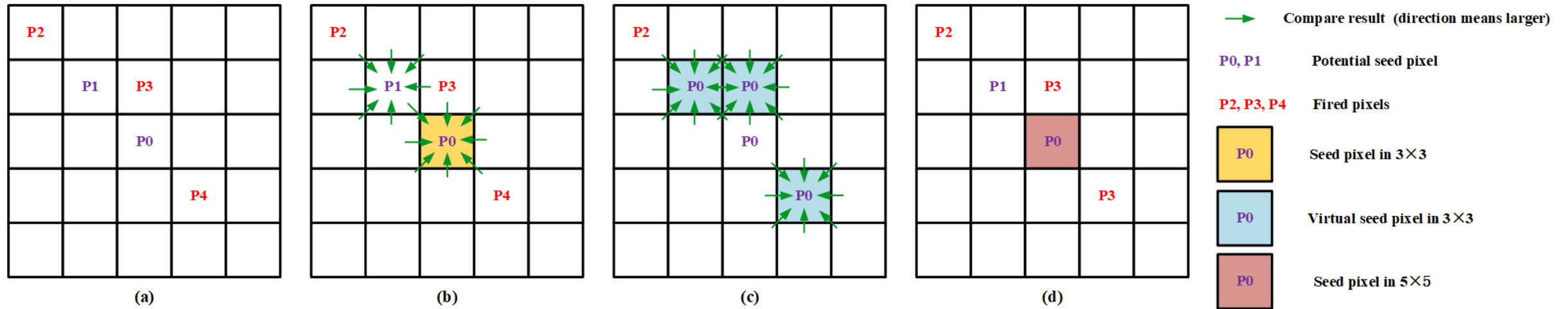
(b) 块的规模越大，上述种子像素被丢弃的概率越大，簇计数也会相对减少。

算法实现的功耗及面积分析（部分）

| Window | Multiplexer | ADC(bits) | Clock(MHz) | Column height(μm) | Column power(mW) |
|--------------|-------------|-----------|------------|--------------------------------|------------------|
| 7 \times 7 | 16-1 | 8 | 100 | 200.58 | 0.85 |
| | | 8 | 200 | 200.15 | 1.83 |
| | | 4 | 100 | 102.86 | 0.46 |
| | | 4 | 200 | 103.39 | 0.88 |
| 7 \times 7 | 32-1 | 8 | 100 | 197.81 | 0.86 |
| | | 8 | 200 | 197.24 | 1.77 |
| | | 4 | 100 | 101.08 | 0.43 |
| | | 4 | 200 | 102.24 | 0.88 |
| 5 \times 5 | 32-1 | 8 | 100 | 159.39 | 0.68 |
| | | 8 | 200 | 159.2 | 1.45 |
| | | 4 | 100 | 81.56 | 0.37 |
| | | 4 | 200 | 82.92 | 0.71 |

采用Tower-Jazz 0.18 μm CMOS工艺综合的面积和功耗结果

基于像素级ADC的簇找寻算法



Clustering algorithm: k is determined by cluster window size