密 级：□公开 □内部 🗹机密 □绝密

|  |  |
| --- | --- |
| 发放号 |  |
| 受控状态 |  |

B4-DIM-3003

**影像子系统与软件子系统接口协议**

V1.0

|  |  |  |
| --- | --- | --- |
| 签名 | | 日期 |
| 编 制 |  |  |
| 审 核 |  |  |
|  |  |
|  |  |
|  |  |
| 批 准 |  |  |
| 生效日期 |  | |

**修订记录页**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 修订人 | 变更日期 | 变更内容描述 | 备注 |
| V1.0 | 范策 |  | 初始发放 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[1. 摘要 4](#_Toc4758633)

[2. CT模块 4](#_Toc4758634)

[2.1设计输入 4](#_Toc4758635)

[2.2 设计内容 4](#_Toc4758636)

[2.2.1 探测器通讯接口 4](#_Toc4758637)

[2.2.2 球管通讯接口 6](#_Toc4758638)

[2.2.3 触发板通讯接口 7](#_Toc4758639)

[2.2.4 图像重建算法接口 17](#_Toc4758640)

[2.2.5 软件扫描Configuration文件结构 27](#_Toc4758641)

[2.2.6 数据采集流程 28](#_Toc4758642)

[3. PET/SPECT模块 29](#_Toc4758643)

[3.1 设计输入 29](#_Toc4758644)

[3.1.1 设计需求 29](#_Toc4758645)

[3.1.2 术语及缩略语 29](#_Toc4758646)

[3.1.3 设计原理 30](#_Toc4758647)

[3.2 设计内容 30](#_Toc4758648)

[3.2.1 PET工作流程 30](#_Toc4758649)

[3.2.1.1 初始化流程 32](#_Toc4758650)

[3.2.1.2 PET采集流程 34](#_Toc4758651)

[3.2.2 SPECT采集流程 36](#_Toc4758652)

[3.2.2.1 SPECT采集流程 37](#_Toc4758653)

[3.2.3 状态监控 38](#_Toc4758654)

[3.2.4 控制接口定义 39](#_Toc4758655)

[3.2.4.1 物理接口 39](#_Toc4758656)

[3.2.4.2 通讯协议 39](#_Toc4758657)

[3.2.4.3 MMP地址说明 41](#_Toc4758658)

[3.2.4.4 White Rabbit MMP功能说明 41](#_Toc4758659)

[3.2.4.5 CM MMP功能说明 42](#_Toc4758660)

[3.2.5 数据接口定义 47](#_Toc4758661)

[3.2.5.1 物理接口 47](#_Toc4758662)

[3.2.5.2 通讯协议 47](#_Toc4758663)

[3.2.5.3 数据包格式 47](#_Toc4758664)

[3.2.6 数据存储格式 49](#_Toc4758665)

[3.2.6.1表模式数据格式 49](#_Toc4758666)

[3.2.6.2重建结果格式 49](#_Toc4758667)

[3.2.7 PET算法与软件接口 49](#_Toc4758668)

[3.2.7.1 解析重建算法 50](#_Toc4758669)

[3.2.7.2 迭代重建算法 52](#_Toc4758670)

[3.2.7.3 实时显示算法（图像刷新时间待定） 55](#_Toc4758671)

[3.2.7.4 质控算法 56](#_Toc4758672)

[3.2.7.5 迭代重建衰减校正说明 57](#_Toc4758673)

[3.2.7.6 配准参数文件 58](#_Toc4758674)

[3.2.7.7 定量校正系数更新 58](#_Toc4758675)

[3.2.7.8 死时间校正 58](#_Toc4758676)

[3.2.8 SPECT算法与软件接口 58](#_Toc4758677)

[3.2.8.1 迭代重建算法 59](#_Toc4758678)

[3.2.8.2 实时显示算法 61](#_Toc4758679)

[3.2.8.3 质控相关算法 63](#_Toc4758680)

[4. 备注 64](#_Toc4758681)

[7.1 示例代码 64](#_Toc4758682)

[4.2 电子学与算法提供文件汇总 65](#_Toc4758683)

# 摘要

本文档为B3 PET/SPECT模块与软件采集系统的接口定义，用于软件开发人员进行PET/SPECT数据采集软件开发的参考。

# PET模块

# 设计输入

## 3.1.1 设计需求

为明确PET/SPECT模块软件采集的流程，特制定软件接口协议说明。PET/SPECT电子学与软件接口分为控制接口与数据接口两部分，其中控制接口用于规范上层软件对底层硬件运行控制与状态监控等相关过程，数据接口用于规范上层软件与底层硬件间的数据打包及解析等相关过程，为软件与电子学的协同开发提供依据。

## 3.1.2 术语及缩略语

TCP/IP：网络通讯协议

UDP：用户数据报协议

TCP：传输控制协议

White Rabbit Switch：大白兔同步交换机

10G Switch: 采集数据符合交换机

FEB:前端板

ADB: FPGA内部根据ASIC接口进行控制的模块

DPB/Cute\_WR：数据采集同步子板

CM: 符合单元模块（一个CM包括一块DPB和两块FEB）

MMP：Memory Mapped Register 地址映射寄存器

APC：算法服务器

Fiber：光纤

SFP：光信号接口器件

UTP：双绞线

FiberR/T Transceiver：光纤收发器

Gigabit Router：千兆网路由器

COM5402：TCP/IP协议栈模块

ListMode Data: 表模式数据

Projection: 投影

MLEM: 极大似然期望最大化

System Matrix: 系统传输矩阵

## 3.1.3 设计原理

具体工作原理如下，算法服务器软件共开设1个UDP端口和9个TCP端口，其中：

（1）Cute\_WR通过UDP端口将采集到的SE数据发送到10G Switch交换机；10G Switch交换机将10个CM发送来的UDP数据包解析，提取有效数据后，存储到各自的RX\_FIFO中；Timer模块负责物理延迟恢复，即读出每个RX\_FIFO中的gamma事件数据，并按照事件中包含的物理时间戳进行物理延迟后（模拟gamma事件的真实到来时刻，单纯增加物理延迟，所有通道的相对延迟为0）输出给符合逻辑；符合逻辑进行符合事件挑选，并存储到COM5402 对应UDP数据通道的TX FIFO中；COM5402将UDP TX FIFO中的符合数据按照UDP协议打包后发送给服务器数据接收用UDP(UDP\_D)端口。

（2）APC服务器设置10个TCP控制端口TCP0\_I-TCP9\_I，分别用来与10个CM模块的Cute\_WR TCP端口进行通信，形成CM模块的控制通道。服务器建立TCP连接，发送控制数据包到相应的Cute\_WR TCP Server；Cute\_WR接收到控制包，转换成转换成对控制寄存器以及状态寄存器的读写（MMP），进而完成CM模块偏压设置、逻辑工作设置、偏压温度监控、逻辑状态监控等控制功能。

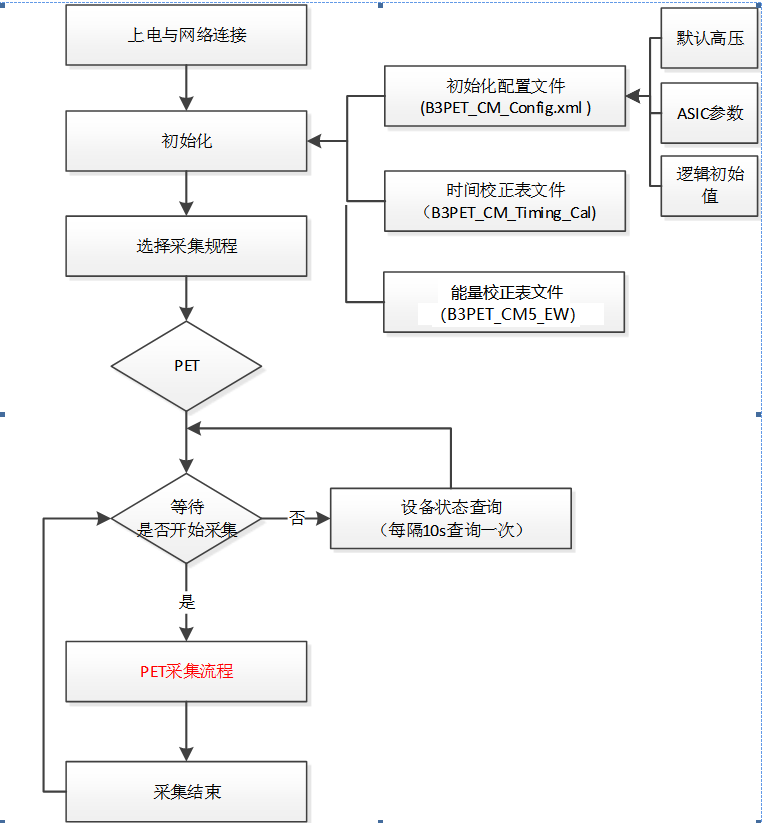
（3）APC服务器设置1个UDP端口UDP11\_I，完成对10G Switch交换机逻辑模块控制，以及MMP寄存器的读写，进而完成符合模块逻辑参数的设计。

1. 预留APC服务器到10G Switch交换机的网络通讯接口，用于10G Switch交换机的设置等。

# 3.2 设计内容

## 3.2.1 PET工作流程

PET工作流程图如下：



PET采集流程图如下：



### 3.2.1.1 初始化流程

（1）软件启动后，连接电子学，包括10G Switch 交换机、WR交换机和10个CM建立连接；

（2）软件向UDP\_D发送数据段内容全为零、长度为512的UDP数据包，用于底层硬件获取服务器Mac地址；

（3）设置服务器端口号WR\_REG\_PORT及IP地址WR\_REG\_IP\_L和WR\_REG\_IP\_H（算法服务器地址和端口）；

（4）依次查询10个CM的同步状态：

依次读取10个CM的DPB\_REG\_SYNC\_FLAG寄存器，当所有CM的同步状态都为1时，进行下一步操作

（5）依次对10个CM电子学进行初始化操作：

8个CM的初始化配置文件为B3PET\_CM0\_Config.xml ~ B3PET\_CM9\_Config.xml，软件解析配置文件，并按照初始化要求，顺序下发MMP寄存器地址与数据，完成系统初始化操作。示例代码见备注7.1,命令包示例见B3PET\_CM(CFG)文件。

具体参数下发顺序如下所示：

1. 设置DPB参数：
2. 地址0x0003 写服务器1段端口号
3. 地址0x0004 写服务器1段网卡的IP地址（高16bit）
4. 地址0x0005 写服务器1段网卡的IP地址（低16bit）
5. 依次初始化查表相关设置
6. 地址0x0104 写0x0000；
7. 地址0x0100 写0x0009；
8. 地址0x0103 写0x0002；
9. 地址0x0100 写0x0008；
10. 等待100ms；
11. 地址0x0104 写0x0002；
12. 地址0x0000 写 0x0030；
13. 依次设置ADB0、ADB1、ADB2、ADB3、ADB4、ADB5、ADB6、ADB7、ADB8、ADB9的高压；
14. ASIC参数下发
15. 对ASIC进行复位，地址0x0400 写 0x0010
16. 按照地址将ASIC0单通道微调<32-63><ASICCALENSEL1SEL0RBSEL><ASIC慢通道选通宽度调节><ASIC定时输出宽度调节><ASIC全局阈值> ASIC单通道微调<31-0>顺序下发；
17. 将ASIC0的XBaseline YBaseline EBaseline依次下发；
18. 对ASIC0进行参数下发使能，地址0x0400 写 0x0060
19. 按照地址将ASIC1单通道微调<32-63><ASICCALENSEL1SEL0RBSEL><ASIC慢通道选通宽度调节><ASIC定时输出宽度调节><ASIC全局阈值> ASIC单通道微调<31-0>顺序下发；
20. 将ASIC1的XBaseline YBaseline EBaseline依次下发；
21. 对ASIC1进行参数下发使能，地址0x0400 写 0x00A0
22. 按照地址将ASIC2单通道微调<32-63><ASICCALENSEL1SEL0RBSEL><ASIC慢通道选通宽度调节><ASIC定时输出宽度调节><ASIC全局阈值> ASIC单通道微调<31-0>顺序下发；
23. 将ASIC2的XBaseline YBaseline EBaseline依次下发；
24. 对ASIC2进行参数下发使能，地址0x0800 写 0x0060
25. 按照地址将ASIC3单通道微调<32-63><ASICCALENSEL1SEL0RBSEL><ASIC慢通道选通宽度调节><ASIC定时输出宽度调节><ASIC全局阈值> ASIC单通道微调<31-0>顺序下发；
26. 将ASIC3的XBaseline YBaseline EBaseline依次下发；
27. 对ASIC3进行参数下发使能，地址0x0800 写 0x00A0
28. 按照地址将ASIC4单通道微调<32-63><ASICCALENSEL1SEL0RBSEL><ASIC慢通道选通宽度调节><ASIC定时输出宽度调节><ASIC全局阈值> ASIC单通道微调<31-0>顺序下发；
29. 将ASIC4的XBaseline YBaseline EBaseline依次下发；
30. 对ASIC4进行参数下发使能，地址0x0C00 写 0x0060
31. 按照地址将ASIC5单通道微调<32-63><ASICCALENSEL1SEL0RBSEL><ASIC慢通道选通宽度调节><ASIC定时输出宽度调节><ASIC全局阈值> ASIC单通道微调<31-0>顺序下发；
32. 将ASIC5的XBaseline YBaseline EBaseline依次下发；
33. 对ASIC5进行参数下发使能，地址0x0C00 写 0x00A0
34. 按照地址将ASIC6单通道微调<32-63><ASICCALENSEL1SEL0RBSEL><ASIC慢通道选通宽度调节><ASIC定时输出宽度调节><ASIC全局阈值> ASIC单通道微调<31-0>顺序下发；
35. 将ASIC6的XBaseline YBaseline EBaseline依次下发；
36. 对ASIC6进行参数下发使能，地址0x1000 写 0x0060
37. 按照地址将ASIC7单通道微调<32-63><ASICCALENSEL1SEL0RBSEL><ASIC慢通道选通宽度调节><ASIC定时输出宽度调节><ASIC全局阈值> ASIC单通道微调<31-0>顺序下发；
38. 将ASIC7的XBaseline YBaseline EBaseline依次下发；
39. 对ASIC7进行参数下发使能，地址0x1000 写 0x00A0
40. 按照地址将ASIC8单通道微调<32-63><ASICCALENSEL1SEL0RBSEL><ASIC慢通道选通宽度调节><ASIC定时输出宽度调节><ASIC全局阈值> ASIC单通道微调<31-0>顺序下发；
41. 将ASIC8的XBaseline YBaseline EBaseline依次下发；
42. 对ASIC8进行参数下发使能，地址0x1400 写 0x0060
43. 按照地址将ASIC9单通道微调<32-63><ASICCALENSEL1SEL0RBSEL><ASIC慢通道选通宽度调节><ASIC定时输出宽度调节><ASIC全局阈值> ASIC单通道微调<31-0>顺序下发；
44. 将ASIC9的XBaseline YBaseline EBaseline依次下发；
45. 对ASIC9进行参数下发使能，地址0x1400 写 0x00A0

读取10个CM的时间校正表文件B3PET\_CM0\_Timing\_Cal~B3PET\_CM9\_Timing\_Cal（时间校正表文件为纯二进制文件），然后按照DPB时间校正表寄存器地址DPB\_REG\_TIME\_ST依次写入对应地址。

### 3.2.1.2 PET采集流程

1. 核素信息设置：核素ID、能窗：

依次对10个CM进行核素选择及能窗文件下载，以CM0为例

1. 将核素ID下发到对应的控制寄存器中：配置寄存器DPB\_REG\_CTRL（高8位核素ID，低8位0x30）;
2. 将能窗上下阈下发到能窗上下阈设置寄存器：用户设置下阈写入寄存器DPB\_REG\_EW\_D，用户设置上阈写入寄存器DPB\_REG\_EW\_U;
3. 读取能窗文件B3PET\_CM0\_EW（能窗文件为纯二进制文件，包括斜率和截距两部分），然后按照DPB能窗寄存器DPB\_REG\_EW\_ST地址依次写入对应地址；
4. 符合时间窗设置

将符合时间窗下发到符合时间窗寄存器：用户设置符合时间窗写入寄存器WR\_REG\_CO\_TW（暂定192, 符合时间窗6ns）；

符合时间窗实际下发值= 符合时间窗（单位ns） \* 256 / 4 / 2；

1. 开始进行数据采集；
2. 采集模式设置: 设置10G Switch交换机控制寄存器WR\_REG\_CTRL使FIFO清零；
3. 设置10G Switch交换机控制寄存器WR\_REG\_CTRL的SE输出禁止；
4. 设置10G Switch交换机控制寄存器WR\_REG\_CTRL的DE输出使能；
5. 设置10G Switch交换机控制寄存器WR\_REG\_CTRL的DE FALSE数据输出使能；
6. 设置10G Switch交换机控制寄存器WR\_REG\_CTRL的采集时刻数据输出使能；
7. 软件开始向UDP\_D端口发送数据包
8. 软件开始接收UDP\_D端口发出的UDP数据包；
9. 数据由数据起始包起始，紧跟N个8字节数据包，依次循环。
10. 软件接受到的两次数据起始包之间的数据的长度如为8字节的整数倍，则该段数据为有效数据，软件存储该数据并根据8字节中的type标志位累加事件数（不含数据起始包）；否则丢弃该段数据。
11. 根据重建选项，选择是否进行实时重建;
12. 如果选择了实时重建，参照3.7.3 实时显示算法进行。
13. 单床位结束采集：

a、判断是否到达采集结束条件；

采集结束条件判断：软件支持以采集时间（精度为s）或事件数结束采集。

b、首先设置10G Switch交换机控制寄存器WR\_REG\_CTRL的DE、数据时刻包输出关闭；

c、根据重建选项，执行相应的图像重建算法，详见3.7.1解析重建算法接口和3.7.2迭代重建算法接口。

1. 完成所有床位采集之后，采集流程结束。

**注：质控采集流程暂未确定，将在质控需求中详述。**

## 3.2.3 状态监控

温度监控：实时查询温度寄存器ADB\_REG\_TEMP，正常温度范围为20℃~35℃，若超出此范围，则显示相对应的CM和FEB温度异常，并提示用户停止采集。

偏压监控：实时查询偏压寄存器ADB\_REG\_HV\_MONI，正常偏压范围为25V~30V，若超出此范围，则显示相对应的CM和FEB偏压异常，并提示用户停止采集。

连接状态：通过温度监控与偏压监控控制包状态判断网络连接状态；

增加电子学日志，要求如下：

1、10个CM各输出一个温度，取中间ADC的温度；

2、10个CM各输出一个计数率信息；

5、10个CM各输出三个偏压信息；

4、每隔1分钟输出一次；

6、单独形成记录文件，每行信息包括当前时间和10个CM盒的温度信息、计数率信息和偏压信息；

Time CM0Temp CM0CoutRate CM0\_Bias0 CM0\_Bias1 CM0\_Bias2 ……CM9Temp CM9CoutRate CM9\_Bias0 CM9\_Bias1 CM9\_Bias2；

7、数据采集过程中，每隔1分钟输出总计数率（PET为符合计数率），存在采集过程日志中。

## 3.2.4 控制接口定义

为实现软件对电子学硬件设置以及逻辑参数设置的完全控制，共设定了11个TCP通信链路，每个通路的物理接口、通讯协议及通信流程完全一致，仅在MMP地址映射寄存器的设置和初始化流程上存在区别。

### 3.2.4.1 物理接口

光纤，全双工通信

### 3.2.4.2 通讯协议

TCP/IP协议，其中控制命令传输使用TCP协议。

在软件服务器端与硬件设备端建立的TCP通信链路上，发送与接收的控制数据包格式如图3-1所示：



图3-1 TCP控制包格式

struct commPackage

{

Byte flag; /\* 通讯数据包标志 \*/

Byte cont; /\* 通讯数据码值\*/

Byte Serial\_ID; /\* 序列号 \*/

Byte Length; /\* 当前数据包携带的数据字节数 \*/

Byte Reg\_Add\_L; /\*寄存器地址低8位 \*/

Byte Reg\_Add\_H; /\*寄存器地址高8位 \*/

};

|  |  |  |
| --- | --- | --- |
| 通讯数据包标志(flag) | 码值 | 含义 |
| PKG\_CMD | 0xFF | 命令包头 |
| 通讯码值(cont) | 码值 | 含义 |
| PKG\_CMD\_WR | 0x80 | 写入参数 |
| PKG\_CMD\_RD | 0xC0 | 读取状态 |
| PKG\_CMD\_WR\_ACK | 0x88 | 写入应答包 |
| PKG\_CMD\_RD\_ACK | 0xC8 | 读取应答包 |
| PKG\_CMD\_ERROR | 0xE8 | 错误状态反馈包 |
| 序列号(Serial\_ID) | 码值 | 含义 |
| PKG\_CMD\_SID | 0x01-0x7F | 序列号，初始值为1，每次发送自动加1，作为包序号，应答包中数值与下发值一致 |
| **数据包长度（Length）** | 码值 | **含义** |
| PKG\_CMD\_LENGTH | 0x00-0x7F | 数据包长度，单位字节 |
| **寄存器地址（Reg\_Add）** | 码值 | **含义** |
|  |  |  |

具体通信流程：

（1）写流程：软件服务器端发送写控制数据包到底层硬件，其中CMD字节为0x80，序列号初始值为1（每次发送自动加1，作为包序号），数据长度为2（根据命令内容确定），寄存器地址为要读写寄存器的地址（16bit，2字节），数据值为写入寄存器的值（16bit，2字节），发送之后工作于阻塞模式等待回复或超时；底层硬件接收到TCP包解析成寄存器地址和值，将相应寄存器值更改，并同时发送控制数据包答复包，包内容与接收到的控制包一致，除CMD改为0x88，数据值为寄存器更改后的值；软件服务器等待底层硬件发送的答复包，如在超时时间（2s）内收到答复包，且确认包内容正确，则此次写操作完成，如果包内容与发送的包内容不一致或者超时，则重新发送此控制包(重复次数最多5次)，否则，此次写操作失败；

（2）读流程：软件服务器端发送写控制数据包到底层硬件，其中CMD字节为0xC0，序列号初始值为1（每次发送自动加1，作为包序号），数据长度为2（根据命令内容确定），寄存器地址为要读写寄存器的地址（16bit，2字节），数据值为0（16bit，2字节），发送之后工作于阻塞模式等待回 复或超时；底层硬件接收到TCP包解析成寄存器地址，读回相应地址寄存器的值后，发送控制数据包答复包，包内容与接收到的控制包一致, 除CMD改为0xC8，数据值为寄存器值；软件服务器等待底层硬件发送的答复包，如在超时时间（2s）内收到答复包，且确认包内容正确，则此次读操作完成，如果包内容与发送的包内容不一致或者超时，则重新发送此控制包(重复次数最多5次)，否则，此次读操作失败。

### 3.2.4.3 MMP地址说明

MMP地址分成两部分进行描述，White Rabbit (WR) 与CM，八个CM具有完全相同的地址分配。

### 3.2.4.4 White Rabbit MMP功能说明

White Rabbit寄存器地址映射表如表3-1所示。

表3-1 White Rabbit 寄存器列表

|  |  |  |  |
| --- | --- | --- | --- |
| 寄存器代码 | 偏移地址 | w/r | Register Name |
| WR\_REG\_CTRL | 0x0000 | w/r | 控制寄存器 |
| WR\_REG\_CO\_TW | 0x0001 | w/r | 符合时间窗 |
| WR\_REG\_IP\_L | 0x0002 | w/r | 服务器IP低位 |
| WR\_REG\_IP\_H | 0x0003 | w/r | 服务器IP高位 |
| WR\_REG\_PORT | 0x0004 | w/r | 服务器端口号 |
| WR\_REG\_STATUS | 0x0010 | r | 状态寄存器 |
| WR\_REG\_CM0\_CR\_L | 0x0013 | r | CM0 SE计数率低位 |
| WR\_REG\_CM0\_CR\_H | 0x0014 | r | CM0 SE计数率高位 |
| WR\_REG\_CM1\_CR\_L | 0x0015 | r | CM1 SE计数率低位 |
| WR\_REG\_CM1\_CR\_H | 0x0016 | r | CM1 SE计数率高位 |
| WR\_REG\_CM2\_CR\_L | 0x0017 | r | CM2 SE计数率低位 |
| WR\_REG\_CM2\_CR\_H | 0x0018 | r | CM2 SE计数率高位 |
| WR\_REG\_CM3\_CR\_L | 0x0019 | r | CM3 SE计数率低位 |
| WR\_REG\_CM3\_CR\_H | 0x001A | r | CM3 SE计数率高位 |
| WR\_REG\_CM4\_CR\_L | 0x001B | r | CM4 SE计数率低位 |
| WR\_REG\_CM4\_CR\_H | 0x001C | r | CM4 SE计数率高位 |
| WR\_REG\_CM5\_CR\_L | 0x001D | r | CM5 SE计数率低位 |
| WR\_REG\_CM5\_CR\_H | 0x001E | r | CM5 SE计数率高位 |
| WR\_REG\_CM6\_CR\_L | 0x001F | r | CM6 SE计数率低位 |
| WR\_REG\_CM6\_CR\_H | 0x0020 | r | CM6 SE计数率高位 |
| WR\_REG\_CM7\_CR\_L | 0x0021 | r | CM7 SE计数率低位 |
| WR\_REG\_CM7\_CR\_H | 0x0022 | r | CM7 SE计数率高位 |
| WR\_REG\_CM8\_CR\_L | 0x0023 | r | CM8 SE计数率低位 |
| WR\_REG\_CM8\_CR\_H | 0x0024 | r | CM8 SE计数率高位 |
| WR\_REG\_CM9\_CR\_L | 0x0025 | r | CM9 SE计数率低位 |
| WR\_REG\_CM9\_CR\_H | 0x0026 | r | CM9 SE计数率高位 |
| WR\_REG\_DIAG\_CR\_L | 0x0027 | r | DIAG计数率低位 |
| WR\_REG\_DIAG\_CR\_H | 0x0028 | r | DIAG计数率高位 |
| WR\_REG\_SE\_CR\_L | 0x0029 | r | SE计数率低位 |
| WR\_REG\_SE\_CR\_H | 0x002A | r | SE计数率高位 |
| WR\_REG\_DE\_CR\_L | 0x002B | r | 符合计数率低位 |
| WR\_REG\_DE\_CR\_H | 0x002C | r | 符合计数率高位 |
| WR\_REG\_DEFALSE\_CR\_L | 0x002D | r | 延迟符合计数率低位 |
| WR\_REG\_DEFALSE\_CR\_H | 0x002E | r | 延迟符合计数率高位 |
| WR\_REG\_SW\_REV | 0x002F | r | 主FPGA固件版本 |
| WR\_REG\_WR\_REV | 0x0030 | r | 从FPGA固件版本 |

1. 控制寄存器：

地址0x0000，可读可写，负责大白兔路由器符合逻辑模块功能控制

[15:9]、[7:6]：保留，默认值0；

[8]：置1完成符合逻辑FIFO复位，置0无操作，复位完成，自动清0，默认0；

[5]：置1 SE切换为查表前数据，置0 SE切换为查表后数据输出，默认0。

[4]：置1 采集时刻数据输出使能，置0 采集时刻数据输出关闭，默认0。

[3]：置1 DE FLASE数据输出使能，置0 DE FALSE数据输出关闭，默认0。

[2]：置1 DE数据输出使能，置0 DE数据输出关闭，默认0。

[1]：置1 SE数据输出使能，置0 SE数据输出关闭，默认0。

[0]：置1测试数据输出使能，置0测试数据输出关闭，默认0。

1. 符合时间窗：

地址0x0001，可读可写，符合时间窗（单位：细分时间格，等于4/256 ns）

[15:0]：符合时间窗

1. 服务器IP低16位：

地址0x0002，可读可写，服务器IP低16bit

[15:0]：IP L16

1. 服务器IP高16位：

地址0x0003，可读可写，服务器IP高16bit

[15:0]：IP H16

1. 服务器端口号：

地址0x0003，可读可写，服务器UDP接收端口号

[15:0]：PORT

1. 状态寄存器：

地址0x0010，只读，负责大白兔路由器符合逻辑功能模块状态监控

[15:1]：保留，默认值0；

[0]：1代表PLL锁定，0代表PLL失锁。

1. 底层计数率统计信息：

地址0x0013-0x0030，只读，单位：事件数/s

### 3.2.4.5 CM MMP功能说明

CM功能模块的具体功能主要是由一块DPB与二块FEB板子功能单元实现的。

DPB MMP寄存器地址映射表如表3-2所示：

表3-2 DPB寄存器列表

|  |  |  |  |
| --- | --- | --- | --- |
| 寄存器代码 | 偏移地址 | w/r | Register Name |
| DPB\_REG\_CTRL | 0x0000 | w/r | 控制寄存器 |
| DPB\_REG\_EW\_U | 0x0001 | w/r | 能窗上阈设置 |
| DPB\_REG\_EW\_D | 0x0002 | w/r | 能窗下阈设置 |
| DPB\_REG\_FIRMWAREVERSION | 0x0018 | r | DPB固件版本号 |
| DPB\_REG\_SYNC\_FLAG | 0x001A | r | 同步标志位 |
| DPB\_REG\_EW\_ST [0:10239] | 0x8000-0xA7FF | w/r | 能窗查找表 |
| DPB\_REG\_TIME\_ST  [0:10239] | 0xB000-0xD7FF | w/r | 时间校正表 |

1. 控制寄存器：

地址0x0000，可读可写，负责DPB逻辑模块功能控制

[15:13]：核素ID，对应位置查找表入口地址，PET核素均为0x00,其他待定；

[12:9]：保留，默认值0；

[8]：置1 打开能量表

[5]：置1 打开能窗

[4]：置1 查表使能

[7:1]：保留，默认值0；

[0]：置1 ADB FIFO清空

1. 能窗上阈寄存器：

地址0x0001，可读可写，负责DPB能窗上阈功能控制

1. 能窗下阈寄存器：

地址0x0002，可读可写，负责DPB能窗下阈功能控制

1. DPB固件版本号寄存器

地址0x0018只读。负责记录DPB的固件版本号

1. 同步标志位寄存器

地址0x001A只读。负责DPB同步状态，即CM同步状态

[15:1]保留；

[0] 1代表同步成功，0代表同步失败。

1. 能窗查找表：

地址0x8000-0xA7FF，可读可写，对应晶体能窗查找表

其中0x8000-0x83FF BK0晶体能窗查找表

0x8400-0x87FF BK5晶体能窗查找表

0x8800-0x8BFF BK1晶体能窗查找表

0x8C00-0x8FFF BK6晶体能窗查找表

0x9000-0x93FF BK2晶体能窗查找表

0x9400-0x97FF BK7晶体能窗查找表

0x9800-0x9BFF BK3晶体能窗查找表

0x9C00-0x9FFF BK8晶体能窗查找表

0xA000-0xA3FF BK4晶体能窗查找表

0xA400-0xA7FF BK9晶体能窗查找表

1. 时间校正表：

地址0xB000-0xD7FF，可读可写，对应晶体时间校正表

其中0xB000-0XB3FF BK0晶体时间校正表

0xB400-0xB7FF BK5晶体时间校正表

0xB800-0xBBFF BK1晶体时间校正表

0xBC00-0xBFFF BK6晶体时间校正表

0xC000-0xC3FF BK2晶体时间校正表

0xC400-0xC7FF BK7晶体时间校正表

0xC800-0xCBFF BK3晶体时间校正表

0xCC00-0xCFFF BK8晶体时间校正表

0xD000-0xD3FF BK4晶体时间校正表

0xD400-0xD7FF BK9晶体时间校正表

ADB寄存器对应的MMP地址即一级地址+偏移地址,一级地址如表3-3所示，偏移地址如3-4所示。

表3-3 ADB寄存器一级地址

|  |  |  |
| --- | --- | --- |
| 宏代码 | 单元名称 | 一级地址 |
| SEG\_ADB0 | ADB0 | 0x0400 |
| SEG\_ADB1 | ADB1 | 0x0800 |
| SEG\_ADB2 | ADB2 | 0x0C00 |
| SEG\_ADB2 | ADB3 | 0x1000 |
| SEG\_ADB2 | ADB4 | 0x1400 |

表3-4 ADB寄存器偏移地址

|  |  |  |  |
| --- | --- | --- | --- |
| 寄存器代码 | 偏移地址 | w/r | Register Name |
| ADB\_REG\_CTRL | 0x0000 | w/r | 控制寄存器 |
| ADB\_REG\_DOUT\_EN | 0x0001 | w/r | 数据输出模式 |
| ADB\_REG\_BKID | 0x0005 | w/r | 位置编号 |
| ADB\_REG\_HV0\_SET | 0x0006 | w/r | 偏压0设置寄存器 |
| ADB\_REG\_HV1\_SET | 0x000E | w/r | 偏压1设置寄存器 |
| ADB\_REG\_ASIC0\_THR\_L  [32:63] | 0x0100-0x011F | w | ASIC0 32到63通道微调阈值 |
| ADB\_REG\_ASIC0\_CALEN | 0x0120 | w | ASIC0 CALEN |
| ADB\_REG\_ASIC0\_SEN | 0x0121 | w | ASIC0慢通道选通调节 |
| ADB\_REG\_ASIC0\_T\_WD | 0x0122 | w | ASIC0定时输出宽度 |
| ADB\_REG\_ASIC0\_THR\_G | 0x0123 | w | ASIC0全局阈值 |
| ADB\_REG\_ASIC0\_THR\_L  [31:0] | 0x0124-0x0143 | w | ASIC0 31到0通道单通道微调阈值 |
| ADB\_REG\_ASIC1\_THR\_L  [32:63] | 0x0200-0x021F | w | ASIC5 32到63通道微调阈值 |
| ADB\_REG\_ASIC1\_CALEN | 0x0220 | w | ASIC5 CALEN |
| ADB\_REG\_ASIC1\_SEN | 0x0221 | w | ASIC5慢通道选通调节 |
| ADB\_REG\_ASIC1\_T\_WD | 0x0222 | w | ASIC5定时输出宽度 |
| ADB\_REG\_ASIC1\_THR\_G | 0x0223 | w | ASIC5全局阈值 |
| ADB\_REG\_ASIC1\_THR\_L  [31:0] | 0x0224-0x0243 | w | ASIC5 31到0通道单通道微调阈值 |
| ADB\_REG\_TEMP | 0x0301 | r | 温度监控寄存器 |
| ADB\_REG\_ASIC1\_VB | 0x0302 | r | ASIC5 VB Monitor |
| ADB\_REG\_ASIC0\_VB | 0x0303 | r | ASIC0 VB Monitor |
| ADB\_REG\_HV\_MONI | 0x0304 | r | 偏压监控寄存器 |
| WR\_REG\_BK0\_CR\_L | 0x0305 | r | BK0 SE计数率低位 |
| WR\_REG\_BK0\_CR\_H | 0x0306 | r | BK0 SE计数率高位 |
| WR\_REG\_BK1\_CR\_L | 0x0307 | r | BK5 SE计数率低位 |
| WR\_REG\_BK1\_CR\_H | 0x0308 | r | BK5 SE计数率高位 |
| WR\_REG\_BK0T\_CR\_L | 0x0309 | r | BK0 T计数率低位 |
| WR\_REG\_BK0T\_CR\_H | 0x030A | r | BK0 T计数率高位 |
| WR\_REG\_BK1T\_CR\_L | 0x030B | r | BK5 T计数率低位 |
| WR\_REG\_BK1T\_CR\_H | 0x030C | r | BK5 T计数率高位 |

1. 控制寄存器：

地址0x0000，可读可写，负责ADB逻辑模块功能控制

[15:8]：保留，默认值0；

[7]：置1 Asic 5选中，置0 Asic 5不选中，默认0；

[6]：置1 Asic 0选中，置0 Asic 0不选中，默认0；

[5]：置1完成ASIC参数下发操作，置0无操作，参数下发完成，自动清0，默认0；

[4]：置1完成ASIC复位操作，置0无操作，ASIC复位完成，自动清0，默认0；

[3]：保留

[2]：置1完成逻辑复位操作，置0无操作，逻辑复位完成，自动清0，默认0；

[1]：置1完成ADC及数据接收复位操作，置0无操作，复位完成，自动清0，默认0；

[0]：置1完成时钟复位操作，置0无操作，时钟复位完成，自动清0，默认0；

1. 数据输出控制寄存器：

地址0x0001，可读可写，负责输出使能

[15:8]：保留，默认值0；

[6]：置1 BK5 E触发波形输出使能，置0 BK5 E触发波形输出关闭，默认0。

[5]：置1 BK5 自触发波形输出使能，置0 BK5 自触发波形输出关闭，默认0。

[4]：置1 BK5 SE输出使能，置0 BK5 SE数据输出关闭，默认0。

[2]：置1 BK0 E触发波形输出使能，置0 BK0 E触发波形输出关闭，默认0。

[1]：置1 BK0 自触发波形输出使能，置0 BK0 自触发波形输出关闭，默认0。

[0]：置1 BK0 SE输出使能，置0 BK0 SE数据输出关闭，默认0。

1. 位置编号：

地址0x0005，可读可写，BK位置编号（其中BKID 0bit 由底层硬件自动确定）

[15:7]：保留，默认值0；

[6:3]：CMID，默认0；

[2:1]：BKID，默认0；

[0]:低位默认0。

1. 偏压设置寄存器：

地址0x0006，为HV0可读可写，负责偏压控制

地址0x000E，为HV1可读可写，负责偏压控制

[15:12]：保留，默认值0；

[11:0]：偏压设置，偏压值与设置值函数关系：

if (真实偏压值>= 20.900)

设置值=（真实偏压值-20.900+偏压纠正值）\*2111.861483);

else

设置值= 0;

举例说明：hv = floor((HVSetting - 20.900+m\_dHVMonitor\_sub)\*2111.861483);

1. ASIC0 通道32-63单通道微调阈值：

地址0x0100-0x011F，可读可写，负责设置ASIC0通道**32-63**微调阈值

[15:8]：保留，默认值0；

[7:0]：默认0x00，通过ASIC阈值标定获得

1. ASIC0 CALEN：

地址0x0120，可读可写，负责设置ASIC0工作模式

[15:8]：保留，默认值0；

[7:0]：默认0x0F，不可变

1. ASIC0慢通道选通调节：

地址0x0121，可读可写，负责设置ASIC0慢通道选通调节

[15:8]：保留，默认值0；

[7:0]：默认0x0B，不可变

1. ASIC0定时输出宽度：

地址0x0122，可读可写，负责设置ASIC0的定时输出宽度

[15:8]：保留，默认值0；

[7:0]：默认0x07，不可变

1. ASIC0全局阈值：

地址0x0123，可读可写，负责设置ASIC0的全局阈值

[15:8]：保留，默认值0；

[7:0]：默认0x00，通过ASIC阈值标定获得；

1. ASIC0 通道31-0单通道微调阈值：

地址0x0124-0x0143，可读可写，负责设置ASIC0通道**31-0**微调阈值

[15:8]：保留，默认值0；

[7:0]：默认0x00，通过ASIC阈值标定获得

1. ASIC1 通道32-63单通道微调阈值：

地址0x0200-0x021F，可读可写，负责设置ASIC1通道**32-63**单通道微调阈值

[15:8]：保留，默认值0；

[7:0]：默认0x00，通过ASIC阈值标定获得

1. ASIC1 CALEN：

地址0x0220，可读可写，负责设置ASIC1工作模式

[15:8]：保留，默认值0；

[7:0]：默认0x0F，不可变

1. ASIC1慢通道选通调节：

地址0x0221，可读可写，负责设置ASIC1慢通道选通调节

[15:8]：保留，默认值0；

[7:0]：默认0x0B，不可变

1. ASIC1定时输出宽度：

地址0x0222，可读可写，负责设置ASIC1的定时输出宽度

[15:8]：保留，默认值0；

[7:0]：默认0x07，不可变

1. ASIC1全局阈值：

地址0x0223，可读可写，负责设置ASIC1的全局阈值

[15:8]：保留，默认值0；

[7:0]：默认0x00，通过ASIC阈值标定获得；

1. ASIC1 通道31-0单通道微调阈值：

地址0x0224-0x0243，可读可写，负责设置ASIC1通道**31-0**微调阈值

[15:8]：保留，默认值0；

[7:0]：默认0x00，通过ASIC阈值标定获得

1. 状态寄存器：

地址0x0300，只读，读取状态寄存器

[15:2]：保留，默认值0；

[1]：1表示ADC测试数据接收正确，0表示ADC测试数据接收错误；

[0]：1表示时钟锁定；0表示时钟失锁

1. 温度监控寄存器:

地址0x0301，只读，读取FEB的温度

[15:0]: 温度回读，回读值与温度的函数关系；

温度值= (sqrt(pow(0.39083, 2) - 4 \* (-0.005775)\*(1 - double(temp) \* 0.0518798828125 / 1000)) - 0.39083) / 2 / (-0.00005775); 保留1位小数

1. 偏压监控寄存器：

地址0x0304，只读，读取监控的高压

[15:12]：保留，默认值0；

[11:0]：回读值与真实高压函数关系；

偏压值= 33.01 - (回读值\* 0.0118001302083333);

1. 计数率监控：

地址0x0305-0x30C，只读，计数率信息监控，单位：事件数/2s

## 3.2.5 数据接口定义

CuteWR通过UDP端口将采集到的SE数据发送到10G Switch交换机；10G Switch交换机将10个CM发送来的UDP数据包解析，提取有效数据后，存储到各自的RX\_FIFO中；Timer模块负责物理延迟恢复，即读出每个RX\_FIFO中的gamma事件数据，并按照事件中包含的物理时间戳进行物理延迟后（模拟gamma事件的真实到来时刻，单纯增加物理延迟，所有通道的相对延迟为0）输出给符合逻辑；符合逻辑进行符合事件挑选，并存储到COM5402 对应UDP数据通道的TX FIFO中；COM5402将UDP TX FIFO中的符合数据按照UDP协议打包后发送给服务器数据接收用UDP(UDP\_D)端口；算法服务器与底层建立连接并进行数据传输，并按照数据打包格式，进行数据解析，并进行后续图像重建等流程。

### 3.2.5.1 物理接口

光纤，全双工通信。

### 3.2.5.2 通讯协议

数据输出协议为UDP，

工作流程：CuteWR发送SE数据发送到*10G Switch*交换机；*10G Switch*交换机解析SE数据，并交给符合模块；符合模块进行符合筛选，交给发送模块；发送模块通过UDP端口将数据发送给APC。

### 3.2.5.3 数据包格式

UDP数据包，包长度为8字节的整数倍，在不增加同步字段的前提下，必须保证Cute\_WR打包、以及WhiteRabbit Switch打包过程中是以8字节为单位的；

包括数据起始包、DE数据包、SE数据包、SERaw数据包、波形/自触发数据包、采集时刻数据包。

事件包数据格式定义如下：

数据起始包（64bit，8Byte），数据包起始判断，用于检验数据包是否错位：



图3-2数据起始包

DE数据包（64bit， 1个事件，64bit/事件），包括真符合事件和延迟符合事件：

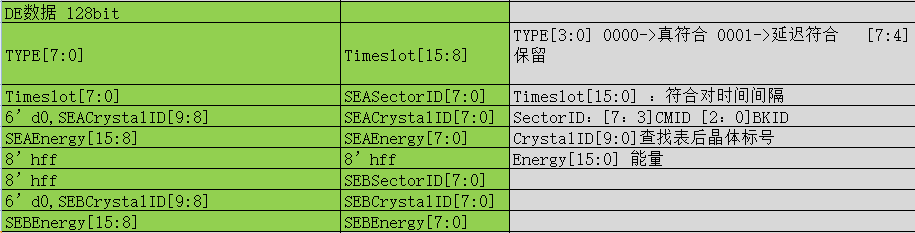


图3-3 DE数据包格式

Timeslot[15:0] 其中：

Bit15代表符合对的时间间隔的符号，1代表A事件的时间戳大于B事件的时间戳，0代表A事件的时间戳小于B事件的时间戳；

Bit14-0代表符合对的时间间隔的绝对值，单位为31.25ps，如果bit14-0为20，则对应的符合时间间隔为20\*31.25ps = 625ps

SE事件包（64bit， 2个事件，32bit/事件）：

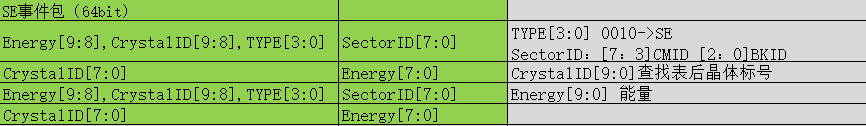


图3-4 SE数据包格式

SERaw数据包（64bit， 1个事件，64bit/事件））：

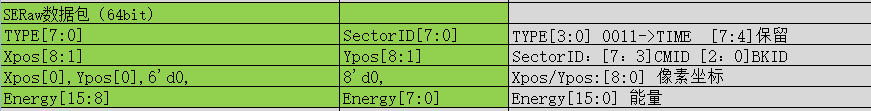


图3-5 SERaw数据包格式

波形数据包/自触发数据包（64bit， 1个事件，64bit/事件）：

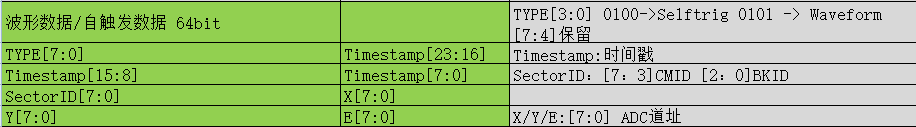


图3-6 波形数据包/自触发数据包格式

采集时刻数据包（64bit，8Byte）：

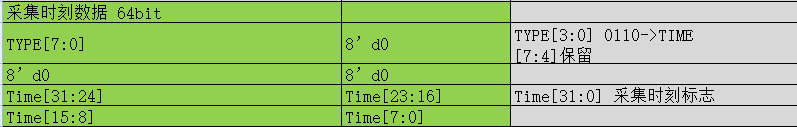


图3-7 采集时刻数据包格式

离线分析数据包

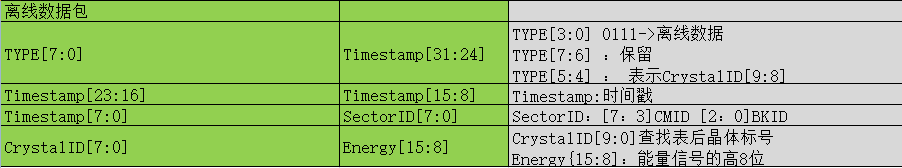


图3-8 离线数据包格式

## 3.2.6 WR交换机

在本设计中，WR交换机只用来进行时间同步，不进行数据传输。

### 3.2.6.1表模式数据格式

同3.5.3

### 3.2.6.2重建结果格式

32位float型，存储方式为xyz，xyz方向定义与系统坐标系相同,如下图所示：

PET系统坐标定义

## 4.2 电子学提供文件汇总

电子学提供文件如下：

10个CM初始化配置文件:B3PET\_CM0\_Config.xml ~ B3PET\_CM9\_Config.xml

10个CM时间校正表文件:B3PET\_CM0\_Timing\_Cal~B3PET\_CM9\_Timing\_Cal

10个CM能窗文件B3PET\_CM0\_EW~B3PET\_CM9\_EW