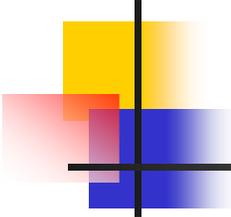


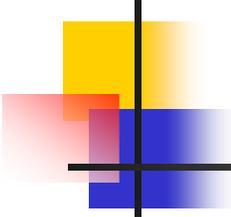
Teststand & diagnostic tools

- Teststand updates (summer 2002)
- In-situ diagnostic tool (summer 2001)



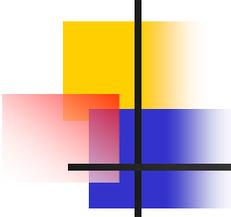
Teststand: PC

- Finally we have a PC near the teststand.
- The PC runs windows 2k, Linux available through VNC
- The CPU terminal is connected to the PC, and the PC runs VNC server -> console reboot from remote



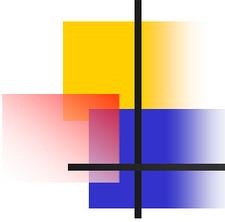
Teststand: randomtest

- Randomtest (RT) is a package written by Bill to test a complete 1/12 of SVT (wedge). Uses svtsim to check the TF output
- Franco's randomtest: modified version of Bill's:
 1. runs automatically on the teststand CPU (tstsvt1.fnal.gov) under VxWorks OS
 2. Uses the XTF system + 3 HF
 3. Writes a logfile
- How it works:
 1. Takes a mapset name as parameter. Initialize the boards with coldstart like functions.
 2. Generates a random event with one track as TF output
 3. Sends the "SVX" hits through one of the HF, cycling between the 3. Uses output spy buffer
 4. Sends the "XFT" word through XTFA + XTFB
 5. The event is processed from MRG + HB + AMS + AM + TF
 6. Reads back the TF output and check it against what expected from SVTSIM
 7. If an error is found it writes the SB content of some boards + expected data to a logfile



Teststand: randomtest 2

- Franco's version of RT is still completely compatible with Bill's: the new functions are activated only if it runs on the teststand and if a command line parameter is given -> the same executable is used to test the real SVT and the teststand
- The executable is part of the standard SVT library (svtvme) and the source is in CVS
- A set of command line parameters are defined to change RT behaviour, (e.g. to run without the XTF system)
- RT is automatically started at the CPU boot with standard parameters.
- A script is defined to stop it.
- RT ran continuously for 14 days , then Stefano stopped it ...



Teststand: RT logfile

LOGFILE for teststand continuous loop
date & time at startup: 02-09-09 & 00:34:24
Initialize ...
pass 0, fail 0, ifail 0, ofail 0
pass 10000, fail 0, ifail 0, ofail
pass 20000, fail 0, ifail 0, ofail 0
pass 30000, fail 0, ifail 0, ofail 0
pass 40000, fail 0, ifail 0, ofail 0

loop 43452:

tfout(D): Dn 15, Sn 8

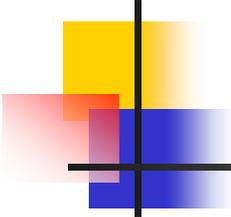
D 149f14 044895 003767 0bda7e 0a1208 062400 200317 149f14 044895 003767 0bda7e 0a1208 062400 200317 6010e8
S 149f14 044895 003767 0bda7e 0a1208 062400 200317 6011e8

....
....

pass 12999898, fail 102, ifail 0, ofail 102
pass 13009898, fail 102, ifail 0, ofail 102
pass 13019898, fail 102, ifail 0, ofail 102
pass 13029898, fail 102, ifail 0, ofail 102
pass 13039898, fail 102, ifail 0, ofail 102
pass 13049898, fail 102, ifail 0, ofail 102
pass 13059898, fail 102, ifail 0, ofail 102
pass 13069898, fail 102, ifail 0, ofail 102
pass 13079898, fail 102, ifail 0, ofail 102
pass 13089898, fail 102, ifail 0, ofail 102
pass 13099898, fail 102, ifail 0, ofail 102
pass 13109898, fail 102, ifail 0, ofail 102
pass 13119898, fail 102, ifail 0, ofail 102

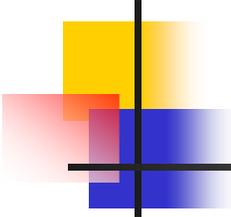


About one error/ 100000 : HB out is good.
TF IN is good. TF defective or SVTSIM fails ?
Do we have TF Spares ?



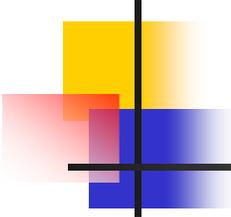
Teststand: RT doc

- A rough documentation of RT, with the description of parameters, customization files and how to stop is in:
www-cdf.fnal.gov/~spinella/teststand_randomtest.html



Diagnostic tools: a remind

- In a previous SVT workshop we defined the specs for a diagnostic tool, capable of testing suspicious SVT boards in their normal location without cable reshuffling.
- A document was written and circulated:
www-cdf.fnal.gov/~spinella/svttest.html

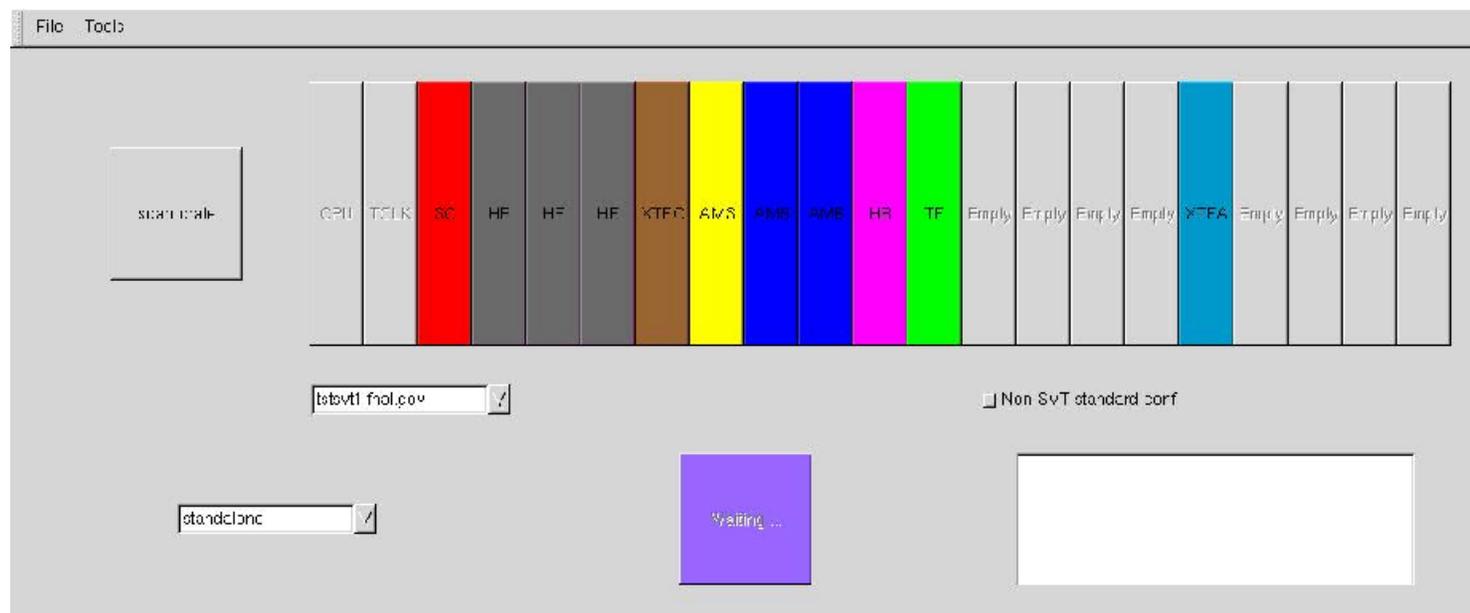


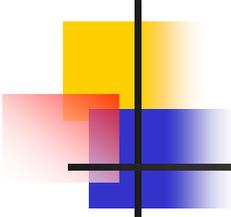
Diagnostic tools: basic concepts

- Board experts requested to write test programs for their board. Two kind of test programs needed:
 1. Standalone: test of the board without using other SVT boards as I/O
 2. Complete: previous test + send/receive data from other boards
- Test programs have to be defined as C functions. Names and parameters defined in the document.
- A graphical interface (C + GTK+) act as a “common” container for all these programs
- The GUI calls the test programs providing the necessary parameters, like a description of connections to the other boards
- At the end of the test a logfile is written, following some rules defined in the document.

Diagnostic tools: GUI

- The GUI code was written in summer 2001:

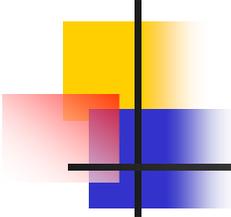




Diagnostic tools: GUI 2

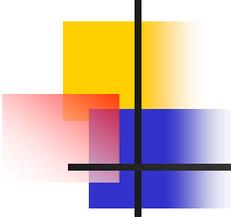
- At the moment the GUI includes only the test programs for the Spy Control

```
0100 SVT GUI VER 1.00
0105 logfile name: /cdf/people2/spinella/test_GUI/LOGFILES/SC/5/SC205020909111546.log
0110 current date & time: 02 09 09 & 11 15 46
0115 Board Under Test type: SC serial: 5
0120 Board Under Test crate: tstsvt1.fnal.gov slot: 3
0125 Test type: standalone
0130 Boards connected to the BUT:
0134 connector = 4 => crate = tstsvt1.fnal.gov slot = 4 conn = 0 type = HF
0135 === Spy control test func version 1.0
0140 BUT opened successfully
0145 Eprom checked OK
0146 register: SC SVTINIT force written OK
0147 register: SC SVTFREEZE force written OK
0150 register: SC Master Mode read OK
0155 register: SC LAST in CHAIN Mode read OK
...
...
...
```



Diagnostic tools: GUI 3

- The package include also a routine to test all the SVT cables, after reading a file with the cable description
- Written by a summer student in 2001, but never tested ...
- Probably needs some reworking.



Diagnostic tools: GUI 4

- **NEXT ?**

This document was created with Win2PDF available at <http://www.daneprairie.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.