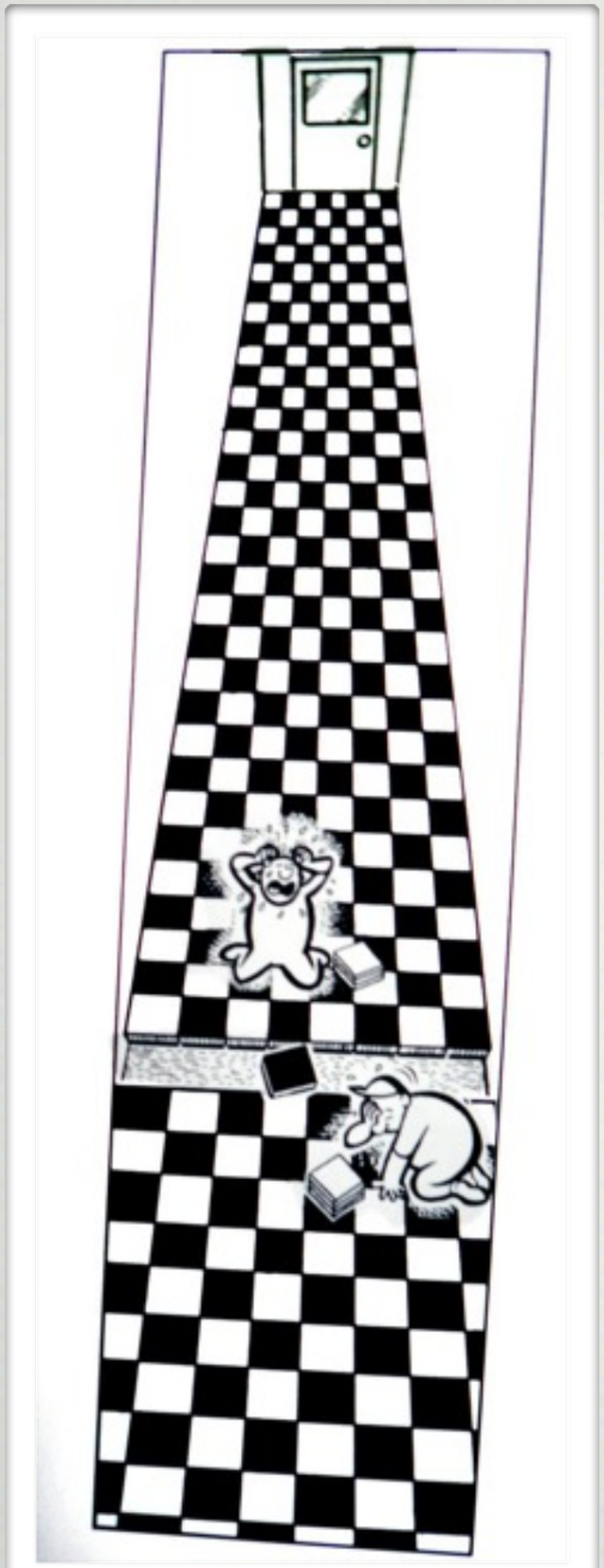


SVD UNPACKER: NEWS FROM THE ONLINE - OFFLINE FRONTIER

EUGENIO FOR THE SVD UNPACKING TEAM:

GIULIA CASAROSA, PETER KVASNICKA, TOBIAS SCHLÜTER

TIMELINE AND CONTRIBUTORS

- [ ] GIULIA AND I VOLUNTEERED TO WRITE THE SVDUNPACKER MODULE DURING THE LAST SVD-PXD JOINT MEETING
- [ ] PETER KINDLY MADE A MAJOR REWORK OF THE CODE IN DECEMBER
- [ ] IN JANUARY THE SVD UNPACKER MISERABLY FAILED IN DECODING THE FIRST DATA FROM DESY
- [ ] TOBIAS KINDLY AND PROMPTLY ACTED AS A SUBSTITUTE (GIULIA WAS TRAVELING AND I WAS IN VACATION) TO FIX SEVERAL THINGS:
  - [ ] THE FTB HEADER AND TRAILER NO ONE TOLD US ABOUT
  - [ ] THE BLUNDER WE MADE WITH THE ORDER OF THE BITFIELDS (BIG ENDIAN VS LITTLE ENDIAN)

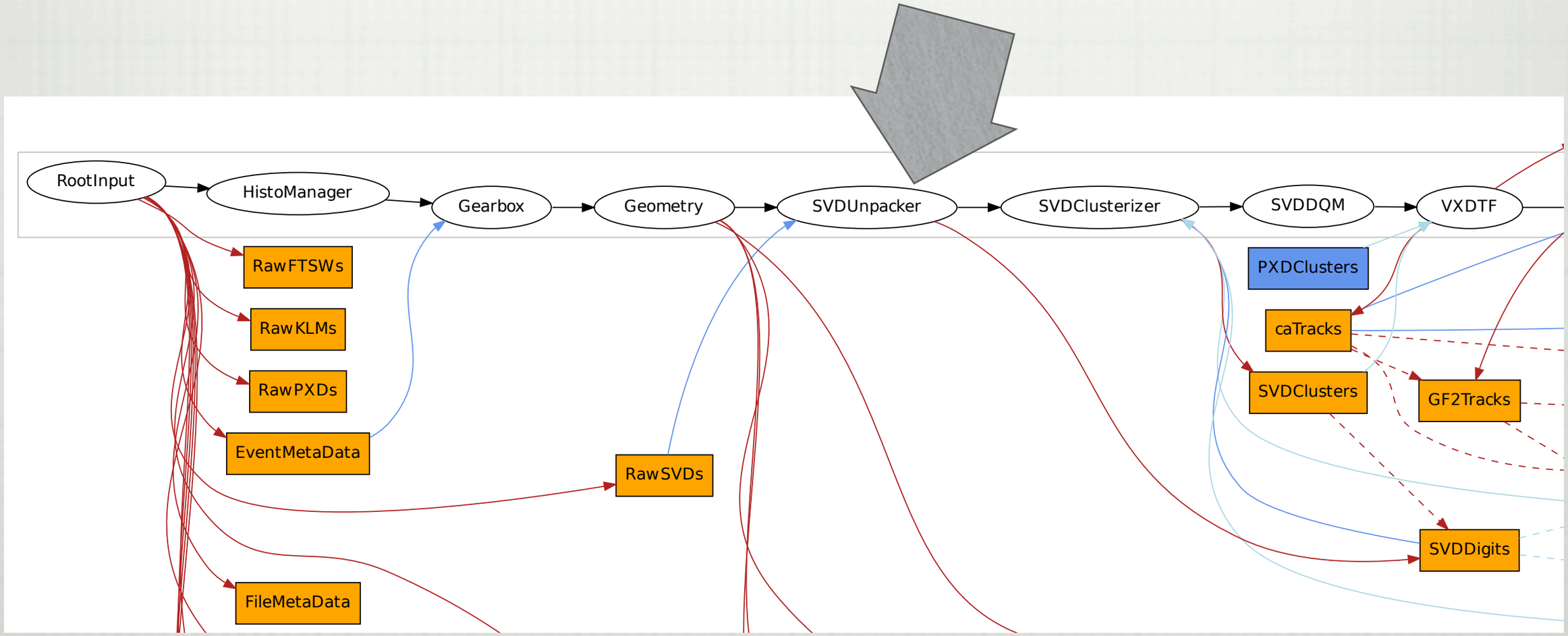
LESSONS LEARNED

- [ ] THRUST IN TOBI: EVENTUALLY HE WILL FIX YOUR BLUNDERS

- [ ] IF YOU DO NOT THRUST IN TOBI, OR TOBI IS AWAY OR WORKING ON SOMEONE ELSE BLUNDERS: PLEASE WRITE CLEAR AND COMPLETE DOCUMENTATION FOR THE REST OF US

PURPOSE OF THE SVD UNPACKER

- [ ] MAKE THE SVD DATA AVAILABLE IN THE BASF2 FRAMEWORK: I.E. CONVERT THE RAW DATA FROM THE COPPER TO A LIST OF SVD DIGITS



SVD UNPACKER PARAMETERS

```
-bash-4.1$ basf2 -m SVDUnpacker
>>> basf2 Python environment set
>>> Framework object created: fw

===============
  SVDUnpacker
===============
Description: Produce SVDDigits from RawSVD. NOTE: only zero-suppressed mode is
currently supported!
Found in:    /belle-rpc2/user/paoloni/belle2/2014-01-23/modules/Linux_x86_64/
debug/libsvdUnpacker.so
Package:     svd
--------------------------------------------------------------------------------
Parameter         Type  Default Description
--------------------------------------------------------------------------------
APVLatency        float 0.0      APV latency (in ns)
APVSamplingTime   float 1.0      APV sampling time (in ns)
rawSVDListName    str            Name of the raw SVD List
svdDigitListName  str            Name of the SVD Digits List
xmlMapFileName    str            path+name of the xml file
```

CAVEAT EMPTOR!

FAKES NUMBERS

STEERING FILE

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

import os
import sys

# etc... etc.. etc...

SVDUnpack = register_module('SVDUnpacker')

SVDUnpack.param('xmlMapFileName', 'testbeam/vxd/data/SVD-OnlineOfflineMap.xml')

# etc... etc.. etc...

main.add_module(SVDUnpack)

# etc... etc.. etc...
```

DAQ TO OFF-LINE MAP

- [ ] THE SVD DAQ KNOWS ABOUT:
  - [ ] FADC NUMBER, APV NUMBER, APV CHANNEL NUMBER
- [ ] THE OFF-LINE RECONSTRUCTION NEEDS:
  - [ ] LAYER NUMBER, LADDER NUMBER, SENSOR NUMBER, SIDE, U/V COORDINATES (~ STRIP NUMBER)
- [ ] A GENERAL MAP IS NEEDED

MAP FORMAT

```
<SVD>
    <layer n="3">
        <ladder n="1">
            <sensor n="3">
                <side side="u">
                    <chip n="5" FADCn="129" strip_number_of_ch127="767" strip_number_of_ch0="640"></chip>
                    <chip n="4" FADCn="129" strip_number_of_ch127="639" strip_number_of_ch0="512"></chip>
                    <chip n="3" FADCn="129" strip_number_of_ch127="511" strip_number_of_ch0="384"></chip>
                    <chip n="2" FADCn="129" strip_number_of_ch127="383" strip_number_of_ch0="256"></chip>
                    <chip n="1" FADCn="129" strip_number_of_ch127="255" strip_number_of_ch0="128"></chip>
                    <!--not connected to strips-->
                    <chip n="0" FADCn="129" strip_number_of_ch127="127" strip_number_of_ch0="000"></chip>
                </side>
                <side side="v">
                    <chip n="0" FADCn="1" strip_number_of_ch0="000" strip_number_of_ch127="127"></chip>
                    <chip n="1" FADCn="1" strip_number_of_ch0="128" strip_number_of_ch127="255"></chip>
                    <chip n="2" FADCn="1" strip_number_of_ch0="256" strip_number_of_ch127="383"></chip>
                    <chip n="3" FADCn="1" strip_number_of_ch0="384" strip_number_of_ch127="511"></chip>
                    <chip n="4" FADCn="1" strip_number_of_ch0="512" strip_number_of_ch127="639"></chip>
                    <chip n="5" FADCn="1" strip_number_of_ch0="640" strip_number_of_ch127="767"></chip>
                </side>
            </sensor>
        </ladder>
    </layer>
```

PITFALL: WHERE IS P? WHERE IS N? WHO IS U?

PHILOSOPHICAL CONSIDERATION

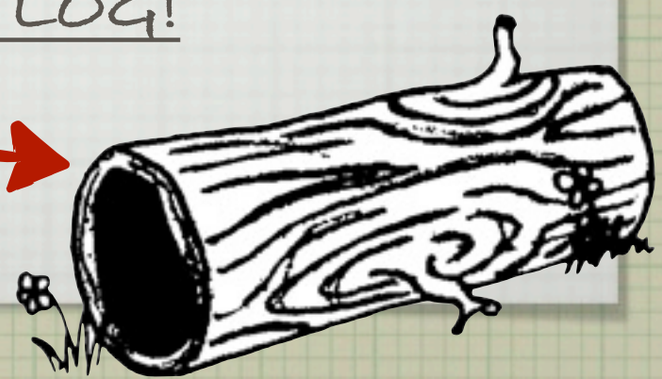
- BASF2 ON HLT CANNOT BE STOPPED BY A MODULE DETECTING AN ERROR. E.G.:

  ```
  assert ( MUST_be_true_or_die );
  ```

  THIS IS NOT AN OPTION.

  - BASF2 HAS TO CONTINUE TO RUN TRYING TO SURVIVE IN A CRUEL AND ADVERSE ENVIRONMENT IF ERROR ARE DETECTED.

- BASF2 COMPLETION IS NOT A GUARANTEE OF ERROR FREE OPERATIONS: YOU GOTTA WATCH THE LOG! AND I DO NOT MEAN THIS LOG



E.G. RUN 439(WHO IS GOING TO LOOK AT THE OTHER 438?)

```
[INFO] SeqRootInput :  181343 events read with total bytes of 5.63606e+06 kB
[INFO] SeqRootInput : event rate = 0.556454 (KHz)
[INFO] SeqRootInput : flow rate = 17.2944 (MB/s)
[INFO] SeqRootInput : event size = 31.0796 +- 10.7306 (kB)
[INFO] SeqRootInput: endRun done.
[INFO]  total failed checks = 0
[INFO]    m_wrongFTBHeader = 0
[INFO]    m_wrongFADCTrailer = 0
[INFO]    m_wrongFADCcrc = 0
[INFO]    m_wrongFTBcrc = 0
[INFO]    m_badEvent = 0
[INFO]    m_wrongFTBtrailer = 0
[INFO]  m_noAPVHeader = 0
[INFO]  m_noNewDigit = 0
[INFO]  m_NewDigit = 135464634
[INFO]  FTB Error Field
[INFO]    m_f0 = 1418
[INFO]    m_f3 = 0
[INFO]    m_f5 = 361272
[INFO]    m_f6 = 0
[INFO]    m_f7 = 0
[INFO] SeqRootInput: terminate called
```

~ 0.135 BILLION SAMPLES ~

~ 124 DIGIT/EVT ~

FTB #EVT == FADC #EVT

FADC #EVT != TTD #EVT

ERRORS: WHAT HAVE WE TO DO?

- [ ] AT PRESENT WE FORGIVE THE WRONG TTD EVENT NUMBER AND OTHER MISALIGNMENTS IN THE EVENT COUNTERS
  - [ ] OPTION A: IF AN ERROR IS DETECTED FORGET THE RAWSVD
    (AT PRESENT ~99.6% WILL BE THROWN AWAY)
  - [ ] OPTION B: FIX THE ERROR FIELDS IN THE FTB HEADER
  - [ ] OPTION C: DO NOT CHECK THE ERROR FIELD IN THE FTB HEADER

ERRORS: WHAT HAVE WE TO DO?

- [ ] WHAT IF THE FTB CRC FAILS?
- [ ] WHAT IF ONE OF THE FTB FLAGS FIELD REPORTS AN ERROR?
- [ ] THESE DECISION SHOULD NOT BE TAKEN BY THE SVDUNPACKER DEVELOPERS ALONE
- [ ] PLEASE LET US KNOW WHAT WE HAVE TO DO A PART RECORDING THE ERROR ON THE LOG

FTB CRC CHECK

- [ ] THANKS TO HARA-SAN! HE PROVIDED US TWO RECIPES TO EVALUATE THE CRC. THE FIRST ONE IS A DIRECT C++ IMPLEMENTATION OF THE VHDL (FOR TEST PURPOSES: NOT EFFICIENT) THE SECOND USES BOOST... BUT WITH ENDIANNES SWAP :(

```
for (int i = 0; i < nWords; i++)
  tmpBuffer[i] = htonl(data32_start[i]);

//compute crc
boost::crc_basic<16> bcrc(0x8005, 0xffff, 0, false, false);
bcrc.process_block(tmpBuffer, tmpBuffer + nWords);
unsigned int checkCRC = bcrc.checksum();
//  B2DEBUG(1,"OUR crc = "<<std::hex << std::setw(8) << s
```

COPY AND SWAP ENDIANNESS

CRC EVALUATION

- [ ] ANY SUGGESTION ON HOW TO AVOID THE COPY AND SWAP?

PLANS FOR THE FUTURE

- [ ] ONCE THE TEST BEAM WILL BE COMPLETED
  - [ ] THE ON-LINE TO OFF-LINE MAP SHOULD BE HANDLED BY THE BASF2 GEARBOX (I.E. WITHOUT SPECIFYING A FILE NAME)
  - [ ] THE ON-LINE TO OFF-LINE MAP SHOULD BE WRITTEN IN TERMS OF P-SIDE , N-SIDE AND PHYSICAL STRIP NUMBER
  - [ ] AT PRESENT EACH SVDDIGIT STORES A SINGLE SAMPLES THERE SHOULD BE A SINGLE RAWSVDDIGIT CONTAINING THE 6 SAMPLES ALLTOGETHER

  - [ ] THE #STRIP, SIDE TO U-V COORDINATE SHOULD BE CODED IN THE MAP

PLANS FOR THE FUTURE II

- [ ] IF YOU ARE NOT ANNOYED, AND YOU PROMISE TO GIVE US FULL SUPPORT AND DOCUMENTATION, THEN WE ARE GOING TO CODE THE UNPACKER FOR THE OTHER MODES