

6.25组会汇报

# TJ-Monopix像素中无Freeze情况下的延迟考虑

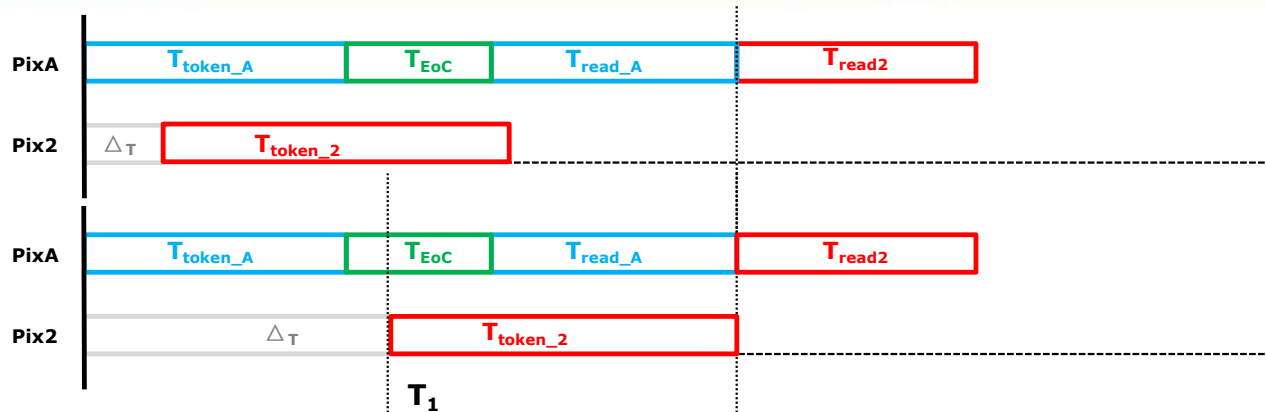
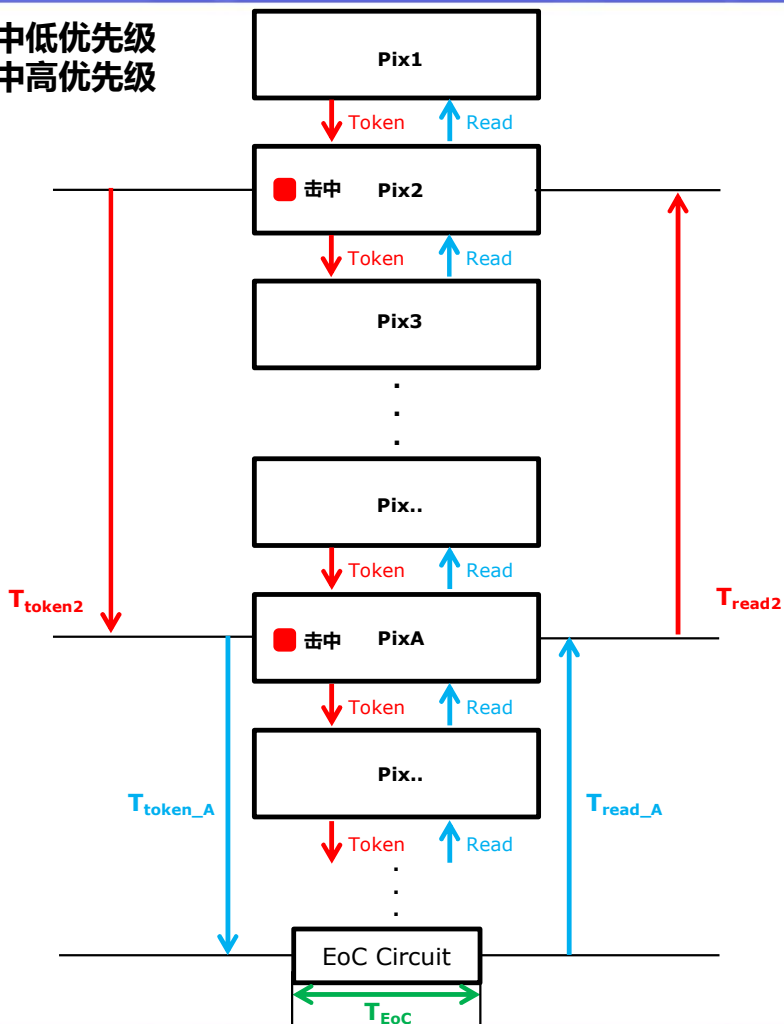
高能物理研究所

郁晗



# 延迟分析

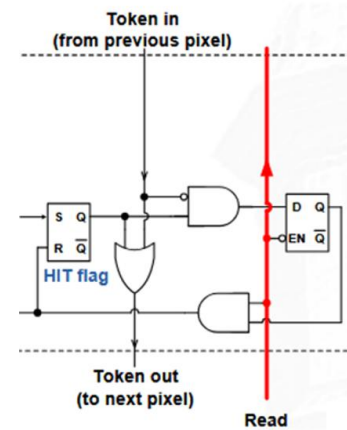
先击中低优先级  
后击中高优先级



- $T_{token2}$  : Pix2 被击中后产生 token 传到 PixA 的时间
- $T_{token\_A}$  : pixA 被击中后产生 token 传到 EoC 的时间
- $T_{read\_A}$  : EoC 收到 token\_A 产生的 Read 传到 PixA 的时间
- $T_{read2}$  : Read 信号从 PixA 传到 Pix2 的时间
- $T_{EoC}$  : EoC 收到 token 后到发出 Read 的时间
- $\Delta T$  : 第一次击中和第二次击中相差的时间

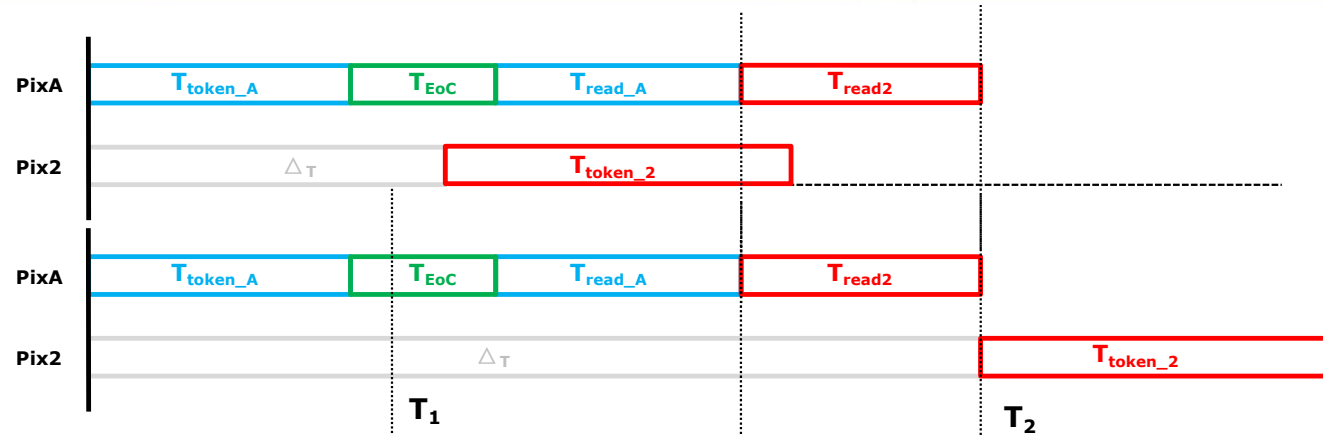
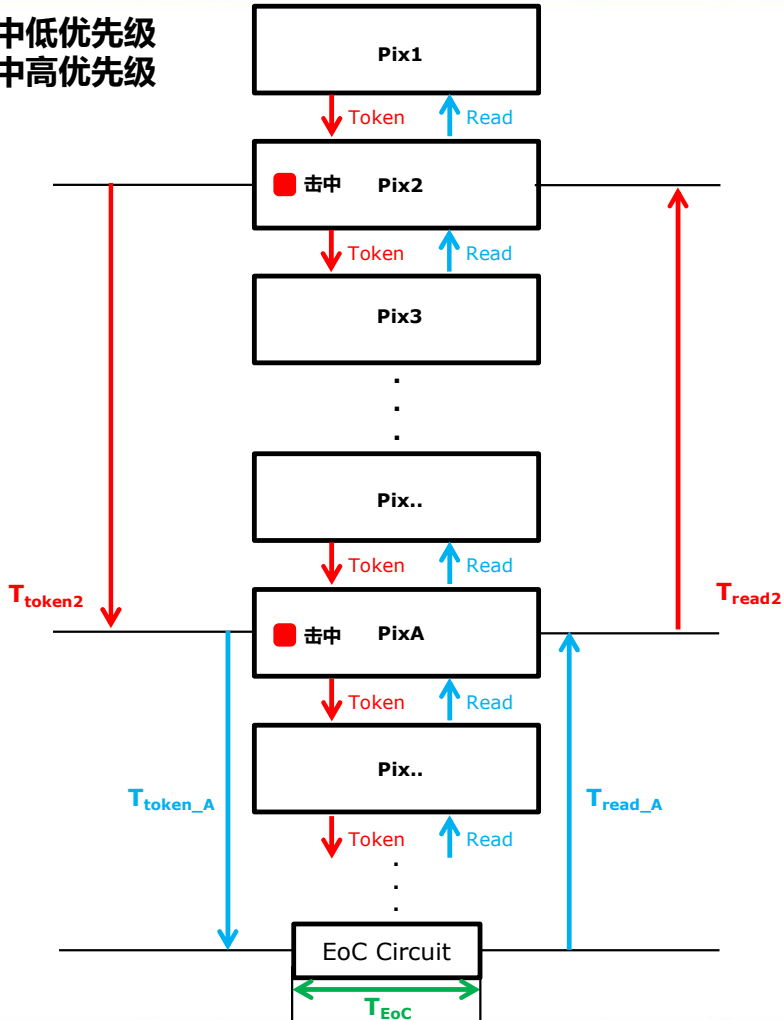
当  $0 < \Delta T < T_1$  先读出 Pix2, 后读出 PixA

Pix2 向下传的 token 先于 Read 到 PixA, 使得 PixA 接受 Read 后无法产生 Readint, 不会 Reset PixA 像素的击中, Read 信号会继续上行直至读出 Pix2 并 Reset Pix2; 后续会继续发 read, 读出 PixA



# 延迟分析

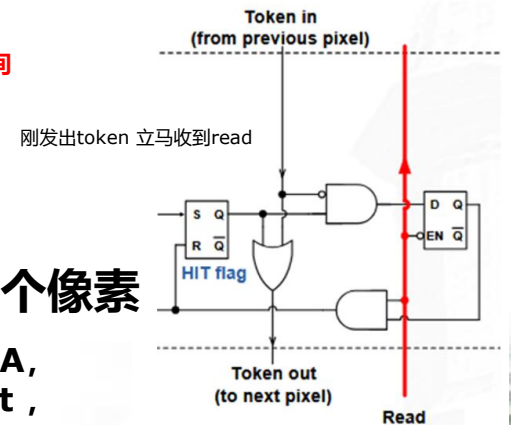
先击中低优先级  
后击中高优先级



- $T_{token2}$  : Pix2 被击中后产生 token 传到 PixA 的时间
- $T_{token\_A}$  : pixA 被击中后产生 token 传到 EoC 的时间
- $T_{read\_A}$  : EoC 收到 token\_A产生的 Read 传到 PixA的时间
- $T_{read2}$  : Read 信号从 PixA 传到 Pix2 的时间
- $T_{EoC}$  : EoC 收到 token 后到发出 Read 的时间
- $\Delta T$  : 第一次击中和第二次击中相差的时间

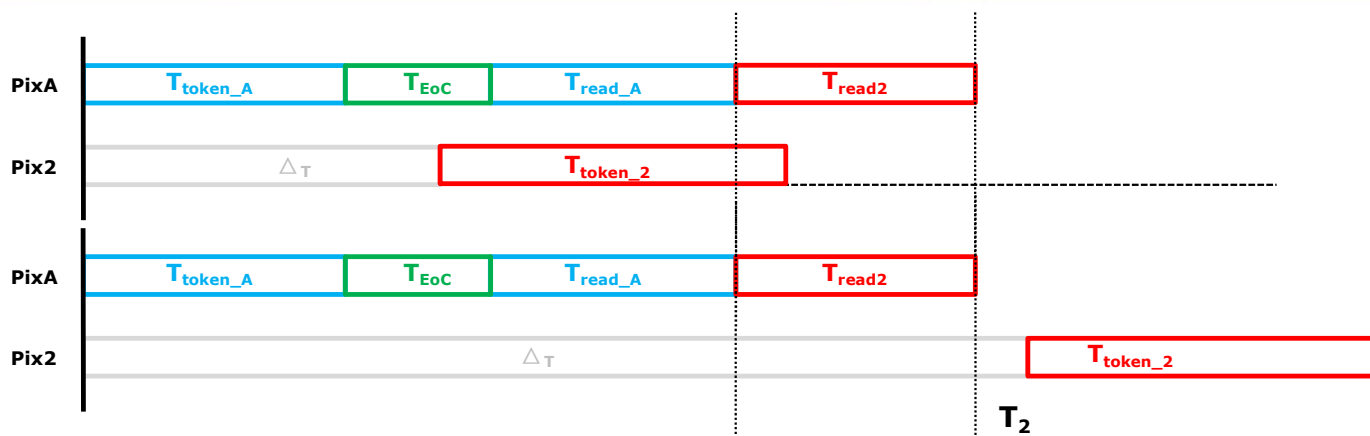
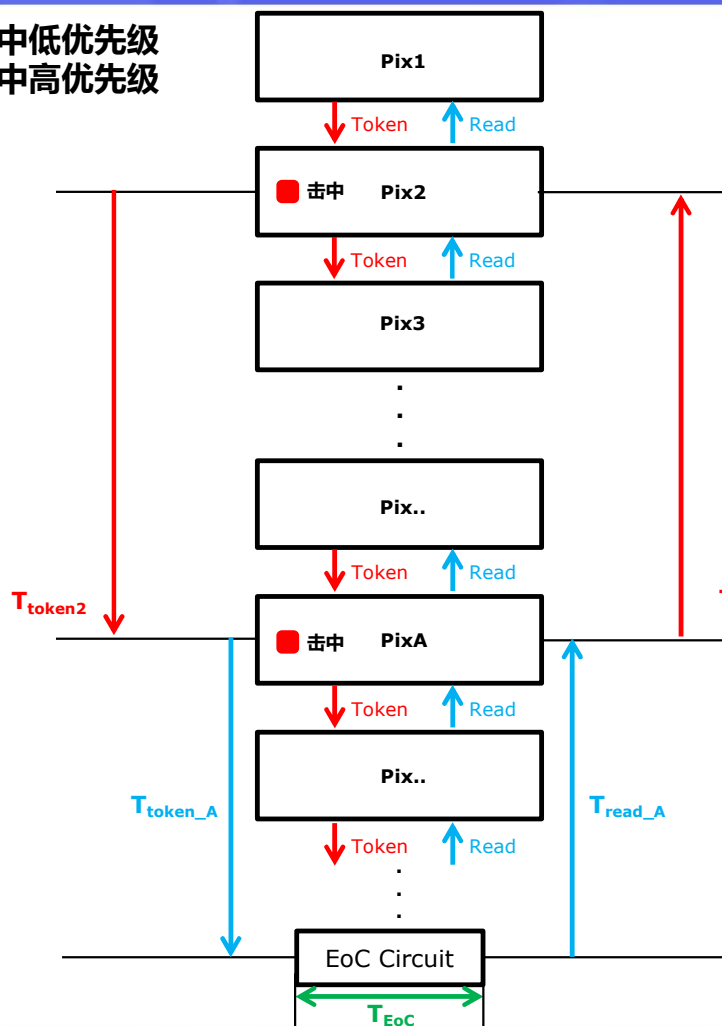
当  $T_1 < \Delta T < T_2$  产生一个Read会读出两个像素

Pix2 向下传的 token 晚于 Read 到 PixA, PixA 会先接收 Read 信号读出地址, 并 Reset, 后继续传至 Pix2, 读出 Pix2的地址并 Reset Pix2



# 延迟分析

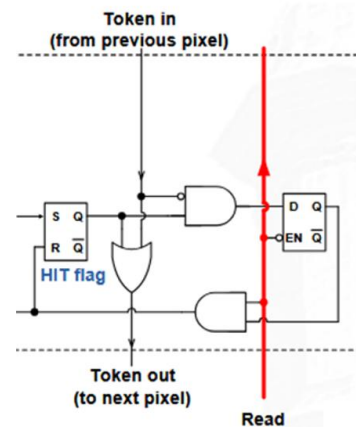
先击中低优先级  
后击中高优先级



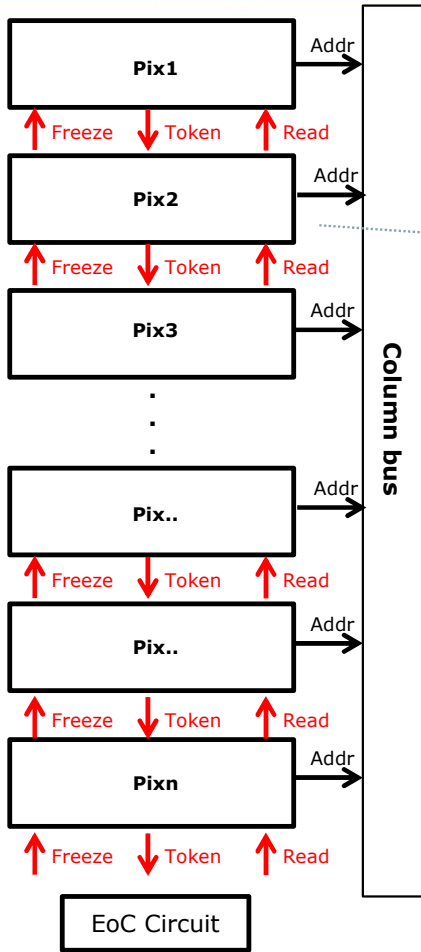
- $T_{token\_2}$  : Pix2 被击中后产生 token 传到 PixA 的时间
- $T_{token\_A}$  : pixA 被击中后产生 token 传到 EoC 的时间
- $T_{read\_A}$  : EoC 收到 token\_A产生的 Read 传到 PixA的时间
- $T_{read\_2}$  : Read 信号从 PixA 传到 Pix2 的时间
- $T_{EoC}$  : EoC 收到 token 后到发出 Read 的时间
- $\Delta T$  : 第一次击中和第二次击中相差的时间

当  $T_2 < \Delta T$  产生两个Read, 读出两个像素

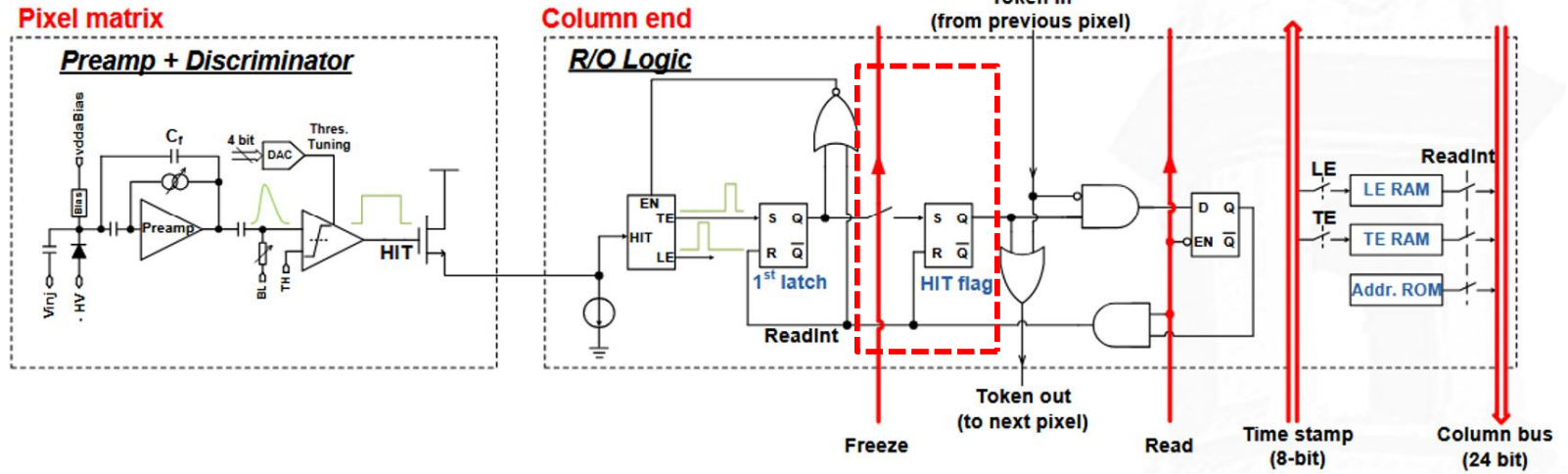
PixA已经被读出, 且Read信号已经传至最上方的像素, 单个像素读出流程已经结束; 此时Pix2被击中, 开始下一次读出流程



# TJ-Monopix像素内数字逻辑



Column drain R/O

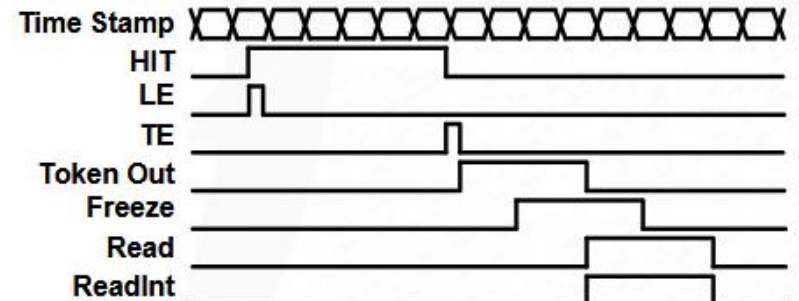


TJ-Monopix像素内数字逻辑原理图

## Freeze部分

有击中像素：在Freeze期间锁住HIT flag  
禁止在Read上升沿时列内产生高优先级的token

无击中像素：允许新的击中存在 First latch



单个像素被击中时序图

# 总结

总结:

不存在freeze信号时，在特定情况下的多次击中，会存在地址多发错发的问题，可能会影响地址读出。  
LJ-Monopix中使用freeze信号，将整列冻结，阻止在像素读出过程中其他击中像素的干扰。

