# BLM Corrected Sums

*Local application*

Thu, Dec 2, 2004

The Booster Beam Loss Monitor signals are summed by the local applicaiton BLMS. But they exhibit a kind of background in the absence of beam that is especially severe for the 400 Mev BLMs, where the integration takes place over a very short time. Simple corrections may apply for the 100 second sums that can easily be done in the front end. In order to better investigate the ability to make this correction, a new local applicaiton can be install,ed in such front ends that can take the sums produced by BLMS and correct them for a background component, producing new corrected sums. It must be done for each BLM and for each clock event. As BLMS is configured, there can be as many as 16 BLMs and 16 clock events, although in practice, the number of BLMs in use in one node may be 12 or less, and the number of clock events is 11. Let the new LA be called BLMC, for BLM corrected sums.

***Parameters for LA***

| *Text* | *Meaning* |
|---|---|
| ENABLE B | Usual enable Bit# for LA |
| BLMSUM C | Base channel# for BLM 100 second sums |
| #BLMS | Number of BLMs |
| EVTSUM C | Base channel# for clock event sums |
| PART #CY | Partial sum #cycles |
| BLMCOR C | Base channel# for BLM corrected sums |

The value of the PART #CY parameter is likely to be 00FA, or 250 decimal, which corresponds to one-sixth of 100 seconds. The 100 second sums built by BLMS are updated every 250 cycles, and this parameter is meant to match that.

About every 17 seconds, the values of the 100 second sums are examined, and using the counts of the clock events, corrected by subtracting a background that is determined by the event 12 sums. (Event 12 cycles never have any beam.) As an example, consider the case for computed the corrected event 1D sum for a given BLM. Call the uncorrected sum BLM1D, the count of event 1D cycles EVT1D, and the corrected sum BLM1DC, we have:

```
BLM1DC = BLM1D – (BLM12/EVT12)*EVT1D
```

The term BLM12/EVT12 gives an estimate of the background per cycle. This is used to correct the BLM1D sum, or any other sum. The result is deposited in a new raw floating point channel.

It would be better to perform the correction as soon as new data is updated from BLMS. One way to do this is to watch for changes in the uncorrected BLM sums, say either one of the event 11 or event 10 sums. As soon as a change is detected, do the arithmetic illustrated above. If we do it this way, we would not even need the parameter labeled PART #CY. Also, to the outside world, the uncorrected and corrected sums would appear to be updated simultaneously.