

# 大面积硅微条探测器在轨数据压缩算法的设计与实现\*

张永强<sup>1,2,3</sup> 郭建华<sup>1,2†</sup> 韦家驹<sup>1,2</sup> 张 岩<sup>1,2,3</sup>

(1 中国科学院紫金山天文台 南京 210033)

(2 中国科学院暗物质与空间天文重点实验室 南京 210033)

(3 中国科学院大学 北京 100049)

**摘要** 硅微条探测器具有位置分辨高、响应快、低噪声、低功耗等优点, 广泛应用在各大加速器试验中, 测量粒子径迹. 新世纪以来, 逐渐应用于空间探测领域. 计划中的“悟空”2号暗物质粒子探测卫星的硅微条探测器将至数十万计, 将产生海量的原始数据. 如何实现探测器快速实时的数据压缩, 是其需要解决的一大难题. 立足于面向空间应用的硅微条探测器在轨实时压缩算法, 算法采用FPGA (Field-programmable Gate Array)搭建流水线结构的方式实现, 在提高系统集成度、节省逻辑资源的同时, 批量数据处理时最高可将数据压缩率提升至38.4 M通道/s. 算法结构具有通用性, 设计思想和方案将为“悟空”2号的径迹探测器的研制提供参考.

**关键词** 仪器: 探测器, 暗物质, 计算方法: 数据压缩

**中图分类号**: P111; **文献标识码**: A

## 1 引言

暗物质问题是当今物理界的热点. 空间间接探测实验方面, 目前在国际上具有较大影响力的在轨高能粒子探测器中, Fermi卫星主要测量100–300 GeV的伽马射线, 接收度高达 $\sim 1 \text{ m}^2 \cdot \text{sr}$ , 在几百GeV能段分辨率却仅为 $\sim 10\%$ <sup>[1]</sup>; AMS-02 (Alpha Magnetic Spectrometer 02)能量分辨率可达2%, 但接收度非常小( $\sim 0.055 \text{ m}^2 \cdot \text{sr}$ )<sup>[2]</sup>; 中国的暗物质粒子探测卫星“悟空”号(DAMPE)的能量分辨率在100 GeV以上能段优于1.5%, 但伽马射线的接收度( $\sim 0.3 \text{ m}^2 \cdot \text{sr}$ )同Fermi相比依然有不小的差距<sup>[3]</sup>; 当下尚无PeV能段的宇宙线直接探测的科学装置. 总的来说, 受仪器探测能力(能量分辨、粒子鉴别能力、接受度等)的限制, 上述空间实验虽然看到了一些暗物质粒子存在的迹象, 如银心的GeV超出<sup>[4]</sup>、宇宙射线中直到 $\sim 1 \text{ TeV}$ 的正电子和反质子超出<sup>[5–8]</sup>、电子能谱在 $\sim 0.9 \text{ TeV}$ 处的

2019-04-02收到原稿, 2019-05-06收到修改稿

\*国家自然科学基金项目(11503095、U1831206、11622327、11873020), 中国科学院A类战略性先导科技专项(XDA15010200), 核探测与核电子学国家重点实验室开放课题(SKLPDE-KF-201812)资助

†jhguo@pmo.ac.cn

拐折等<sup>[9]</sup>,但这些疑似信号仍需要进一步观测才能确定或排除.因此,基于对“悟空”号建造及运行的经验,中国科学院紫金山天文台的团队在提交的《光电空间领域天文子领域(暗物质)中长期发展规划研究报告》中提出了新的空间探测计划“悟空”2号.

“悟空”2号的科学目标为暗物质粒子探测和高能时域天文,它将在继承DAMPE优异的能量分辨率和粒子鉴别本领的基础上,显著强化伽马射线的探测能力,以期在伽马射线谱等暗物质特征谱型的探测方面取得突破性进展,并在GeV–TeV伽马射线天文研究方面取得重要成果.“悟空”2号计划中的工作能段为0.2–20 TeV(电子/伽马),接收度将高达 $3\text{ m}^2\cdot\text{sr}$ (@50 GeV伽马),这就需要超大面积硅微条作为径迹探测器.初步估计,微条通道数至少10倍于DAMPE,触发率将达数千赫兹.

如图1所示,硅微条探测器的本质就是在硅片上集成大量微米量级宽的条形PN结阵列.当带电粒子穿过探测器时,附近的微条便会产生感应信号.通过微条上的信号可以得到入射粒子径迹的一维位置信息,位置分辨率由微条的间距等因素决定,微条上的总信号幅度可以反映入射粒子的电荷数.受硅晶圆生产尺寸的限制,单片硅探测单元的探测面积有限,将多片探测单元沿着微条方向首尾拼接构成一个ladder<sup>[10]</sup>,多个ladder并排布置,便可构成大面积硅微条探测器.“悟空”2号使用的微条间距将在 $100\text{ }\mu\text{m}$ 左右,硅微条探测器的读出通道数50万左右,使用具有多通道微条信号读出功能的ASIC(Application Specific Integrated Circuit)芯片,单芯片实现数十路甚至上百路微条信号读出,以提高系统集成度,降低功耗.

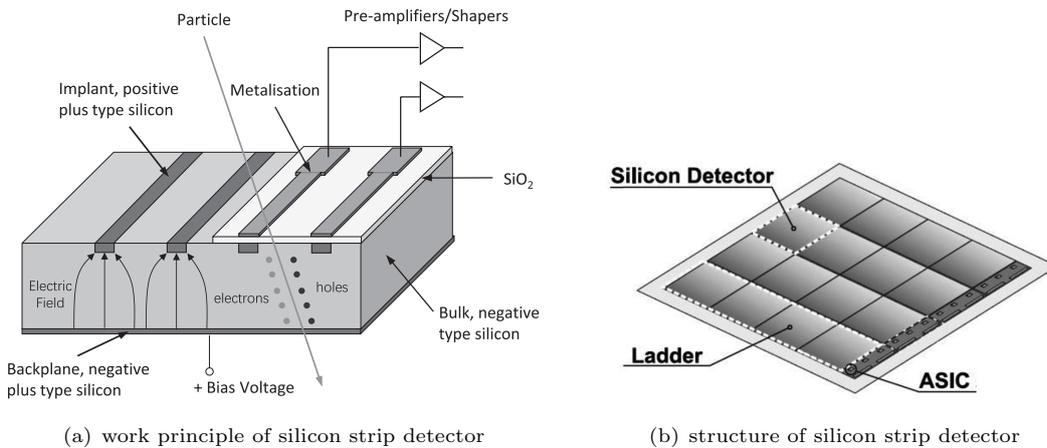


图 1 硅微条探测器结构与工作原理

Fig. 1 The structure of silicon strip detector, and its work principle

“悟空”2号中,硅微条探测器如果每个事例均采集原始信号,则数据量十分庞大.但实际上,入射粒子穿过硅微条探测器时,仅有少量微条通道着火(即被击中),其余通道皆处于基线无信号状态<sup>[11]</sup>.因此,可以剔除大量未击中通道,仅将着火通道数据存储或下传,以大大缓解星上临时存储和数据下传的压力.硅微条探测器上寻找着火通道、剔除空闲通道的过程又被称为微条数据压缩或过“零”压缩.

“悟空”2号触发率达数千赫兹,如何在轨实现快速有效的微条数据压缩极为关键.为此,本文针对“悟空”2号的需求,提出了一种实现硅微条探测器高速数据压缩的新思

路, 并完成了算法设计, 在实验室已有探测器基础上, 搭建了验证平台, 给出了算法验证结果.

## 2 实时压缩算法的设计与实现

### 2.1 已有数据压缩解决方案比较与分析

在硅微条探测器的空间应用中, 数据压缩算法的速率直接影响到数据获取系统的采集速率和资源消耗, 进而影响整个探测器的探测效率和成本. 国内外主要空间高能粒子探测任务中, 硅微条探测器信号获取和数据压缩的典型设计方案如表1所示.

表 1 硅微条探测器信号获取与压缩的典型设计方案  
Table 1 Typical methods of data acquisition and compression for the silicon strip detectors

Mission	Strip Channel Number	Readout Chip	Processor Type	Information Extracted	Maximum Event Rate/Hz	Data Compression Rate/(Channels · s <sup>-1</sup> )
Fermi	800 k	GTFE+GTRC (California University)	FPGA	fired channel ID	10 k	18 M
AMS-02	200 k	VAHDR9 (IDEAS)	FPGA+DSP	fired channel ID fired signal amplitude	3 k	2.5 M
DAMPE	70 k	VA140 (IDEAS)	FPGA	fired channel ID fired signal amplitude	330	1.4 M

Fermi的设计者定制了2款ASIC, 完成了星上硅微条探测器从多路信号读出、信号幅度比较、数据压缩等一整套的数据采集与压缩过程<sup>[12]</sup>. 探测器的每个数据获取单元负责处理1792路通道, 仅记录各通道是否着火, 使用二进制“0”和“1”表示. 整体实现了10 kHz事例率下的数据采集, 单个硬件压缩算法模块的压缩运算速率约为18 M通道/s.

AMS-02和DAMPE均选用IDEAS公司设计的面向硅微条探测器读出的VA系列电荷测量芯片<sup>[13-15]</sup>, 实现多路探测器通道信号的读出. AMS-02使用DSP (Digital Signal Processing)芯片完成数据压缩, 每颗DSP负责832路通道数据, 最高采集速率达3 kHz, 单个压缩算法模块的压缩运算速率约为2.5 M通道/s. DAMPE的数据压缩处理由FPGA完成, 每颗FPGA负责4608路数据压缩, 最高采集速率为330 Hz, 单个压缩算法模块的压缩速率约为1.4 M通道/s. AMS-02和DAMPE上硅微条探测器同时记录了探测器着火通道的位置和信号幅度, 以获得更加精确的高能粒子径迹. 另一方面, 着火通道的信号幅度, 可以反映入射粒子的电荷量, 辅助粒子鉴别.

对比3者, Fermi的压缩速率最高, 却丢弃了信号幅度信息; AMS-02利用FPGA搭配DSP完成探测器数据的收集和压缩, 同时保留了位置和信号幅度信息, 但由于其单颗DSP仅能负责832路数据压缩, 集成度偏低, 压缩速率也偏低; DAMPE的数据获取和压缩处理均在FPGA上完成, 集成度比AMS-02高, 但压缩速率以及能支持的事例率却

均是3者最低。“悟空”2号拟借鉴DAMPE或AMS-02,同时保留硅微条上的位置和幅度信息,探测器信号获取方案主要沿用DAMPE:使用VA140实现多通道微条信号读出;读出信号的进一步调理、数字化及数据收集分别由运放电路、ADC (Analog-to-Digital Converter)和FPGA完成。由于FPGA在数据处理上具备天然的并行性优势,“悟空”2号依然选择在FPGA上完成数据压缩处理,通过对算法架构进行并行化设计,实现高速压缩,数据获取与压缩处理流程如图2所示。

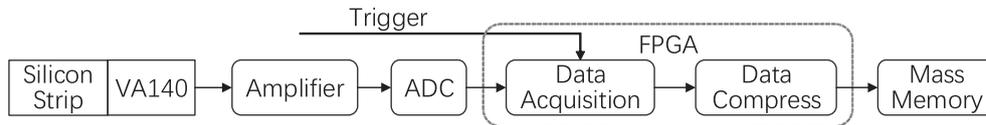


图2 “悟空”2号硅微条探测器数据获取与压缩处理流程

Fig. 2 Process of data acquisition and compression for the silicon strip detector of DAMPE-02

## 2.2 硅微条探测器压缩算法流程

硅微条探测器通常包括多个ladder,数据压缩以ladder为单位进行批量处理,算法流程可以分解为如下两个步骤<sup>[16-17]</sup>:

### (1) 预处理

硅微条读出和数字化获得的原始数据由通道基线、共模噪声和微条实际信号3部分叠加构成,在判别着火通道前,必须先行扣除,即预处理。预处理主要包括基线扣除、共模计算和共模扣除3个步骤:

#### (a) 基线扣除

基线主要由读出ASIC芯片的特性和ADC之前的调理电路共同决定,通道间的电子学存在个体性差异,基线水平并不一致,必须逐一扣除,以显现出信号的真实幅度;

#### (b) 共模计算

共模噪声(Common Mode noise, CM),即数据采集时由各种环境因素引起多个通道信号电平的整体偏移,主要来自多路读出ASIC芯片,伴随着每一次的数据采集,具有实时性。将该次采集中的未着火通道扣除基线后求取平均,即可获得该次的共模噪声;

#### (c) 共模扣除

将步骤a中已扣除基线的通道数据再减去由步骤b获得的该通道共模噪声,即可获得微条实际信号。

### (2) 寻找信号簇(cluster)

带电粒子穿过探测器时,通常会引起1至数个相邻微条通道着火,这些着火通道信号的加和同入射粒子的沉积能量相对应。这些相邻的着火通道被称为cluster,寻找着火通道又常称为寻找cluster。扣除基线和共模噪声后,可通过设置适当的信号阈值判定通道是否着火,判别过程如图3所示。

为寻找cluster,通常设置两种阈值:着火阈值和种子阈值,种子阈值大于着火阈值。依次遍历一个ladder上的通道,判断该通道是否大于种子阈值。当该通道大于种子阈值时,判断其前后相邻通道是否大于着火阈值。着火通道同种子通道一起构成一个cluster,表征入射粒子在该ladder的位置及能量沉积。

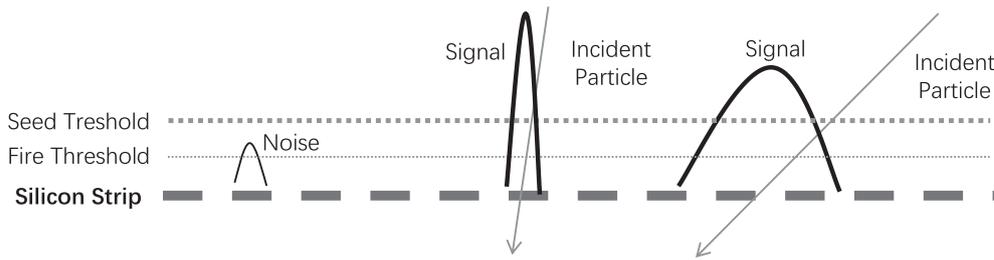


图3 cluster判别

Fig. 3 Cluster discrimination

### 2.3 流水线结构的算法设计

传统处理器核心通常只包含一个运算单元和一个控制单元, 单个时钟周期只能执行一次操作, 在批量数据高速实时处理方面难有大的作为. FPGA实现功能主要靠的是内部硬件资源的堆叠, 生成的运算模块是独立的硬件. 结合算法运算特点, 在FPGA中构建流水线结构, 使尽可能多的运算模块同时运行起来, 往往可以在处理速度上实现大的突破.

依据2.2节压缩算法流程, 将整个算法的实现分为预处理和寻找cluster两部分. 预处理部分负责扣除原始数据的基线和共模噪声, 依次包含减基线、共模计算和减共模3个运算模块. 寻找cluster部分负责通过阈值比较判断微条通道是否着火, 保留着火通道数据.

分析算法各运算模块特点可知, 除共模计算外, 其余模块皆为逐通道遍历运算, 且皆可单步骤完成, 十分方便连续批量处理. 唯独共模计算必须要完成一个ASIC里相关读出通道数据累加后, 再做除法运算才能完成, 打断了运算的连续性. 为解决共模计算带来的停顿, 引入两个存储模块构成乒乓结构作为中间缓存, 将整个算法分为两拍执行: 第1拍, 读取原始数据, 完成减基线、共模计算, 并将扣除基线后的数据存入中间缓存Buffer A (或Buffer B); 第2拍, 读取第1拍扣除基线后的数据缓存Buffer A (或Buffer B), 进行减共模运算、寻找cluster, 同时减基线模块和共模计算模块展开对另一个ladder原始数据的处理, 并将扣除基线后的数据存入缓存Buffer B (或Buffer A). 两拍耗时相同, 均为ladder通道数×FPGA时钟周期, 算法结构如图4所示.

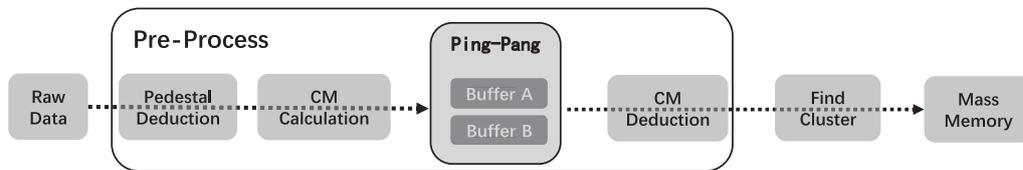


图4 数据压缩算法结构

Fig. 4 Structure of the data compression algorithm

图4中的每个功能模块在FPGA中均有具体的硬件对应, 它们相互同步、独立运行. 原始数据按照通道先后顺序, 以FPGA时钟为节奏构成数据流, 源源不断进入各运算模块, 中间不作停顿, 算法处理时间由数据流长度和时钟频率共同决定, 运行过程如图5

所示.

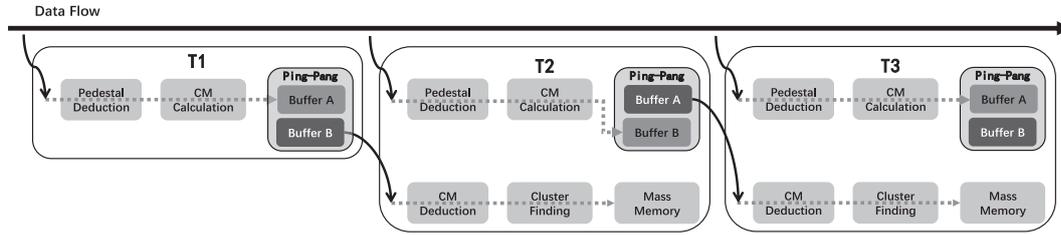


图 5 数据压缩算法运行过程

Fig. 5 Operation of the data compression algorithm

使用乒乓双缓存将算法执行分为两拍的好处在于: 所有运算模块始终处于忙碌状态, 将算法占用的FPGA资源利用到了极致, 大大缩短了算法整体运行时间. 当此种结构算法为多个ladder做批量压缩处理时, 理想情况下, 一个ladder的平均处理时间 $t$ 可以表示为

$$t = (N_{\text{channel}} \times T) \times (N_{\text{ladder}} + 1) / N_{\text{ladder}}, \quad (1)$$

其中,  $N_{\text{channel}}$ 为单个ladder包含的微条通道数,  $T$ 为FPGA时钟周期,  $N_{\text{ladder}}$ 为连续参与压缩运算的ladder个数.

由上述表达式可知, 连续参与运算的ladder越多, 平均压缩速率越快,  $N_{\text{ladder}}$ 趋于无穷时, 单个ladder压缩所耗时间无限接近 $N_{\text{channel}} \times T$ , 折合每个通道的数据压缩仅需要一个FPGA时钟周期, 压缩速率约等于FPGA时钟频率(单位: 通道/s). 即便仅对单个ladder做压缩处理, 所耗时间为 $2 \times N_{\text{channel}} \times T$ , 压缩速率也可高达FPGA时钟频率的一半.

## 2.4 算法实现

### 2.4.1 预处理

预处理部分算法结构如图6所示, 各模块具体实现过程如下:

#### (1) 减基线模块

数据流首先进入减基线模块, 该模块由一个减法器 and 两个RAM (Random Access Memory) 读取控制模块构成, 将原始数据流和基线数据按照相同的通道顺序依次导入该减法器, 即可实现基线扣除操作. 扣除基线之后的数据流同时进入缓存和共模计算模块;

#### (2) 共模计算模块

如2.2节所述, 将同一个ASIC中未着火通道依次扣除基线, 求取平均, 即可获得该ASIC芯片在此次数据集中的共模噪声. 实际操作过程中, 在扣除共模前, 无法预先对每个通道是否着火给出确切判别. 对于信号幅度较小的着火通道, 即便直接参与共模计算, 经过共模计算中除法平均以后, 后续影响也可以忽略不计; 而对于幅度较大的着火通道信号, 特别是自身电荷数较多的粒子产生的大信号, 容易对计算结果造成干扰, 且容易辨别, 需要加以剔除. 因此, 在共模计算时, 首先依据共模的分布设置一个合适的共模阈值, 以排除较大信号带来的影响. 本文将在3.1节通过具体分析, 给出针对地面MIP (Minimum Ionizing Particle) 事例的共模阈值选取方法.

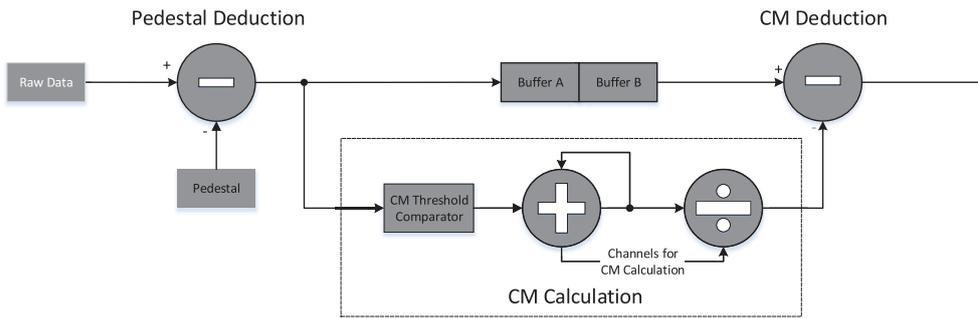


图6 FPGA上算法预处理结构框图

Fig. 6 Structure of the data pre-processing in FPGA

基于上述分析, 将共模计算模块设计为3个子模块: 共模阈值比较子模块、通道数据累加子模块和除法器子模块. 扣除基线的数据首先进入共模阈值比较子模块, 该模块利用比较器判别各通道数据是否大于设置的共模阈值, 若结果为“真”, 则该通道数据不参与后续共模运算环节; 若结果为“假”, 该通道数据将继续进入通道数据累加子模块.

通道数据累加子模块由一个循环累加器构成, 接收来自共模阈值子模块筛选后的通道数据, 做累加运算. 每完成一个读出ASIC中相应通道数据的累加, 该模块便立刻将累加结果以及参与累加运算的通道数发送给除法器子模块, 分别作为被除数和除数; 同时将累加和以及累加通道计数清零, 继续开始新一轮累加运算.

除法器子模块主要由一个除法器构成, 接收来自通道数据累加子模块的除数与被除数, 进行除法运算, 运算结果送入后面的减共模模块. 在FPGA中, 单时钟实现除法运算, 需要耗费巨大的逻辑资源, 在空间应用中极不可取. 因此, 本文采用了移位相减的方式实现除法, 每个时钟周期完成一个bit的处理, 一次取整除法运算, 所耗时钟周期数为除数与被除数的bit数之和. 为了简化计算, 可将除法器中设定除数与被除数为固定位宽, 因此所需时间也是固定的. 就本算法而言, 共模求取中的一次除法运算通常不会超过30个时钟周期.

### (3) 减共模模块

当一个ladder的所有通道完成减基线并存入中间缓存后, 按照通道顺序读取该缓存, 再次形成数据流进入减共模模块. 该模块仅包含一个减法器, 将进入该模块的通道数据依次减去共模计算模块获得共模值, 所得差值继续进入下一个运算环节.

#### 2.4.2 寻找cluster

根据cluster判别原理, 结合FPGA并行性, 将寻找cluster部分设计为3个子模块构成: 着火通道判别子模块、种子通道判别子模块和仲裁子模块, 具体结构如图7所示.

预处理以后, 数据流同时进入着火通道判别子模块、种子通道判别子模块和仲裁子模块. 着火通道判别子模块主要负责读取着火阈值RAM获得相应通道的着火阈值, 通过比较器判断数据流中各通道数据是否着火. 同着火通道判别子模块类似, 种子通道判别子模块通过比较器最终给出最近一次过阈的通道号记为 $ID_{seed}$ . 当着火通道判别子模块的着火信号由“假”变为“真”时, 仲裁子模块将发起一次cluster判别, 过程如下:

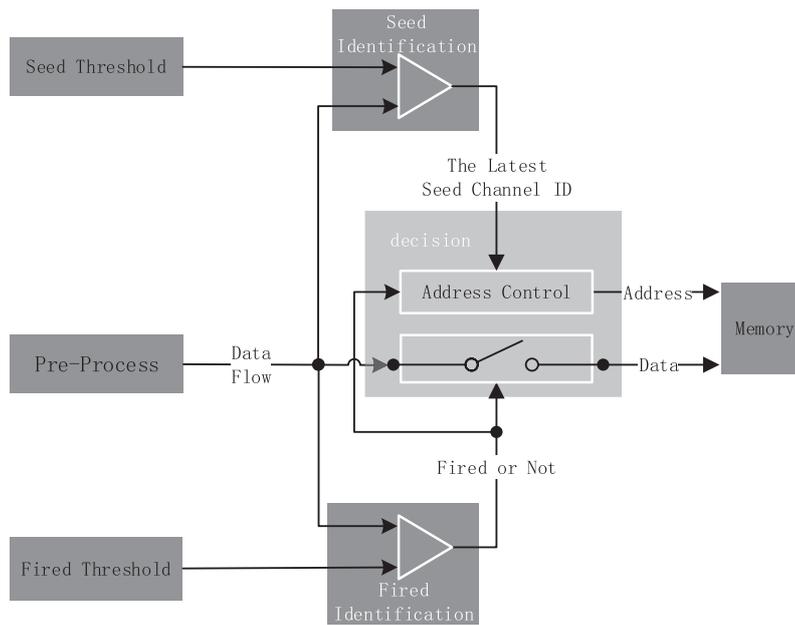


图 7 FPGA上寻找cluster结构框图

Fig. 7 Structure of the cluster finding in FPGA

(1)将相应通道数据写入存储器, 存储器地址控制寄存器加1, 直至着火信号为“假”, 此时保存在存储器里的通道数据为疑似cluster数据, 并记录着火开始时对应的通道号为 $ID_{start}$ , 记录着火停止时对应的通道号为 $ID_{stop}$ ;

(2)在着火信号由“真”变为“假”的同时, 仲裁子模块查询种子通道判别子模块的结果, 当 $ID_{seed} \in [ID_{start}, ID_{stop}]$ 时, 该次写入存储器的cluster数据为有效cluster数据; 否则, 该次写入存储器的cluster数据为无效cluster数据, 仲裁子模块将缓存地址寄存器恢复至该次写入操作之前的值.

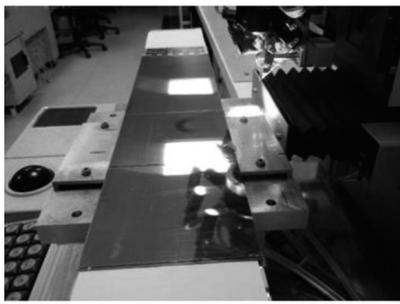
此种寻找cluster的实现形式, 一次通道数据遍历, 即可完成该ladder上的cluster寻找, 所耗时间仅为ladder通道数 $\times$ FPGA时钟周期.

### 3 实时压缩算法的验证

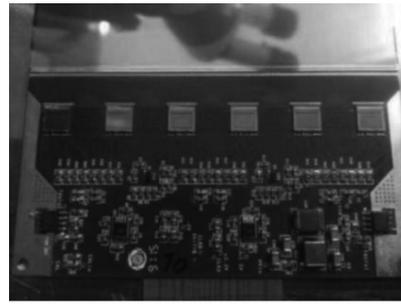
#### 3.1 算法验证环境

基于在DAMPE任务中对硅微条探测器研制的经验和技術积累, 本实验室展开了对硅微条探测器关键技术的研究, 设计组装了基于VA140前端读出的硅微条探测器样机<sup>[10]</sup>, 如图8所示.

样机使用了同DAMPE上参数相同的单面硅微条探测单元, 微条间距 $121 \mu\text{m}$ , 包含一个ladder共384个读出通道, 测得探测器基线水平、共模分布、读出电子学线性等基本电气特性如图9所示. 其中, 图9 (d)为微条读出电子学线性度标定结果, 对其进行线性拟合如图中黑色实线所示. 线性拟合参数中,  $\chi^2$ 表示拟合方差,  $ndf$ 表示拟合的自由度个数,  $p0$ 代表线性拟合的截距,  $p1$ 代表线性拟合的斜率.



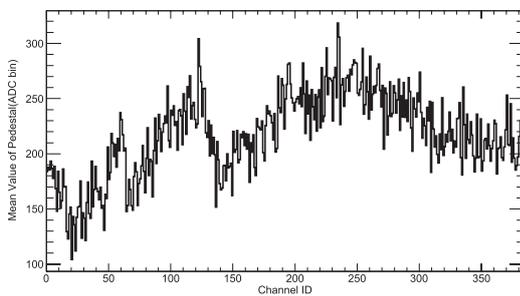
(a) ladder assembled by silicon strip unit



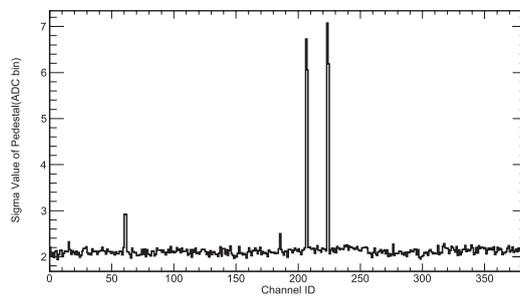
(b) front-end electronics

图 8 实验室硅微条探测器样机设计

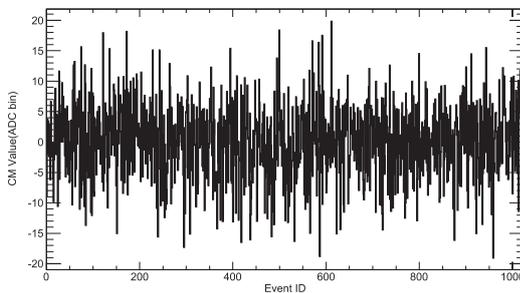
Fig. 8 Prototype design of the silicon strip detector in the lab



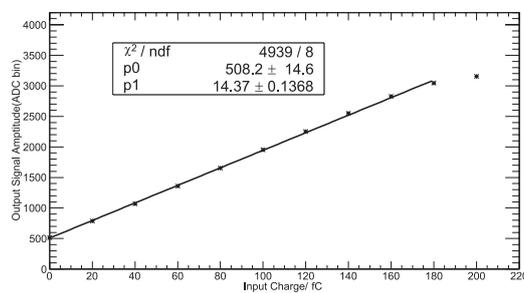
(a) mean value distribution of the strip channels' pedestal



(b) sigma value distribution of the strip channels' pedestal



(c) a typical channels' common mode noise distribution



(d) a typical channels' electronics linearity

图 9 探测器基本电气特性

Fig. 9 Electronics characteristics of the silicon strip detector

使用该探测器测得实验室环境下宇宙线最小电离粒子(MIP)能谱如图10所示。

实时压缩算法的验证工作基于上述探测器展开。为验证压缩算法的性能,实验室设计了一套简易的算法验证平台,如图11所示。该平台通过USB接口将上述探测器样机获取的宇宙线事例数据由PC快速发送给FPGA中的原始数据缓存,控制原始数据缓存以事例为单位将探测器原始数据(共384通道)批次导入数据压缩算法模块,算法模块处理的中间结果或最终结果存入压缩结果缓存,最终通过USB接口发送给PC,同离线分析结

果作对比, 以达到算法调试和验证的目的.

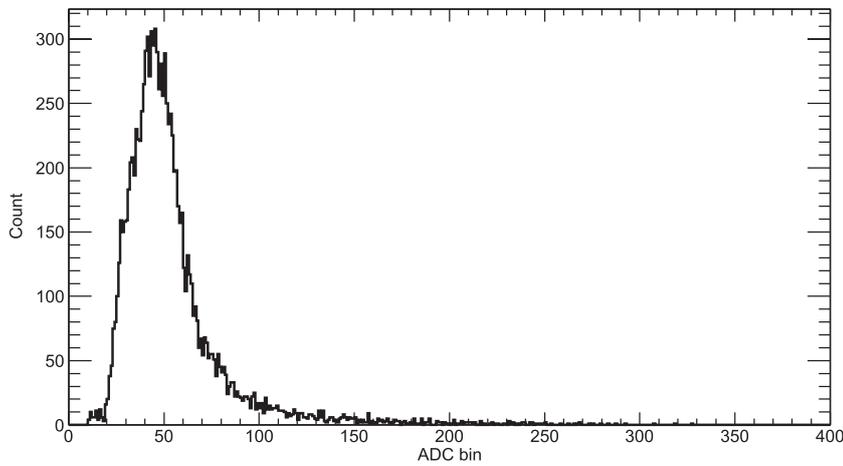
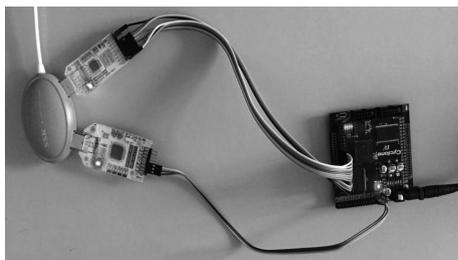
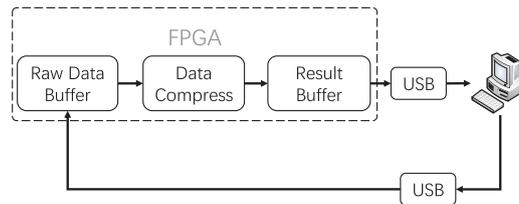


图 10 MIP能谱

Fig. 10 MIPs spectrum



(a) verification platform



(b) structure of the verification platform

图 11 实时压缩算法简易验证平台

Fig. 11 Verification platform for the real-time data compression algorithm

图10所示MIP峰位ADC约为50, 依据图9 (c)所示共模分布可知, 绝大多数情况下, 共模噪声绝对值在20以内. 因此, 针对MIP事例, 算法的共模运算中选择共模阈值为20以减少较大着火信号对共模计算的偏差. 对于幅度较小的着火通道信号, 参与共模求平均后, 计算结果的偏差将在1个ADC bin以内, 基本可以忽略不计.

如上文的第2.2节和图3所示, 入射粒子通常会导1至多个微条通道产生感应信号, 距离入射粒子击中位置最近的微条信号最大, 距离较远的微条信号较小. 又由图9 (b)和图10可知, 硅微条通道基线符合高斯分布, 标准差在2左右(大于3通道的被认定为坏道, 不作处理), 远低于MIP信号峰位. 为了尽可能排除探测器通道基线噪声的影响, 减少误判, 且不影响MIP信号的获取, cluster识别条件设定如下: 首先设置种子阈值为5倍的该通道基线标准差(该通道基线ADC值超过该值的概率低于0.0001%), 信号过阈则认定为MIP事例; 由于带电粒子穿过探测器时可能会引起数个相邻微条通道着火, 这些着火通道信号的加和同入射粒子的沉积能量相对应, 试验中为了较为准确地获得MIP事例能量沉积, 设置着火阈值为3倍的该通道基线标准差(该通道基线ADC值超过该值的概率低

于0.26%), 种子通道周围如果恰好有通道信号大于此阈值, 则认为该过阈通道信号同样来自于MIP事例粒子, 该通道信号应该予以保留.

### 3.2 验证结果

PC离线数据处理和FPGA在线处理得到的MIP能谱如图12所示. 可以发现, 2者基本重合, 表明FPGA上的压缩算法结果正确, 误差在合理范围. 分析认为, 由于FPGA上的实时压缩处理涉及到的运算均采用整型处理以节省逻辑资源, 运算精度略差于PC离线处理采用的浮点运算, 致使2者结果出现细微的偏差. 将PC离线压缩处理中的运算改用整型, 再次同FPGA压缩结果比对发现, 2者完全一致.

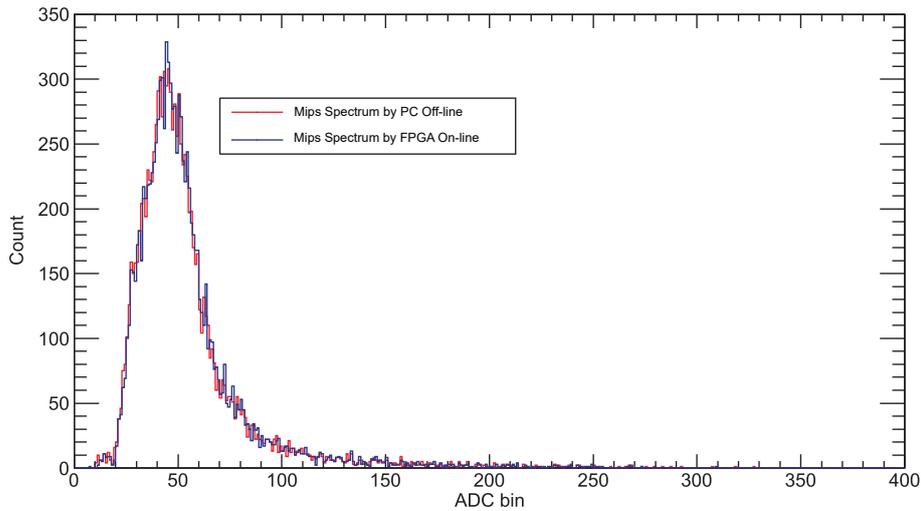


图 12 PC离线数据处理与FPGA在线处理得到的MIP能谱对比

Fig.12 MIPs spectrum comparison between the PC off-line procession and FPGA on-line procession

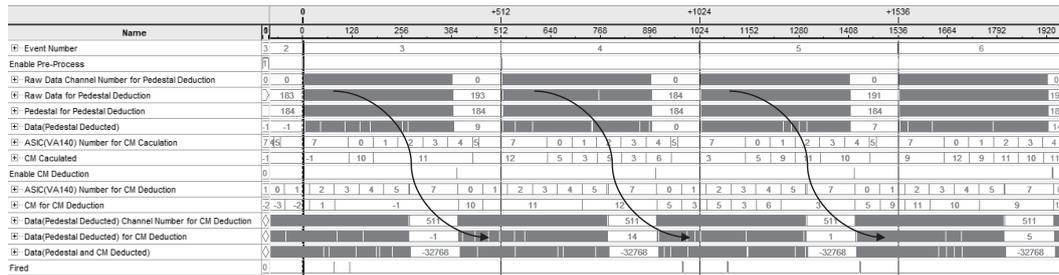
最终得到的数据压缩率如表2所示.

表 2 压缩率统计  
Table 2 Compression rate

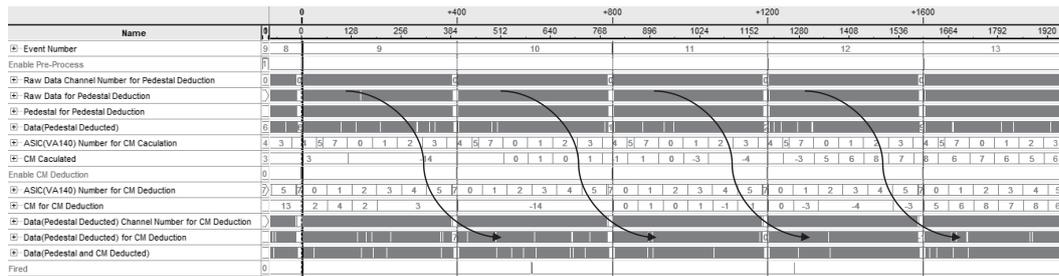
total event number	amount of data before compression/byte	amount of data after compression/byte	compression rate
62085	48177960	586716	1.2%

设置算法验证平台中FPGA原始数据缓存模块向压缩算法模块批量导入数据的时间间隔, 在不同的间隔下, 通过Altera公司的综合性CPLD/FPGA开发软件Quartus II可以观测到算法运行过程中的实际数字信号波形, 如图13所示.

如图13所示, 实际测得, 批量处理ladder数据时, 该压缩算法仅用400个时钟周期即可完成384通道数据压缩运算. 算法验证平台中的FPGA时钟频率为40 MHz, 批量ladder数据处理时, 对应算法压缩速率最高可达38.4 M通道/s, 远超目前所有硅微条探测器在空间应用项目中所能达到的在轨数据压缩的处理速度.



(a) import a detector's raw data packet per 512 clock period



(b) import a detector's raw data packet per 400 clock period

图 13 FPGA 上算法运行过程波形图

Fig. 13 Signal waveform during the algorithm operation in FPGA

### 4 总结

硅微条探测器应用于空间探测在我国尚处起步阶段，亟需在探测器关键技术上独立自主探索。本文为硅微条探测器量身定制了一种基于FPGA流水线结构的在轨数据压缩算法，通过极致的并行化处理，寻求压缩速率的飞跃式突破。实验室环境下对MIP事例进行批量处理时可实现高达38.4 M通道/s的数据压缩，压缩速率远超目前所有相关空间探测项目。该算法设计既可以满足我国计划中的“悟空”2号暗物质粒子探测卫星对大面积硅微条探测器的应用需求，也为我国其他大型空间观测项目上硅微条探测器的运用提供了经验积累和有价值的参考。

## 参考文献

- [1] Atwood W B, Abdo A A, Ackermann M, et al. *ApJ*, 2009, 697: 1071
- [2] Tinge S. *NuPhS*, 2013, 243: 12
- [3] Chang J, Ambrosi G, An Q, et al. *APh*, 2017, 95: 6
- [4] Liang Y F, Shen Z Q, Li X, et al. *PhRvD*, 2016, 93: 103525
- [5] Aguilar M, Aisa D, Alpat B, et al. *PhRvL*, 2014, 113: 221102
- [6] Aguilar M, Alberti G, Alpat B, et al. *PhRvL*, 2013, 110: 141102
- [7] Accardo L, Aguilar M, Aisa D, et al. *PhRvL*, 2014, 113: 121101
- [8] Aguilar M, Aisa D, Alpat B, et al. *PhRvL*, 2015, 114: 171103
- [9] DAMPE Collaboration. *Nature*, 2017, 552: 63
- [10] 韦家驹. *核技术*, 2018, 41: 120402
- [11] Wu X. Triggering on (Low Energy) photons and data volume of STK. DAMPE Workshop, November 2013
- [12] Atwood W B, Hernando J A, Hirayama M, et al. *NIMPA*, 1999, 435: 224
- [13] Ambrosi G. *NIMPA*, 1999, 435: 215
- [14] Zhang F, Fan R, Peng W X, et al. *ChPhC*, 2014, 38: 066101
- [15] VA140 documentation V0R1. *gamma Medica-Ideas*, 2016
- [16] Dong Y F, Zhang F, Qiao R, et al. *ChPhC*, 2015, 39: 116202
- [17] Bulgarelli A, Argan A, Barbiellini G, et al. *NIMPA*, 2010, 614: 213

## The Design and Implementation of On-orbit Data Compression Algorithm for Large Area Silicon Strip Detector

ZHANG Yong-qiang<sup>1,2,3</sup> GUO Jian-hua<sup>1,2</sup> WEI Jia-ju<sup>1,2</sup> ZHANG Yan<sup>1,2,3</sup>

(1 Purple Mountain Observatory, Chinese Academy of Sciences, Nanjing 210033)

(2 Key Laboratory of Dark Matter and Space Astronomy, Chinese Academy of Sciences, Nanjing 210033)

(3 University of Chinese Academy of Sciences, Beijing 100049)

**ABSTRACT** The silicon strip detector (SSD) has the advantages of high position resolution, fast response, low noise, and low power consumption. Since the new century, it has been widely used in space exploration to track incident particles. SSD for the new DARK MATTER PARTICLE EXPLORER-02 (DAMPE-02), whose channel number will be up to five hundreds of thousands, and will generate massive raw data during the data acquisition, must have a fast and efficient data compression solution on orbit. In this paper, a real-time data compression algorithm is proposed, which is based on the FPGA (Field-programmable Gate Array). A pipeline structure in the FPGA is designed for the algorithm, to make the process more parallel and efficiently accelerate the running speed of the algorithm. And finally, the compression velocity, up to 38.4 million channel per second, is achieved. The design idea can provide a reference for the track detector of DAMPE-02.

**Key words** instrumentation: detector, dark matter, computing method: data compression