**DRAFT**
Version 0.2
February 24, 1998

# Error Handling of the SVT Hit Finder Board

J. Berryhill, R. Culbertson, H. Frisch, T. Nakaya, M. Shochet, P. Wilson
with the SVT and SVX-II Groups

University of Chicago

**Abstract**

This is a proposal for error detection, registering, and clearing for the SVT Hit Finder Board. Being a preliminary proposal, it has no contractual value nor is it an official reference of any kind, until we say so.

## 1 Introduction

This is summary of how errors are found and processed under the current design for the Hit Finder Board. All error handling logic is placed in FPGAs, so all of the protocols could be changed at any time by reprogramming the FPGAs *in situ.*

There are two varieties of Hit Finder Board errors; those which induce a reset of the SVT ("hard errors"), and those which do not ("soft errors").

Hard errors cannot be recovered from, and will ultimately result in a reset of the Hit Finder Board and the rest of SVT, in addition to a SPY cycle. Once a hard error is found, all subsequent events processed by the Hit Finder Board will be reported as having that error, until it is reset externally by a VME write or a CDF RECOVER or INIT. In order to mitigate persistent errors, hard error conditions, wherever possible, can be programmably masked over VME, so that they become soft errors, or so that they become ignored completely.

Soft errors are assumed to be single word glitches in data transmission, and should, by definition, have a good chance of being recovered from. They will initiate a SPY cycle, but will not in general initiate a reset. Soft errors are transient, in that subsequent events will not report this error (unless it occurs again). Soft errors can be upgraded to hard errors through a VME–programmable mask.

There are two stages in the error reporting process: generating a preliminary set of errors for each of the 10 input streams of the Hit Finder Board using the Hit Manufacturer (HITMAN) chip; then integrating and registering all of the errors using the Output Processor (OP) chip.

## 2 HITMAN Errors

The Hit Manufacturer (HITMAN) chip is an Altera FLEX10K50 FPGA, which accepts as input a single, demultiplexed SVX II data stream, and which creates as output a series of charge cluster centroids to a FIFO. There are 10 HITMAN chips corresponding to the 10 SVX II input streams, and 10 FIFOs collect their output. As errors are detected, a HITMAN accumulates them in a 16-bit end-of-stream word which is inserted into the output stream after the last $r - \phi$ chip has been processed. Bits 0-7 are a bunch counter number, bits 8-14 are error bits, and bit 15 is an end-of-stream indicator bit. Note that the end-of-stream word generated by a HITMAN does not correspond to the end-of-event word which is passed to the rest of SVT; it is a record internal to the Hit Finder Board.

G-link lost lock cannot be detected by a HITMAN, because it is clocked by a G-link. Lost lock errors are handled by the OP. The errors detectable to a HITMAN are bit transmission errors by a G-link receiver, or perhaps data transmission errors on the Hit Finder Board preceding the HITMAN stage.

Since the SVX II data stream is highly structured, there exist multiple opportunities for error checking. Each of the 10 HITMAN chips check for the following errors:

- **Chip out of order**: SVX III chip ID numbers arrive in a predetermined order, and the arrival of an unexpected chip ID creates an Invalid Data error (see below).

- **Strip out of order**: Channel numbers in the SVX II data stream should strictly increase; the arrival of a non-increasing channel number creates an Invalid Data error.

- **Premature data valid low**: A single data valid (DAV) signal accompanies each input stream. If DAV goes low before the first $r - z$ chip ID has arrived, a Truncated Data error (see below) is created.

- **VTM error**: The fiber optic receivers on the VTM output an error line, VTM_ERROR, which indicates an invalid control frame, probably due to a bit transmission error. After two consecutive VTM_ERRORs, the VTM automatically loses lock for that link. Single VTM_ERRORs do not cause lost lock, but they indicate invalid data. VTM_ERROR is an input for the HITMAN chip using that G-link; if VTM_ERROR is asserted when DAV is high, an Invalid Data error is created.

All of the error bits passed by the HITMAN result in soft errors. The HITMAN stores none of its error bits; all are cleared at the beginning of a new event. A HITMAN takes no direct action whenever it finds an error; it is simply reported and passed downstream.

# 3 OP Errors

The Output Processor (OP) chip is also an Altera FLEX10K50 FPGA. It receives a single stream of input from the merging stage of the Hit Finder Board. It outputs a single 23-bit stream of data to the Front Panel stage which connects to the Merger Boards. During normal running, the OP looks for the 10 separate end-of-stream words from all of the HITMEN, and creates the SVT protocol end-of-event-word upon the arrival of the last of these words. The OP end-of-event word has the bunch counter number in bits 0-7, the event parity in bit 8, errors in bits 9-16, bits 17-20 are unused, bit 21 is end-of-packet and bit 22 is end-of-event.

Unlike the HITMEN, the OP stores its errors and in general they must be reset externally. The complete set of registered errors is as follows:

- **LOSTSYNC**: The synchronization of the 10 different input streams has been compromised. This is detected by the OP by comparing bunch counter numbers from the 10 different end-of-stream words. The first end-of-stream word in an event is assumed to contain the correct bunch counter. All subsequent end-of-stream words for that event are compared to the first, and if the bunch counters do not match, a bit in a 10-bit LOSTSYNC register is set which corresponds to the offending stream number. The end-of-event word for this event will have bit 10 set high. This is considered to be a hard error, so all subsequent end-of-event words will have bit 10 set high.

- **FIFOOVERFLOW**: One or more of the 10 FIFOs which store the output of the 10 HITMEN has overflown. A bit in a 10-bit FIFOOVERFLOW register is set which corresponds to the offending stream number. The end-of-event word for this event will have bit 11 set high. This is considered to be a hard error, so all subsequent end-of-event words will have bit 11 set high.

- **INVALIDDATA**: One or more of the 10 end-of-stream words has indicated an invalid data error. A bit in a 10-bit INVALIDDATA register is set which corresponds to the offending stream number. The end-of-event word for this event will have bit 12 set high. This is considered to be a soft error, so the end-of-event word bit 12 will be cleared at the beginning of the next event.

- **TRUNCATEDDATA**: One or more of the 10 end-of-stream words has indicated a truncated data error, or a maximum number of clock cycles, in the OP, for a single event, has been exceeded. A bit in a 10-bit TRUNCATEDDATA register is set which corresponds to the offending stream number in the former case, and to all unfinished streams in the latter case. The end-of-event word for this event will have bit 13 set high. This is considered to be a soft error, so the end-of-event word bit 13 will be cleared at the beginning of the next event.

- **LOSTLOCK**: One or more of the 4 G-links has lost lock. A bit in a 4-bit LOSTLOCK register is set which corresponds to the offending G-link number. The end-of-event word for this event will have bit 15 set high. After the end-of-event word for a lost lock event has been generated, the OP will proceed to repeatedly generate end-of-event words at 30 MHz, which will be copies of the first end-of-event word with bit 15 set. This is considered to be a hard error, and the 30 MHz end-of-event word echo will continue until the LOSTLOCK register is reset.

This gives a total of $4 \times 10 + 4 = 44$ bits of error registers, all of which can be read or reset over VME at any time. It is important to note that the end-of-event word error bits and the aforementioned OP error registers are separate bits which are set and cleared differently.

There are multiple error masks which logically connect the OP error registers with the end-of-event error bits, with SVT_ERROR, and with CDF_ERROR, and currently all of them are planned to be VME programmable. The default properties of the masks are as follows: soft error bits in end-of-event words are reset every event (though the OP error registers are not), CDF_ERROR is masked from all errors, and SVT_ERROR is never masked. Thus SVT_ERROR is the OR of all 44 error register bits, and CDF_ERROR is never asserted.

Once an error register is set and an error line pulled, the Hit Finder continues to process data (or echo the end-of-event word for the LOSTLOCK case) and produce output, with the appropriate error bits set in the end-of-event word. SPY cycles will presumably be performed, and the OP error registers can be read over VME, in addition to the contents of the SPY buffers. If the error is a hard error, the Hit Finder Board will ultimately have its error registers cleared by the OR of the INIT and RECOVER signals.

# 4 Questions

Which errors should be soft and which should be hard? Ultimately the determining factor is which kinds of errors incur the most dead time on SVT, and how often they happen. This will not be known beforehand, so one must be careful to be flexible and programmable whenever feasible.