Timing distribution / JTAG link

Mikihiko Nakao (KEK-IPNS)

mikihiko.nakao@kek.jp

January 26, 2011

Belle II Trigger/DAQ workshop, Beijing

Outline

1. **Recap of concept and design**
2. **Connection layout and number of modules**
3. **Status of prototype FTSW module**
4. **JTAG-link status**
5. **New boards (optical board, FTSW version 2)**
6. **Plan**

Role of TTD

- Deliver system clock and revolution signal to the front-end boards, COPPERs, and trigger system
- Deliver L1 trigger and related information if OK
- Provide a fast way of slow control, including synchronized initialization for run start
- Collect response for triggers from the front-end, to make sure the data-link is not overflowed
- Collect response also from COPPER to make sure the COPPER buffer is not overflowed
- Collect the data-taking status to be used for run control
- Attach the data-taking status summary to the event

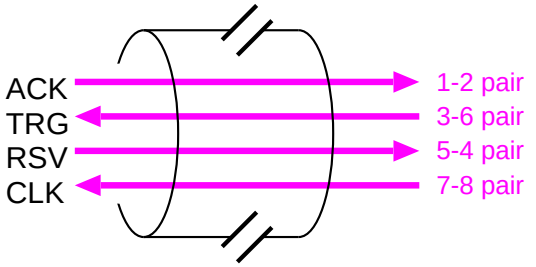
and in addition,

- Provide JTAG link to the front-end using the same hardware

FTSW concept

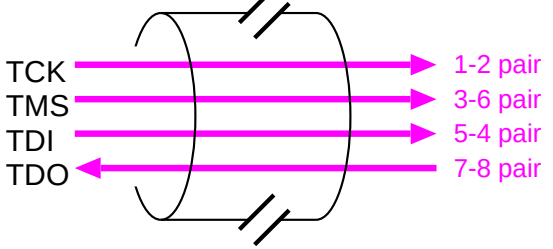
20110126 version

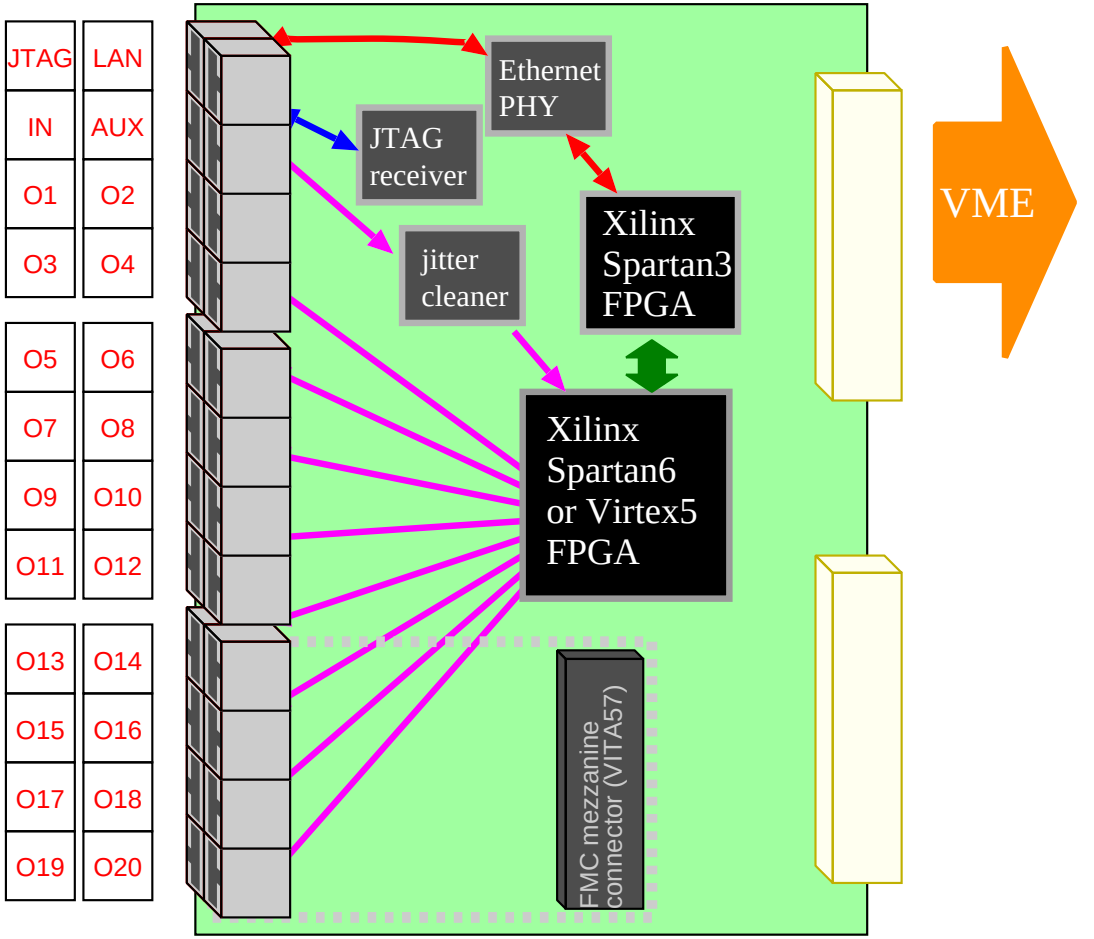
Timing signals over a LAN cable



CLK: 127 MHz (derived from 508 MHz RF clock)
TRG/ACK: serialized, 254 Mbps

JTAG-on-LVDS over a LAN cable





- **1-to-20 distribution in double-width 6U-VME for everything**
- **All connections through RJ-45**
- **TRG/ACK embedded in 8b10b encoded 254 Mbps serial link**

FTSW concept (optical)

20110126 version

Timing signals over one fibre pair

ACK
TRG/CLK

TRG/CLK: 254 Mbps serialized
127 Mbps clock is recovered from data
and goes through the jitter cleaner

Timing signals over two fibre pairs
(backup option)

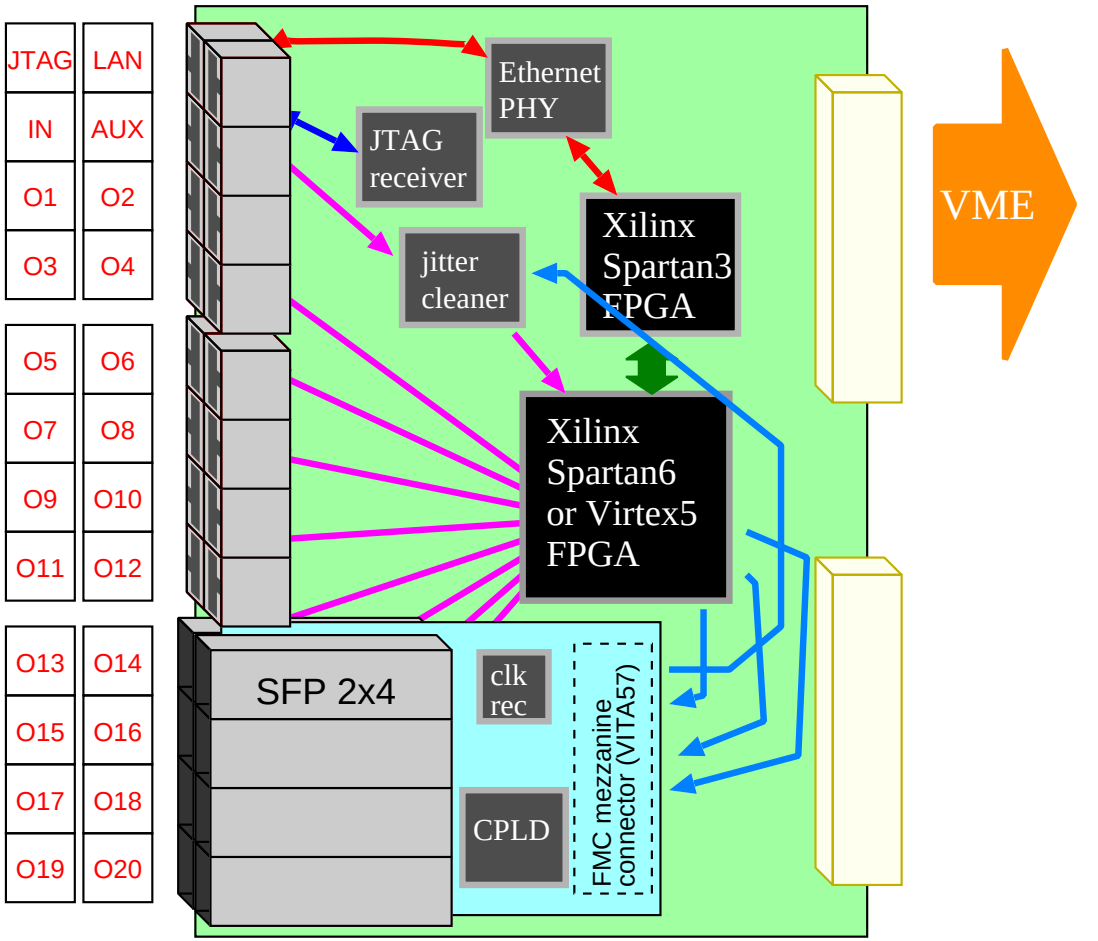
ACK
TRG

RSV
CLK

JTAG signals over one fibre pair

TDO
TCK/TMS/TDI

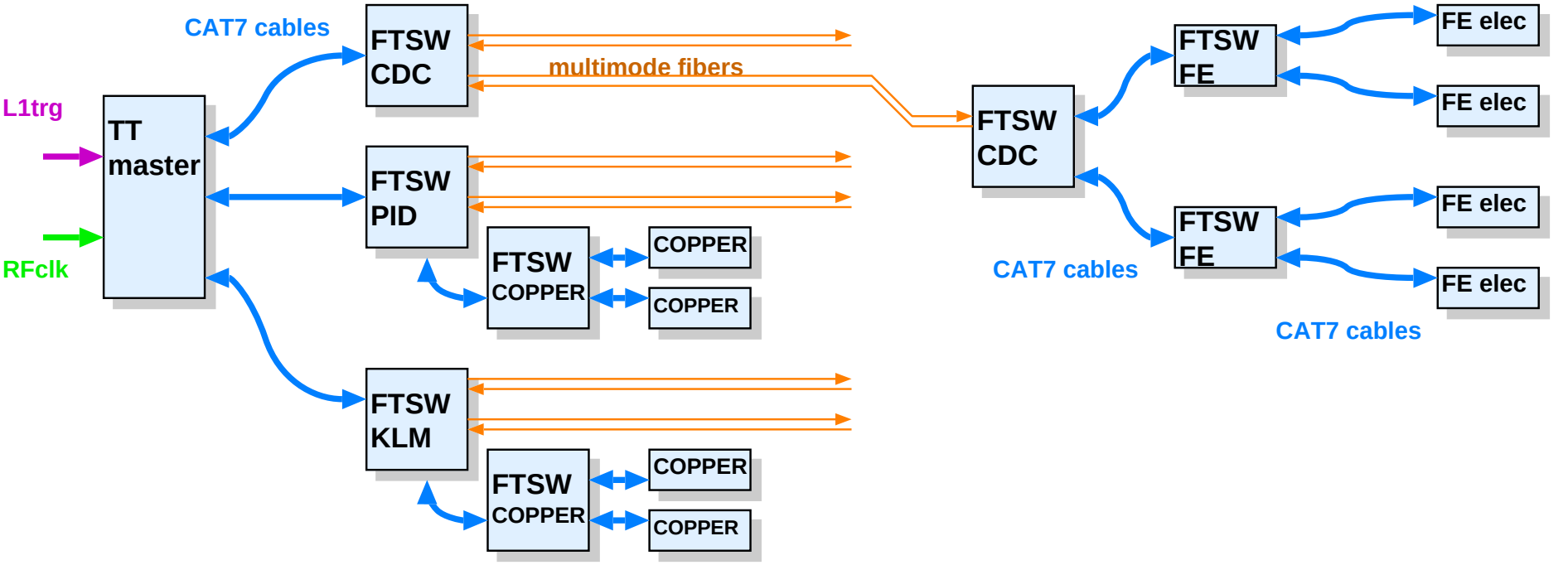
All signals are encoded in 254 Mbps serial data



- **8-port optical transceivers on an FMC daughter card**
- **1(cat7)-to-8(opt) or 1(opt)-to-12(cat7) distribution**
- **Clock recovery from serial data**

Connecton layout

- **E-hut and detector to be electrically isolated**
- **Timing link and JTAG link (share or separate cables)**
- **E-hut: master FTSW & one FTSW per each sub-detector sub-detector FTSW also connects to COPPER**
- **Detector: VME6U crate**
- **3 or 4 stages of FTSW modules to the frontend**



Number of modules and cables

- **CDC (302 links)**
  - **1-to-4 $\times$ [1-to-4 $\times$ 1-to-20] $\times$ 2**
  - **[VME6U crate with 9 FTSW] $\times$ 4, space needed**
  - **600 $\times$ CAT7 cables into detector ($\sim$130 cm^2, not measured yet)**
- **ECL (52 links)**
  - **1-to-6 $\times$ 1-to-12**
  - **FTSW in 6 of ECL collector crate, no extra space needed**
- **PXD (40), SVD (40), BPID (128), EPID (138), BKLM (86), EKLM (66)**
  - **Discussion item during this WS**
  - **Power, ground, forward/backward separation to be taken into account**

Prototype FTSW module

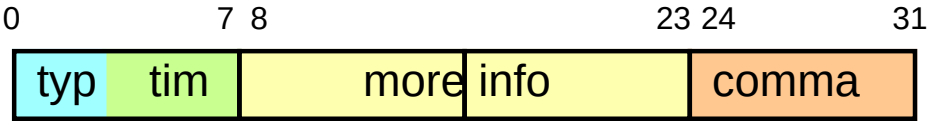
- **First 2 boards (full scale prototype) arrived on Sep 16, 2010**
- **3 more boards arrived on Dec 20, 2010**
- **Most of the functions have been tested**
- **Problems to be fixed in the second version**

FTSW functions

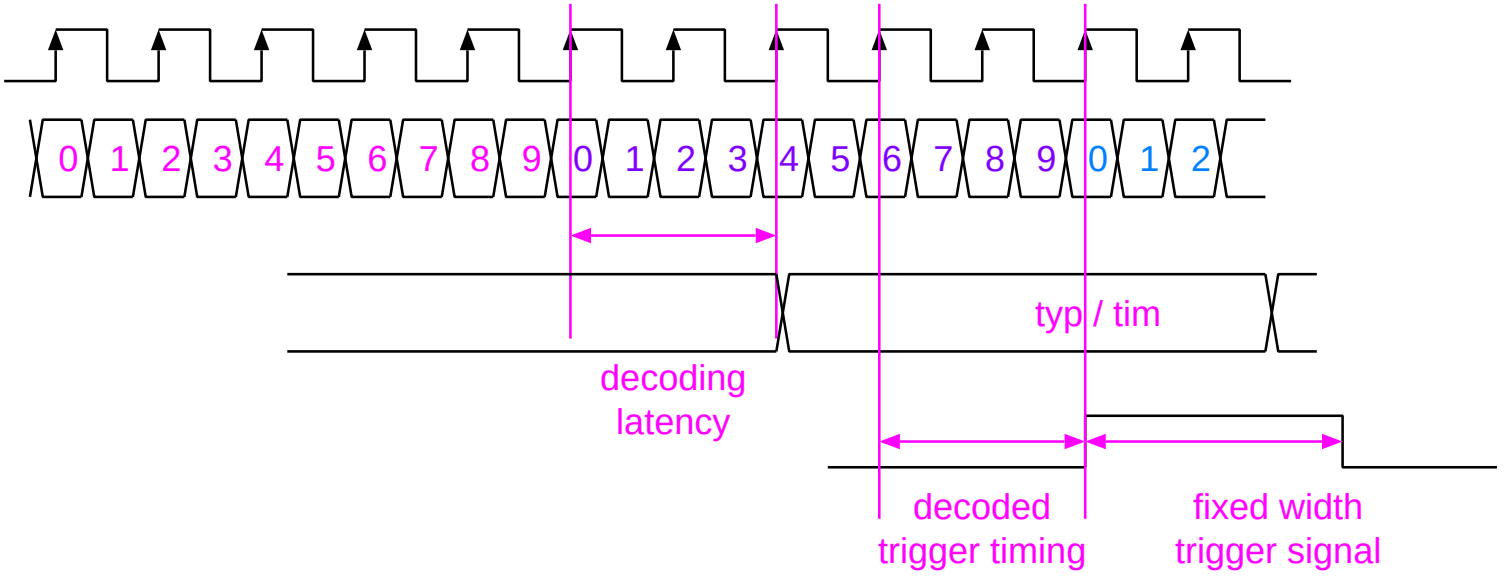
- **Power source — standard VME, 3.3V terminal port**
- **Board control — VME, (JTAG), (Ethernet)**
- **FPGA programming**
  — VME, on-board JTAG, JTAG-link, (configuration flash)
- **LVDS connections**
  — DC or AC coupled direct connection for 22 ports (both for input/output), except 1-pair for clock input
- **Serialization / Deserialization**
  — TRG / ACK signals, 21 ports each
- **Clock source**
  — on-board 125 MHz Xtal, input clock jitter cleaner
- **System monitor**
  — temperature, power, 10-bit ADC on FPGA

Almost all items have been tested. Gray items have some problem or not been tested.

Serialized trigger



- **254 Mbps, 32-bit word with 8b10b encode in 40-bit**
- **Trigger info (8-bit), general info (16-bit), comma (8-bit)**
- **64 words in one revolution (synchronized)**
  - **Revolution signal is decoded, reduced clocks are synchronized**
  - **1024-bit general info per turn, per each direction**
- **Timing is decoded before decoding the entire 32-bit**



Clock jitter performance

- **Setup**
  - **clock cleaner: FTSW**
  - **clock source: 125MHz from another FTSW**
  - **clock receiver: yet another FTSW**
  - **cable to scope: LEMO attached to 1.5m flat CAT7**
- **Jitter achieved with FTSW — 3.5 ps (with unrealistic setup)**
- **Noise and cross-talk sources**
  - **VME crate $\Rightarrow$ $\sim$ 5 ps jitter**
  - **Clock before jitter cleaner $\Rightarrow$ bad ($\sim$10ps)**
  - **With trigger signal on the CAT7 cable $\Rightarrow$ VERY BAD ($\sim$8ps with 3m cable, but $\sim$20ps with 15m cable)**
  - **Crosstalk on the cable, crosstalk on the FTSW board?**
- **Major revision of the FTSW will be made**
- **I suggest to add a jitter cleaner on the BPID frontend board**

Measure	P1:freq(C2)	P2:period(C2)
value	125.03 MHz	7.998 ns
mean	125.0019 MHz	7.99988 ns
min	124.69 MHz	7.981 ns
max	125.29 MHz	8.020 ns
sdev	54.6 kHz	3.49 ps
num	7.402160e+6	7.402160e+6
status	✔	✔

mmy text dummy text dummy text dummy text dummy
xt dummy text dummy text dummy text dummy text du

FTSW minor problems

(minor = solved or to be solved in the next version)

- **Power supply — enable pin usage corrected**
- **Clock receiver — AC coupled voltage reference, patched**
- **VME interface on Spartan3AN**
  — ported from TT-IO's 9500XL CPLD. A problem in FPGA programming solved by adjusting timing
- **VME sysreset pin — pulldown register added**
- **LED — some dark LEDs, will be fixed**
- **Jitter cleaner usage — solved, error in the datasheet?**
- **RJ-45 connector location**
  — aligned at the board edge, not at the front-panel face

Configuration flash

- **Configuration flash chip cannot be programmed at all**
  - **found in JTAG chain**
  - **but cannot blank check, erase, program, verify**
- **Suspected broken flash chip, but 3 more boards, produced at the second time, had the same problem**
- **Same chip is used in another Virtex-5 board, which works**
  - **Signals other than JTAG may affect, but all the signals checked**
  - **JTAG chain order is different: before or after V5**
- **Now trying to understand the problem**
  - **Replacing the chip**
  - **Test with partially mounted PC board**
  - **Need to fix before the next version**

AC coupling

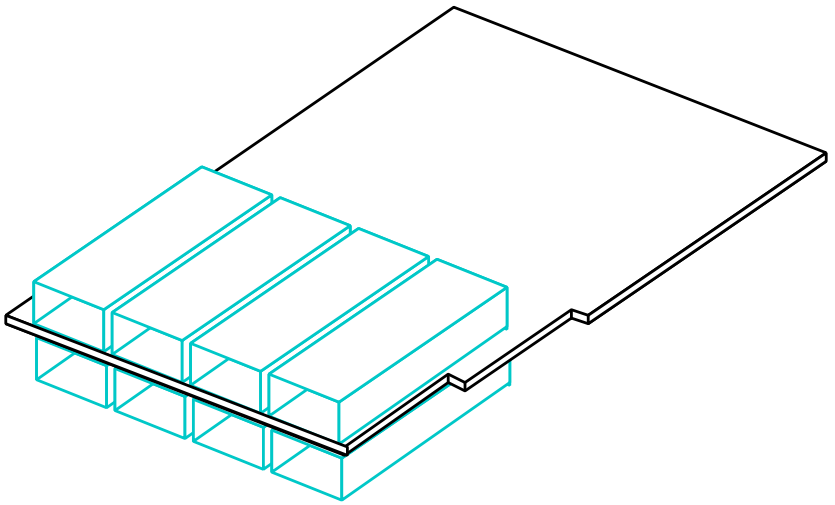
- **All timing-link signals are AC coupled at the receiver side (frontend board and input port of FTSW)**
  - **Ground-level shift does matter**
  - **No capacitors at the FTSW (no point to put at both ends)**
- **All JTAG-link signals are DC coupled (too slow, tested)**
  - **Ground-level shift does not matter**
  - **If JTAG-link input is taken from AUX port of FTSW, no need to prepare two types of FTSW (AC and DC)**

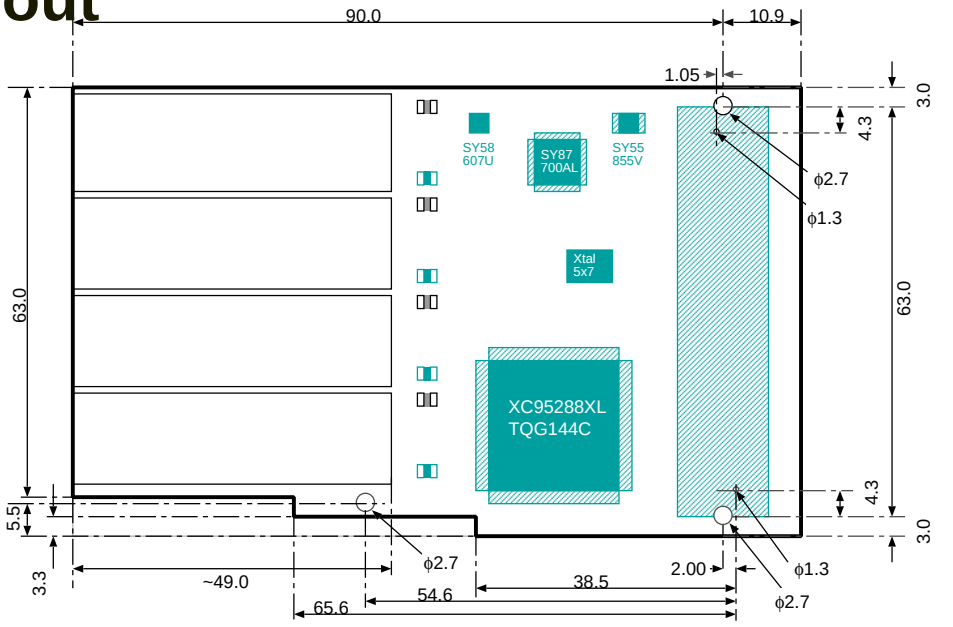
JTAG-link status

- **Confirmed that parallel JTAG programming (both using impact program and XSVF player) can be done**
  - **TCK, TMS, TDI is just fanned out**
  - **TDO is a logical-or of all inputs**
- **“Verify” from impact can be done in parallel, but the result is not clear. Need to think about it.**
- **So far no idea how to do “verify” using XSVF player, Y.Igarashi is working on it**
- **XSVF player is much slower than impact. Need to do tuning**

FTOP

- 8× generic SFP modules, directly connected to FPGA
- 1× Rx port has a clock recovery circuit
  - 15.902 MHz oscillator (508.877 / 32) ordered for 127 MHz
- On-board CPLD for SFP module monitoring (I^2C)
- 4Gbps SFP module was tested to be capable for 254 Mbps
  - Tested down to 64 Mbps
- Just submitted for board layout





FTSW version 2

- Fix all minor problems
- Understand and fix the configuration flash problem
- Better isolation of the clock signal on the PC board
  - 8-layer $\Rightarrow$ 10-layer
  - dedicated clock layer
  - dedicated Virtex5 I/O bank just for clock
  - clock signal is ground-guarded from other signals
- A few new features
  - 7.5V crate capability
  - Local TTD clock
  - Jitter cleaning of the recovered clock from optical link
- Will be built by the end of March
- Will be ready for small production for sub-detectors

Plan

- Board development is finalizing — FTSW2, FTOP
- Interface to the RF clock
- Interface to GDL
- More firmware work from next FY
  - Basic firmware for remote users
  - Timing core to be integrated into frontend
  - Trigger distribution
  - Busy collection and flow control
  - Connection to COPPERs
  - Slow control
- Start mass production