

Speeding Up DSP's Hit Processing

Ron Moore – FNAL

- Overview of hit processing
- Tweaking existing code
- New data formats
- Summary

L2A Processing Overview

- What happens on TDC before/after L2A?
 - DSP polls event register for L2A
 - TDC sees L2A, automatically sets not LOCAL DONE
 - DSP sees L2A bit asserted, begins processing
 - Gets L2 buffer number, scan list, bunch ID for header word
 - For all TDC chips not marked to be skipped:
 - Read raw edge words, combine leading/trailing edges into pulses
 - Format hit word containing calibrated leading edge time, pulse width, channel identifier, “first hit on channel” flag
 - Fill VME FIFO with header word and data words
 - DSP sets LOCAL DONE, indicating ready for readout
 - DSP starts polling event register for L2A again

Hit Processing Overview

- Read edge words from TDC chips (N.B. - 2 μ s buffer!)
- Pair up leading and trailing edges to form a complete pulse
- Check for unpaired leading, trailing edges
- Process hits with leading edges within programmable time window up to a programmable max # hits per channel
 - For COT: max time = 500 ns, max # hits/channel = 8
 - Excess hits or hits outside of time window are ignored
 - Shortcoming - must read out TDC chip to empty it
- Calculate pulse width, apply calibration constant to time
- Merge channel id, “first hit on channel” flag in hit word

TDC Chip Data Format

Bit(s)	Name	Function
31	Chip Empty	0 = not empty 1 = chip empty
30-12	-	unused
11	Edge sense	0 = trailing edge 1 = leading edge
10-0	Edge time	(uncalibrated) edge time (ns)

Event Data Format

Header word:

Bit(s)	Name	
31-23	Module ID	Stored in FRAM 1
22-20	Scan List	From event register
19-18	L2 buffer number	From event register
17-8	# hit words	# hit words to follow
7-0	Bunch ID	From FPGA registers

Hit word:

Bit(s)	Name	
31	First hit	(0) 1 = (not) first hit on channel
30-22	Channel ID	Specified by look-up-table in FRAM 1
21-11	Pulse width	Trailing edge time – leading edge time
10-0	Hit time	calibrated time of leading edge(ns)

Current DSP Code Version - V35

- Useful parameterization of processing time (not DONE \Rightarrow DONE):

$$T_{\text{DSP}} [\text{ms}] = 83 + (280/96) * \sum n_{\text{ch}} + (65/96) * \sum (N_{\text{ch}} - n_{\text{ch}})$$

Where:

ch = channel numbers to be summed over (0-95)

n_{ch} = # hits in channel with desired time window, up to specified max #

N_{ch} = # hits in channel within the full 2 μs buffer

- Processing time sensitive to # hits to process and # hits to ignore
- CDF 5824 contains processing and readout time during physics runs
 - concern that processing + readout time not fast enough for Run IIb

\Rightarrow What can be done to speed up the DSP's hit processing?

(Other than adjusting time window or max # hits?)

Tweaking the Existing Code

- Found 4 more cycles per hit word to save!

(It was good not looking at code for ~ 1 year.)

Estimated processing time:

$$T_{\text{DSP}} [\text{ms}] = 83 + (264/96) * \Sigma n_{\text{ch}} + (65/96) * \Sigma (N_{\text{ch}} - n_{\text{ch}})$$

- Advantages?
 - Retain full formatting from TDC
- Disadvantages?
 - Only minor improvement in speed – need more

⇒ Continued tweaking will not improve speed significantly

New Ideas for Speed

- Change data format from TDC and let reformatter convert to desired bank structure
- Two ideas considered:
 - Minimize formatting done by DSP
 - Let DSP write hit words directly to VME FIFO

Minimal Formatting by DSP

- DSP puts “almost raw” edge words into FIFO
 - Two data words for every hit
 - Just put channel id into edge word
 - Empty chip when any edge outside of time window
 - Add calibration const. w/o penalty
 - No pulse width
 - No first hit flag

New hit word:

Bit(s)	Name	Function
31	-	unused
30-22	Channel ID	Specified by look-up-table in FRAM 1
21-12	-	unused
11	Edge sense	0 = trailing edge 1 = leading edge
10-0	Edge time	calibrated edge time (ns)

Minimal Formatting by DSP cont'd

- All info available for reformatter to create usual bank

Estimated processing time:

$$T_{\text{DSP}} [\mu\text{s}] \leq 60 + (227/96) * \sum n_{\text{ch}} + (65/96) * \sum (N_{\text{ch}} - n_{\text{ch}})$$

- Advantages?
 - Less overhead (60 vs 83 μs)
 - Faster per processed hit (227 vs 264 $\mu\text{s}/\text{hit}$ in every ch)
- Disadvantages?
 - Doubles the data volume read out from TDC

DSP Writes Hit Words Directly to FIFO

- Currently, DSP temporarily stores hit words on stack in order to count data words so that header word can be put into FIFO first.
- Propose to let DSP write hit words directly to FIFO and put header word into FIFO last.
- Keep data word count in SRAM where crate controller can fetch it before reading out FIFO.

DSP Writes Hit Words Directly to FIFO (2)

- Reformatter just has to put header word first?
- Put in existing code or minimal formatting algorithm
- Advantages?
 - Estimated processing time savings:
 - $\sim 1 \mu\text{s}$ overhead?
 - At least 14/96 μs per processed hit
- Disadvantages?
 - Cannot use “early tracer done” mode at all (interaction between the PPC and TDC VME interfaces)
- Need to investigate this option more thoroughly

Processing Time Summary

Algorithm	Overhead	Time per processed hit in <u>every</u> channel	Time per unprocessed hit in <u>every</u> channel	Time for 1 hit/channel in window + 2 hits/channel outside of window
Current V35	83	282	65	495
Tweaked V35	83	264	65	477
Minimal DSP formatting	60	227	65	417
Tweaked V35 + hit words direct to FIFO	82	251	65	463
Minimal DSP formatting + hit words direct to FIFO	59	213	65	402

All times in ms

Summary

- No factors of 2 to gain in DSP processing speed
- Tweaking current algorithm provides little improvement
- Greater speed increase needs new data format options, plus use of reformatter to convert to desired bank format
- Will benchmark the proposed DSP code modifications to get firm timing info