EDOC300

EUROBALL DATA ACQUISITION SYSTEM

Hardware Data Format

Edition 1.0 July 25 1995

Edition 1.1 June 18 1996

Nuclear Physics Support Group

Central Laboratory of the Research Councils
Daresbury Laboratory

Euroball Raw Data Event Format

This document defines the format of Raw Event Data within **Euroball** - ie the format of data generated by the electronics and passed to the data collectors. The format of full events generated within the event builder may (as in the case of **Eurogam**) be different.

It is assumed that compatibility with the **Eurogam** readout is necessary and hence the general format of the data follows the format described in the paper by J.Alexander ***DT32** bus specification Version 2 (**EDOC075**)*.

In **Eurogam** all data sources connect to one **DT32** bus which in effect acts as a hardware event builder. The **Start Event** token is generated within the **Trigger** module which is ensured to be the first data word within an event by the physical location of the **Trigger** module within the first **VXI** crate on the **DT32** bus. Each **ROCO** outputs an **End SubEvent** token for each event and the **End Event** token is just the **End SubEvent** token generated by the last **ROCO** on the **DT32** bus.

For **Euroball** it is required that there will be several **DT32** bus chains and hence a more flexible system is needed. However there can be no radical departure from the defined 32 bit structure since this is a feature of the **Eurogam** readout system.

Format of a data word

```
I xx I xxxxxx I xxxxxxxx I xxxxxxxxxxxxxxxx I

  qq    item     group           data
```

group 255 is used by the Trigger and ROCOs.
items 1 => 47 are allocated to ROCOs with the last ROCO in the chain always being 47

The **Eurogam Start Event** token has

qq = 11
item = 0x3f
group = 0xff
data = event number

and the **End SubEvent** token has

qq = 00 (good) or 10 (ROCO detected error)
item = ROCO number
group = 0xff

data = size of subevent in words (most significant 12 bits)
= least significant 4 bits contain copy of event number

For **Euroball** the requirement for more than one **DT32** bus means that an alternative for the Trigger produced **Start Event** Token has to be found. Also as the requirement is analysed a more general system is desirable. Each **ROCO** should be as independant as possible and the **ROCO** only should be responsible for generation of the tokens.

Within the general format the address fields (group + item) allow data items to be uniquely identfied throughout the system. Thus it is possible to identify the source of data at any time during processing. This feature is a valuable diagnostic tool and is preserved in the following definitions.

The requirement for **Euroball** is that we define both a **Start SubEvent** token and an **End SubEvent** token where the term **SubEvent** here refers to the contribution to the total event from a single **ROCO.**

Format of a token

```
| xxxx | xxxx | xxxxxxxx | xxxxxxxxxxxxxxxx |

 flags  item    group          data
```

Start SubEvent Token

```
| qq00 | xxxx | xxxxxxxx | xxxxxxxxxxxx | xxxx |

        ROCO    group      word      event
         no                count     number
```

End SubEvent Token

```
| q011 | xxxx | xxxxxxxx | xxxxxxxxxxxxxxxx |

        ROCO    group        check sum
         no
```

This format assumes no more than 15 **ROCO**s on a single **DT32** bus. The first in the chain **MUST** be ROCO Number = 1 and the last in the chain **MUST be ROCO Number = 15**. If there is only one **ROCO** on the **DT32** bus then it **WILL BE ROCO Number = 15**.

The group = 0xf0 (248) is assigned to the first **DT32** bus and groups 0xf1 (249), 0xf2 (250) etc are assigned to the second and third **DT32** bus. The maximum number of **DT32** bus chains allowed is 8 and thus groups 248 to 255 are reserved for this purpose.

The qualifier bits in the **Start SubEvent** token will normally be set to zero but may be set by the ROCO to indicate that the data which follows should not be trusted and used to construct an event. For example the ROCO may use these bits to forward data being discarded following a VXI readout error.

The **event number** field is the low order 4 bits of the Trigger event number as broadcast throughout the system.

The **word count** field is the total number of data items (words) within the subevent and includes all the tokens.

The **checksum** is calculated as the twos complement 32 bit sum of all data words within the event including the start and end tokens. The 32 bit sum is converted into a 16 bit checksum by addition of the most significant and the least sigificant 16 bits. For the purpose of the sum the value of the checksum field is taken to be zero.

The qualifier bit in the **End SubEvent** token will initially be set to zero but may be set to one by the event collector to indicate a checksum failure.

Each **SubEvent MUST** also contain a **EventNumber** data item which **MUST** immediately follow the **Start SubEvent** token.

EventNumber Data Item

```
| qq01 | xxxx | xxxxxxxx | xxxxxxxxxxxxxxxx |

         ROCO   group       event number
          no
```

The event number will be reconstructed using the method as described in the document ***Event Number generation within Euroball***. The qualifier bits will normally be set to zero but qq=10 will be used to indicate that a single step resynchronisation occurred and qq=11 will be used to indicate that a multi step resynchronisation occurred.

Each **SubEvent MAY** also contain a **EventNumber Extension** data item which if it is present **MUST** immediately follow the **EventNumber** data item. The **EventNumber Extension** extends the event number to 32 bits as described in the document ***Event Number generation within Euroball.***

EventNumber Extension Data Item

```
| qq10 | xxxx | xxxxxxxx | xxxxxxxxxxxxxxxx |

         ROCO   group       event number
          no                 extension
```

The qualifier bits will normally be set to zero.

The **Eurogam Trigger** module currently generates data using group 255. The format of **Trigger** data words are

| 1111 | xxxx | 11111111 | xxxxxxxxxxxxxxxx |

0 - 15 group data

Data of this format will be considered as a normal data item for group 255. For **Euroball** a redesign of the trigger module will allow use of a group other than 255. Group 247 has been reserved for the **Euroball** trigger.

Using the terms **Start Event** and **End Event** for the collection of all the **SubEvents** which can be readout on a single **DT32** bus we have

Start Event Token | xx00 | xxxx | 11111xxx | xxxxxxxxxxxxxxxx

and

End Event Token | xx11 | 1111 | 11111xxx | xxxxxxxxxxxxxxxx

Where x is a do-not-care bit.

To implement these requirements the firmware within the ROCOs must perform the following tasks

- generation of the Start SubEvent Token
- generation of the End SubEvent Token
- generation of the EventNumber data item
- generation of the EventNumber Extension data item

- generation of these tokens should be enabled and disabled via the command inferface

- group number (8 bits) to be defined via the command interface

- ROCO number (4 bits) to be defined via the command interface

- the settings to be used for the most significant 4 bits of the Start SubEvent Token, the End SubEvent Token, the EventNumber Data Item and the EventNumber Extension Data Item to be defined via the command interface

- Generation of the checksum to be enabled or disabled via the command interface. If disabled the data field of the End SubEvent token should contain zero bits.

These programming options allow virtually any protocol to be generated for the Tokens within the general outline format.