SVT Vertical Slice Test

- Goals
- Schedule
- Hardware to be tested
- Software
  - ➢ pre-existing
  - ➢ developed during the test
  - ➢ developed once test is over
- Support needs from other groups
- How to get there
  - ➢ Open issues
  - ➢ Manpower
  - ➢ Responsibilities

GOALS I

- **SYSTEM INTEGRATION** test:
  - ➢ each boards works at B0 as at home
  - ➢ boards built by different groups talk to each other
  - ➢ many boards in a single crate
  - ➢ SVT people get to talk to each other

GOALS II

- **SYSTEM FUNCTIONALITY** test:
  - ➢ test data flow on full system
  - ➢ from hits to fitted tracks
  - ➢ test at full design speed for extended amount of time
  - ➢ test global control & communication

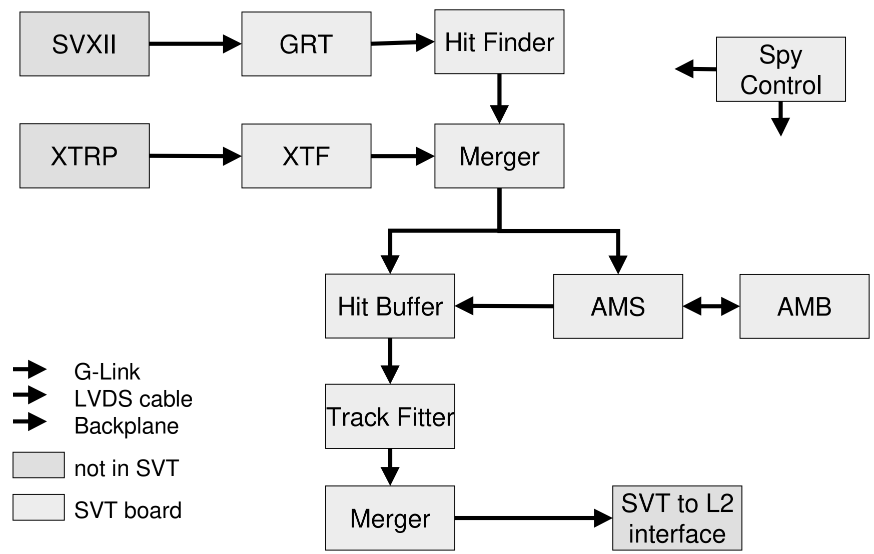
GOALS III

- **REAL LIFE EXERCISE:**
  - ☞ last chance to single out missing requirements (if any) before design is frozen for production
  - ☞ exercise room and development platform for online software

Test Schedule

- Final prototypes for all boards by beginning of 1999 (Luciano's talk)
- SVT assembled by end of 1999 (John People's plan)
- board production starts around June 1999 (to be ready by Xmas)
- 2 ~ 3 month window in April/May (maybe March)
  - test at home until end of February
  - use March to assemble hard/soft at Fnal
  - April/May for system exercise

SVT final configuration ("dream test")



Test Issues

- Real test requires processing ppbar-like data at 50kHz for days in a row, continuously and continuously checking for errors
- Need to emulate SVX readout, XFT Level 1 and Level2… too tough !
- Simplifications: **extensive vs. intensive**
  - INPUT DATA GENERATION
    - ☞Generate data from CPU and download to VME: full random test of everything, but very low duty cycle
    - ☞Prepare a few typical events and loop on them forever: full duty cycle, but limited variation
  - TEST FOR CORRECT RESULT:
    - ☞Read into CPU: full check, but very low duty cycle
      - variation: run into trash basket at full speed and read Spy Buffers
    - ☞Stick to a few typical events and compare with expectations at full speed inside Merger

Test strategy

- Will do some intensive and some extensive tests
- **Extensive** is easy, "just software". Will do with whatever boards will be available.
- I call extensive tests "slow", since duty cycle is poor.
  Do not misunderstand: board clock still ticks at design frequency: data flow at 30 MHz, but system is idle 99.999% of the time.

- **Intensive** is hard, needs to build the hardware with the test in mind, special issue for the Merger, also need care for which boards to install in the test stand and how to connect them.
- Two degrees in "intensiveness"
  - **run fast, check slow**
  - **run fast, check fast**

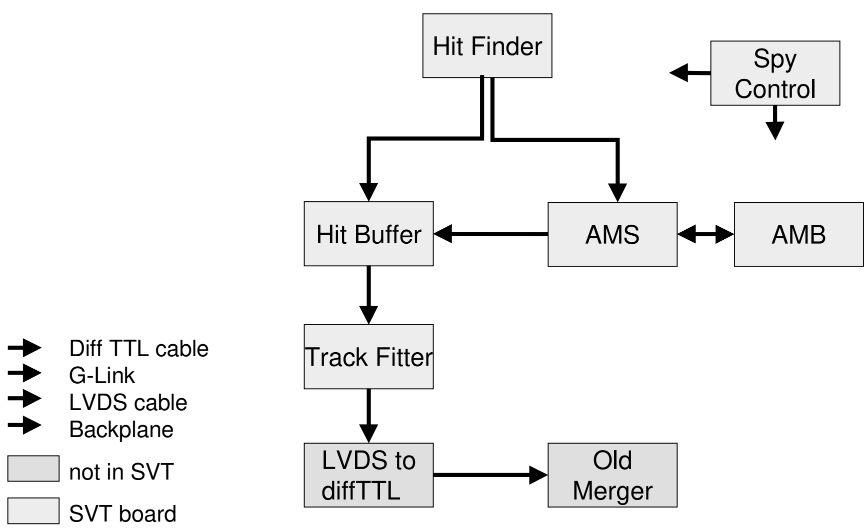
Hardware List

- SVT boards
  schedule & status for these in Luciano's talk
  - XTF
  - GRT
  - HF
  - Merger (will need 2)
  - Spy Control
  - AMS
  - AMB (nice to have 2)
  - HB
  - TF
- Non-SVT boards built by SVT group just to test SVT boards
  - GSTM = SVX simulator: VME memory that sends data on G-Link
    - ☞ **being built**
  - LVDS ↔ diff TTL translator
    - ☞ **being built**
- Non-SVT boards built by other groups for CDF trigger
  - SVT-Level2 interface board
    - ☞ **being built (Jane Nachtman)**
  - XTRP
    - ☞ **by summer end (Mike Kasten)**
  - SVX read out
    - ☞ **Ray's talk**
  - TRACER + whatever it needs

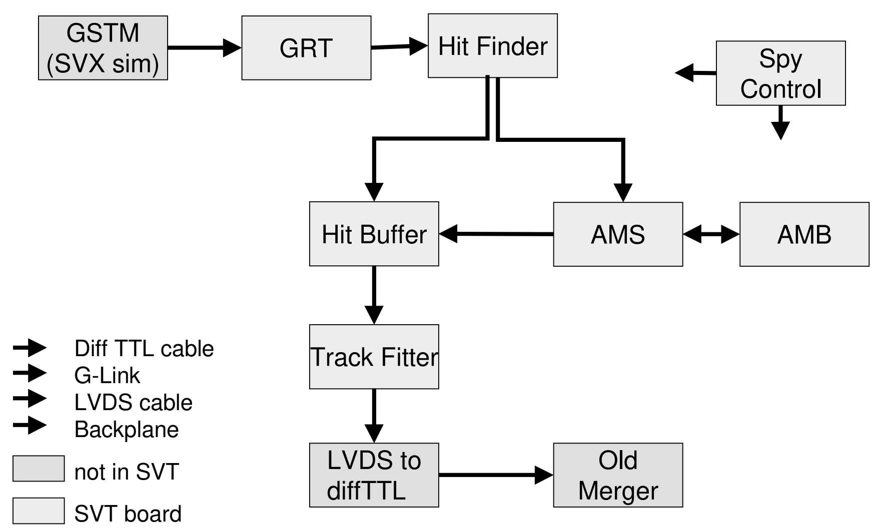
Hardware Integration Steps

- **Startup from home test in winter:**
  PISA: HB + AMS + AM + SC + MRG/(old MRG +lvds↔ttl)
  CHICAGO: GRT + HF + TF
- **Build up increasing complexity at B0:**
  - **I: HF+HB+AMS+AM+TF+SC (+ old MRG)**
  - **II: add GRT (brings in G-link issues)**
  - **III: add 1 new Merger**
  - **IV: add 2 new Mergers**
  - **V: add XTF**
  - **VI: add XTRP**
- Adding SVX readout "factorize", can be done at any time after II.
- Same for Level 2 i/f, though not always possible to plug it. Anyhow only needed to check communication.

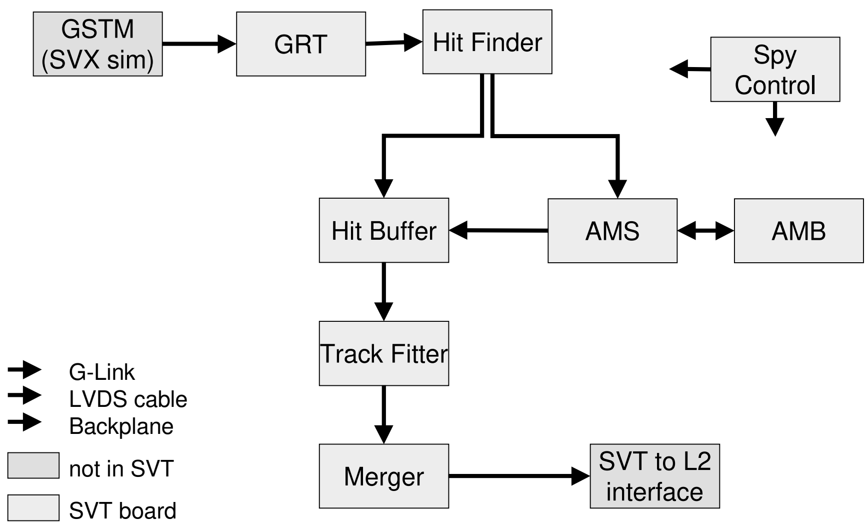
Hardware Configuration I (minimum new hardware) no G-link - brutalize HF - run fast, test slow



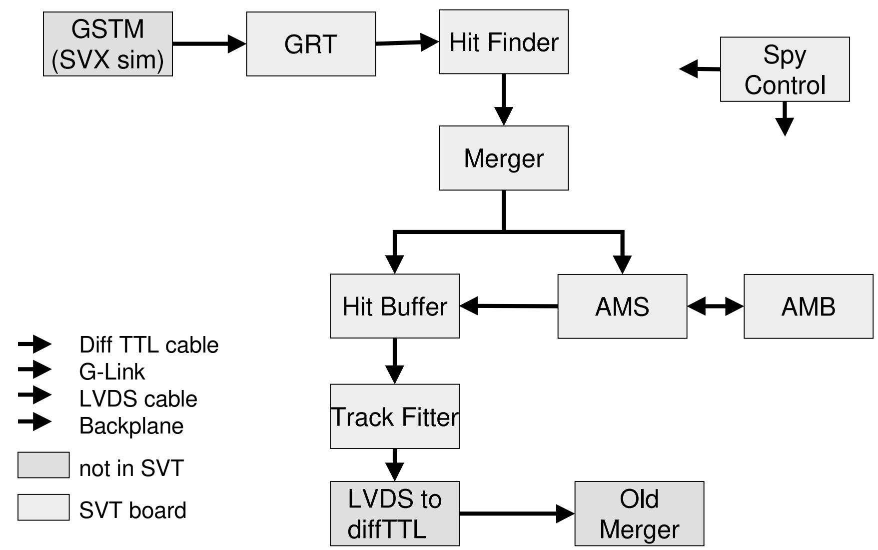
Hardware Configuration II (add G-link) brutalize HF - run fast, test slow



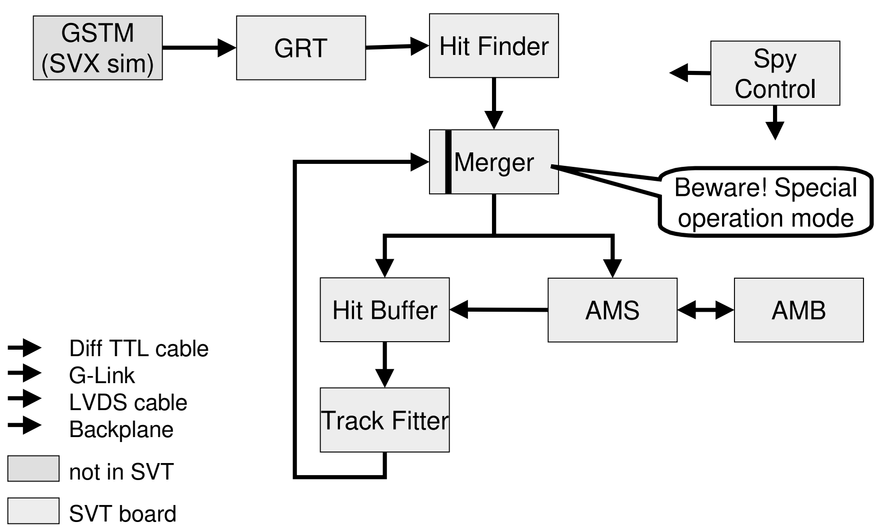
Hardware Configuration IIIa (add MRG: run & test fast) no XTRP - brutalize HF



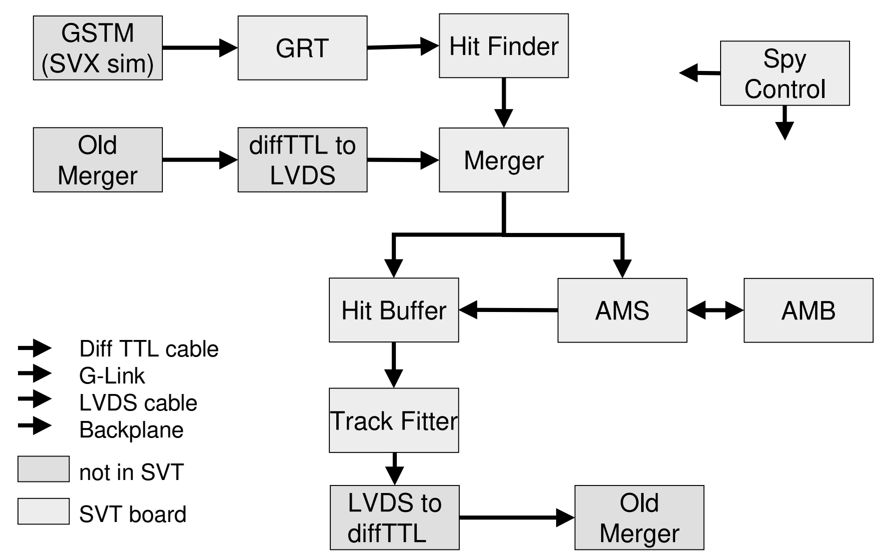
Hardware Configuration IIIb (use MRG for real HF ops) no XTRP - run fast, check slow



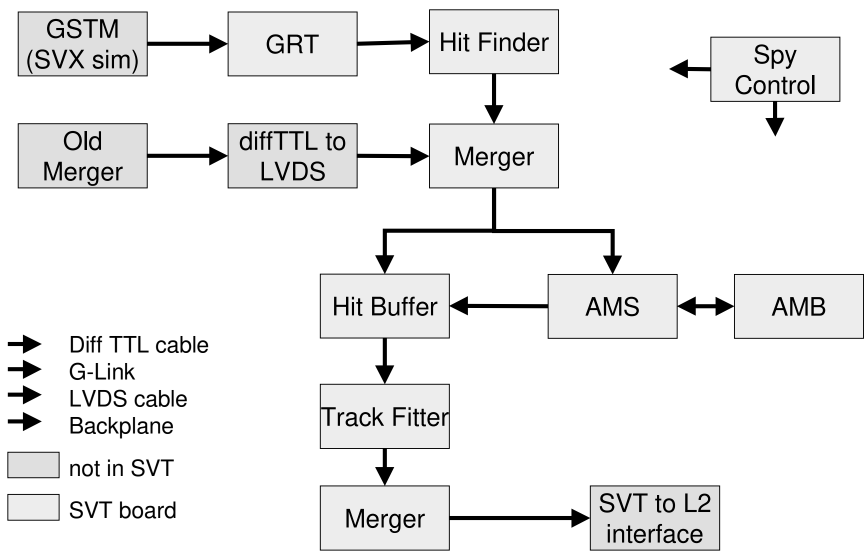
Hardware Configuration IIIc (fast test & check) no XTRP



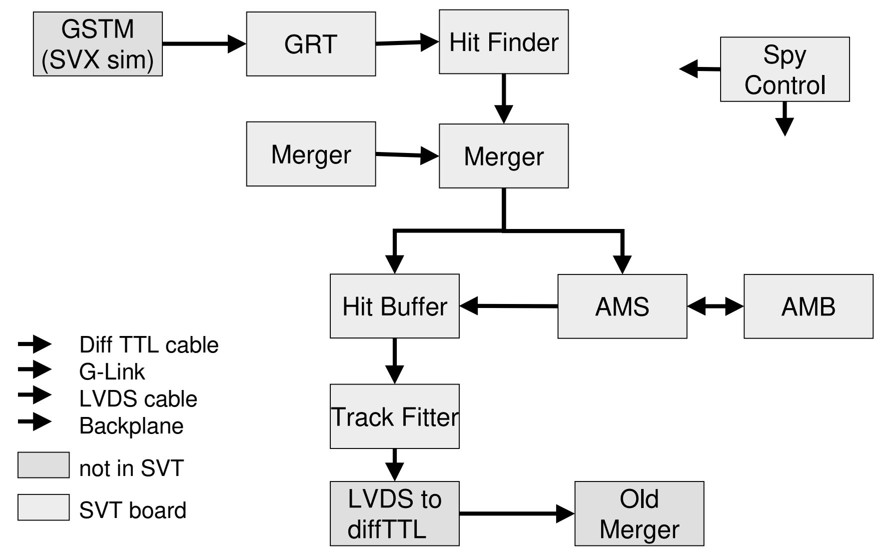
Hardware Configuration IIId (use old MRG as XTRP) only extensive tests - run slow



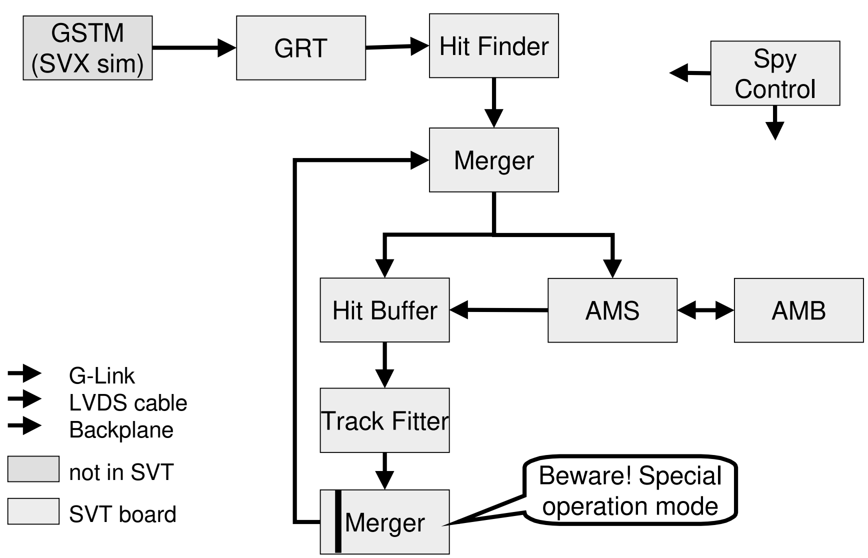
Hardware Configuration IVa (add 2nd Merger at end) only extensive tests - run slow



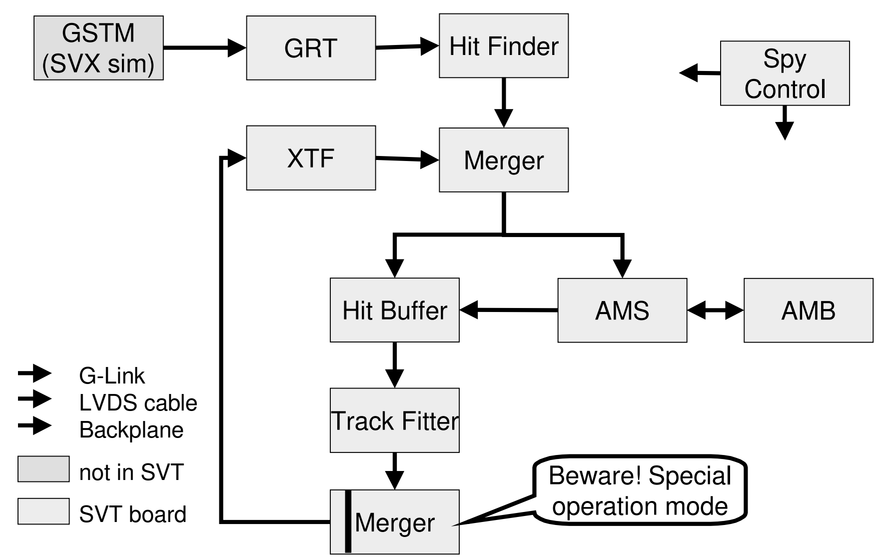
Hardware Configuration IVb (use Merger for XTRP) run fast, check slow only



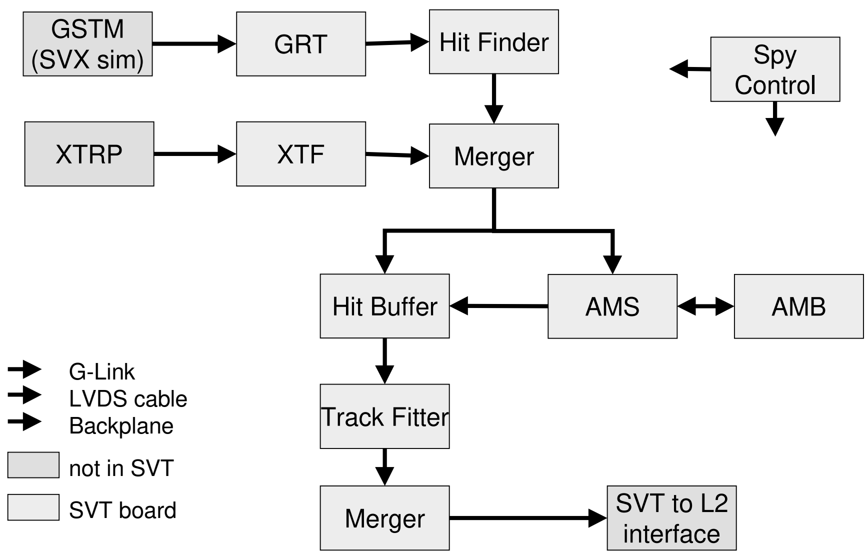
Hardware Configuration IVc (use end MRG as XTRP) can't add L2 i/f



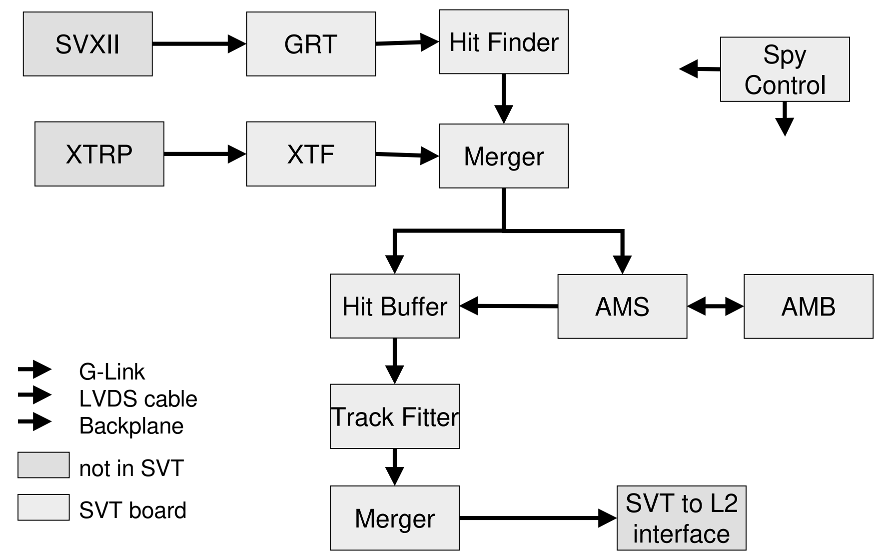
Hardware Configuration V (add XTF)



Hardware Configuration VI (add XTRP)



Hardware Configuration VII (add SVXII) final configuration



Software

- What we will bring to B0:
  - single board test-stand software small test program to test AMS-AM-HB system in non-continuos mode and at slow (test) rate (C version exists, being improved), may have version suited for fast test by then.
- What we will need for the test:
  - common framework for single board test based on CDFVME
  - larger test program to exercise full system
- What we will do once board production start
  - develop final on-line monitoring software (tomorrow's session)
  - develop final board test & diagnostic software for non-experts

Software we need to develop

- Single board test stand software based on CDFVME, ready to be integrated in a common package
- Global test program that generates SVX hits, reads back fitted tracks, and compare the two. Need also to define several data structures:
  - "detector" definition:
    - ☞ strips
    - ☞ hits (strip clusters)
    - ☞ SuperStrips (what to download in AMS and HB)
    - ☞ Patterns (what to download in AM)
    - ☞ fit constants
  - Maps (LUT's): SSMap (2), AMMap, HF clustering, TF constants
- Random test, ppbar-like test, error test … different data, different strategies, different test programs….
- Asynch. Monitoring via Spy Buffers

Needed Support from DAQ/FE group

- One SVT rack with cooling and power
- 2 SVT crates with
  - SVT custom P3 backplane
  - VME CPU connected to B0 LAN
  - TRACER (+ CLOCKSIM ?)
- 2 work places (color X-terminal, PC, Unix workstation… whatever) next to the rack (at least one here would be great) or very close to it
- Disk/CPU usage on B0 online computers:
  - limited CPU
  - user accounts as need
  - have large memories and data files, ~ 2GB of disk needed
- As this turns into development platform for online monitor, we also need suitable hardware (tomorrow)

How to get there

What do we still need:

- Boards: only AMB exists **now** in a suitable version
- Personnel
  - test will require 2 ~ 3 committed people (one each country!) full time plus expertise for the various boards
  - most software can be developed before
- Responsibilities:
  - software for each board (each board must have a name)
  - software coordinator: make sure soft. is consistent across boards
  - integration software developer (higher level stuff for test)
  - infrastructure coord. (make sure all needed hw is there)
  - SVXII DAQ expert/coordinator
  - test coordinator (decides what to test, when, how)