Error Detection/ Correction Policy

James T. Linnemann
Michigan State University
BU STT Workshop
December 19, 1999
Error Handling Strategy

- Confine errors to single event, single channel
  - missing an event boundary: event synch error
    - only SCL_INITIATE will clear this

- count errors as detected
  - VME readback in monitoring for diagnosis

- no elaborate recovery or detection
  - unless can be FIXED locally
    - and faster, more reliably than operator intervention
  - else: support diagnosis, save evidence
    - data flow hang can usually localize problem

- Process malformed events to extent possible
  - tag in header (L2 has standard bits defined)
SCL INITIALIZE

why we avoid it

- Needed if event fragments don’t match
  - must clear all buffers EVERYWHERE and restart
  - violent: touches EVERY front end crate

- Avoidance:
  - redundancy header to trailer (protect 1-bit errors)
  - try to preserve event format (to find trailer)
  - try to preserve event boundary (else must re-init)
    - detect missed event boundary (end or begin)
    - send pads before End Event to reframe if needed
Synchronicity

● Link synchronization losses (physical layer)
  • phase (frame boundary) loss
  • frequency lock loss
  • strategies:
    – stop event flow
    – auto-resynch on pads (cypress)

● Event synchronization loss
  • event number mismatch
    – only detectable where event numbers exist
      ● cft tracks (and L1 SCL info)
      ● not SMT data (until you add a header/trailer)
      ● clusters to fit card
Physical Layer
Synchronization Loss

- Loss of frame boundary (phase) lock
  - cypress: assumed after n consecutive errors
- Loss of frequency (PLL) lock (chipset)
- may have different recovery times
- G-link and Cypress auto-recover on pads
  - between two events possibly (Cypress)
    - preserve event boundary!
  - G-link: raise L1 BUSY
    - only useful because unbuffered: busy stops event flow
    - generates a stream of pads to resynch on
      - allows recovery without SCL Initialize
Event Synchronization Loss

- only detectable where event numbers exist
  - cft tracks (and L1 SCL info)
  - not SMT data (until you add a header/trailer)
    - beware: event end is just special data value
  - clusters to fit card

- Fight loss of event boundaries!
  - Input 2-state machine: event/idle
    - 2 starts, 2 stops insert boundary
    - works with frame synch recovery: ready for next event means will see next start even if previous stop lost
  - redundancy between header and trailer
    - if headers mismatch but correct number in trailer, OK

- Lost boundary? don’t guess which source: SCL_INIT
Event Synch Error Handling: need report to central point

- FRC notified (Admin Alpha in rest L2)
  - sets ERROR1 in SCL (and records time)
  - will provoke SCL_INITIALIZE when ack. from FW
  - Header status bit is sufficient (from STC say)
    - no gain for “immediate” notify via say special character
    - could define so onlyTFC reports to FRC…
    - no pressure to handle SCL_INITIALIZE in 10 sec say
      - could make re-synch of links part of drill
        » if no special mode of Xmit required to resynch…
L2-style Geographic Section

- Will propose modification to GS protocols
  - VBD initialization with crate info
    - maybe re-initialize by computer
  - time-stamped record of L1, L2 error declarations
    - handled by "Dallas Chip" in FE sections
    - TCC must be involved for L2
  - L1 busy handling rationale differs
    - busy not guaranteed to halt event flow (when buffered)
      - a serious confusion already exists if > 16 events
        » let buffers fill, SCL_INIT when mismatch found
    - emphasis on diagnosis, since prevention not guaranteed
L1 Busy

- Only use it when it really helps you:
  - possibly during SMT or CFT input?
  - possibly during link recovery for these G-links
  - possibly during link recovery IFF
    - buffered
      - if it’s a buffered link, won’t stop the data source from sending next event: will get SCL_INIT anyway
    - can recover on pad characters alone
      - if you need to set the transmitter in a resynch mode, you need to do something more violent than a data pause
  - alternative is to scl_init now (don’t finish until link recovers)
VRB surprises…
may impact your CFT inputs?

- VRB G-link inputs intrinsically 8-bit oriented
  - 16-bit word stream split into two 8-bit buffers
    - read all upper halves, then lower halves (or vice versa)
    - great for SMT data; not so hot for CFT tracks...
  - Event boundary:
    - SMT: 16b data, end event from special value (no begin?)
    - CFT, L1FW,L1 Cal:
      - 16b data + 4 control bits = 20b in 24b frame
      - bits 17,18 tag begin/end goes with 1 8-bit stream
      - bits 19,20 goes to other: send begin/end tags twice
# L2 Header

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Notes/Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>B0</td>
<td># objects (NOT IN HEADER)</td>
<td>[note 255 max!]</td>
</tr>
<tr>
<td>B1</td>
<td>Header Length in 4B words (1B)</td>
<td>[=3 for default]</td>
</tr>
<tr>
<td>B2</td>
<td>Object Length in 4B words (1B)</td>
<td>[ALL same size!]</td>
</tr>
<tr>
<td>B3</td>
<td>Header/Trailer Format # ( hi 3 bits)</td>
<td>[ONLY changes if new format]</td>
</tr>
<tr>
<td></td>
<td>Object Format # ( lo 5 bits)</td>
<td>[ONLY changes if new format]</td>
</tr>
<tr>
<td>B4</td>
<td>Data Type # (1B)</td>
<td>[unique in all L2 MBT inputs]</td>
</tr>
<tr>
<td>B5</td>
<td>Bunch # (1B)</td>
<td></td>
</tr>
<tr>
<td>B6-7</td>
<td>Rotation# (2B)</td>
<td>[B6 is LSB of rotation]</td>
</tr>
<tr>
<td>B8</td>
<td>Algorithm Major Version (1 B)</td>
<td>[e.g. 7 from Version 7.1]</td>
</tr>
<tr>
<td>B9</td>
<td>Algorithm Minor Version (1B)</td>
<td>[e.g. 1 from 7.1]</td>
</tr>
<tr>
<td></td>
<td>or Processor Specific Bits (1B)</td>
<td>[esp. if hardware data source]</td>
</tr>
<tr>
<td>B10</td>
<td>Processor Specific Bits (1B)</td>
<td></td>
</tr>
<tr>
<td>B11</td>
<td>Status Bits</td>
<td>[b7 on means some error]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[some standard for L2 Proc]</td>
</tr>
</tbody>
</table>
Standard Status Bits
b7, b0 for all; others if L2proc

7 error on event (any kind): use at own risk
6 no processing attempted (none required)
5 object list truncated (any reason)
4 Receiver error on some input physical trailer
3
2
1 more data-type info (processor-specific)
0 0 for real data, 1 for MC data
L2 Trailer

B0  Bunch # (1B) = B5 of Header
B1  Data Type # (1B) = B4 of Header
   (Swapped even/odd from Header)
B2  Longitudinal Parity of even Bytes
B3  Longitudinal Parity of odd Bytes
   or--if parity too slow to calculate, Turn # (B6-7 of Header)
      MBT Out, SLIC, FIC will append physical trailer with 8-bit
      hardware-generated longitudinal parity

Zero padding to 16 B group FOLLOWS trailer, before End of Event
L2 Physical Trailer

- FIC, SLIC, MBT Out: add a physical 2B trailer
  - after logical trailer, before End Event
    - This BREAKS 16B boundary, but handled by MBT
  - B0  8 bit longitudinal parity of received data
  - B1  Status Bits  [b7 on if any receive error]
    - not included longitudinal parity!
    - b0, b1 are type ID: 0 = FIC, 1 = SLIC, 2 = MBT

- MBT inputs place this in B0, B1 of 16B physical trailer
  - adds B14, its own longitudinal parity of everything received
  - B15 its own Error Bits  [b7 on if any receive error]
  - reserves 4B for incoming, may give error locations in B4-13
  - MBT Outs produce 2B physical trailer like FIC
Endianness and Unpacking

- So Far, spec covers only getting into Alpha memory
- Work has started on understanding how this propagates to L3 and offline
Reframing Cypress: On Provocation model

- **During Reframing:**
  - no data into FIFO
  - stays in reframe until successful (identifies Pad)
    - any benefit of timeout?
  - upstream buffers may fill, generating L1 Busy
  - LED to indicate reframing? Counter/flipflop?
    - How to identify channel? Rotary: channel n or ALL?

- **Causes of reframing:**
  - Powerup
  - n consecutive bad or unknown control char
  - front panel (flipflop?)
  - VME (SLIC, MBT): part of SCL Initialize handling
    - Admin. will hold off clear of L1 Busy: read status
State bit (IDLE/EVENT) is input to FIFO handling logic
Begin Event

- Be sure input fifo correctly aligned
- ready to resume inserting data into FIFO
End Event

- Close up FIFO: stop inserting data
- Mark FIFO with end of event tag
- pad FIFO to 16B? (in case of errors)
  - probably handled by reset on Begin Event
- Add physical trailer recording errors
  - On readout of FIFO
Input during IDLE

- FIFO DISABLED
  - data format insensitive to errors in PAD’s
- Pad Ignore (even tho clocked)
- Data count (IDLE errors)
  - 2 consecutive DATA = BEGIN? Probably not...
- Error count
- Special Character:
  - Begin normal start of EVENT (align FIFO)
  - End assume missed Begin
  - any other: count
Input during EVENT

- **FIFO ENABLED**
  - sensitive to errors in PADS
  - try to preserve data format
- **Pad** Ignore (even tho clocked)
- **Data** insert into FIFO
- **Error** insert into FIFO, tag ERROR

- **Special Character:**
  - **End:** tag End, normal start of IDLE
  - **Begin:** assume missed End
  - **other:** insert into FIFO, tag ERROR (?)
Cypress Self Test

- Excellent for debugging, Bit Error Rate testing
  - transmitter: send sequence of ALL symbols
  - receiver: verify got sequence in order
  - SLIC, MBT: both Xmit and Rcv on board

- Start/stop can’t be special characters!
  - All: VME (or Mbus for MBT)
    – and a front panel flipflop?
  - test point(s) on front panel?
    – Should see a pulse every cycle through test sequence
    – or just an error counter for VME reading