1. White Rabbit Switch Testing Project

2. White Rabbit Switch Tests
   - Pre-Test: Basic visual and electrical tests.
   - The Bootstrap Test Procedure
   - Actual Testing

3. White Rabbit Compliance and Performance Tests

4. Hardware Testing Framework

5. What’s Next

6. Wiki of the Project
The project gathers the methodology and methods for testing the White Rabbit Switch.

**WR Switch Production Tests**
- Manufacturing quality control of the WR Switch
- Avoid that hardware problems manifest themselves as software issues

**Benchmark and Compliance Tests**
- Compliance with the protocol stack: GbE, PTP, WR PTP etc..
- Assessment of the performance: throughput, frame latency etc..

**Other Activities in the Project, Testing Hw Framework**
- Based on the project ”Production Test Suite”
- Software that automatizes the production test procedure
Quality procedures already done and documented.

⇒ **Goal:** detection of soldering/mouting problems

- FlyingProve done at the assembly company
- Visual inspection of the PCBs (SCB & Mini-BP)
- Electrical inspection of the PCBs
The Bootstrap Test Procedure

The test of the switch has been intended to be as flat as possible, this means that we intend to test a component with the least dependency possible of other components.

However, to test a component (FGPA, NAND, ...) we need to have at least the following working:

- ARM (detected during the flashing stage)
- DDR (test during the flashing stage)
- Linux load using Ethernet (TFTP/NFS)\(^1\)

\(^1\)Can be load directly using DDR in case Ethernet fail
The testing procedure is a set of scripts:

- 001-MD5_Checking.sh: Check MD5 of needed files
- 002-USB.sh: Check the USB (Not implemented)
- 003-FPGA_Bridge.sh: CPU (EB1) ↔ FPGA (WB)
- 004-LED_Checking.sh: LED of CPU & miniBP
- 005-FAN_Checking.sh: FAN on/off and PWM speed
- 006-FPGA_QDRIIStress.sh: Check & Stress the QDRII.
- 007-NF_MT29F4G16.sh: NAND flash partial R/W (~ 20m)
- 008-DF_AT45DB642.sh: Dataflash all position (~ 6m).
- 009-FPGA_TempRetrieving.sh: Retrieve the temperature.
- 010-WRS_SFPs.sh: Use standard bitstream to check SFP transfers.
- 011-Flashing.sh: Flash with the latest stable firmware
To perform these steps we have create some generic tools.

**testing.sh**

Main script called to generate proper log information:
- S/N of the board
- Timestamp of each steps
- Error failure codes
- Logging history

**shw_tool.c**

Accessing to libswitchhw library to test various components:
- LEDs (GPIO, wishbone I2C)
- FANs (PWN fan, GPIO)
- Temperatures (wishbone I2C)
Below an example of the log history for the board 014 (v3.2)

/tftpboot/rootfs/alpha-pts/logs/output-032014_history.log

Fri Oct 12 12:24:08 CET 2012

001 > OK (00’02)
002 > OK (00’00)
003 > OK (00’10)
004 > OK (00’52)
005 > OK (00’52)
006 > OK (00’13)
007 > ERROR ( Sub0:OK Sub1:OK Sub2:OK Sub3:OK Sub4:50% Sub5:OK Sub6:OK) (22’40)
008 > OK (02’47)
009 > OK (00’14)
010 > OK (01’20)
011 > OK (04’51)


007 > OK (22’24)
Future Work

Create small HDL to check only a component, not a functionality.
- I2C sensor
- SFP transfers
- PLL frequency

Conclusion

By doing these tests we **improve the quality** of the product:
- Checking the components
- Checking the connection
- Improve development: Check if a bug occurs for new functionality or for a specific component.
- Understanding better the sw/gw
  ⇒ **Better client support**
Integrasy has done the following compliance tests:

- VLAN support
- MRP and MVRP
- Remote Management Interface

CERN (Maciej Limpinsky) has ”tortured” the switch and measured the synchronization performance between 0 and 50 degrees.

**Missing Tests**

- Throughput
- Latency
- Frame Loss Rate
- System Recovery
- Synchronization performance with multicast traffic
- etc...
Do we need this tests?

- This test could detect new issues in the v3 gateware/firmware/design
- Do we need a standardized way of verifying that a combination hw+firmware achieves the White Rabbit features?

Do we need special hw and which one?

- Commercial solutions cover partly our tests
- The community could hardly afford to buy/rent a commercial solutions. For the manufacture it could be seen as an investment.
Proposal

- Use the WR Switch hw platform for developing a compliance and performance tester. It could be also used as a network debugger in production systems.

- We could reuse part of the current gateware/firmware of the switch, but that wouldn’t be enough, it is a new development...
Architecture

Client/Server model, the server steers the tests and gathers the log information of one or several DUT clients.

Features

- Communication with freestanding or hosted hardware
- One-time configuration
- Guidance and test information during the procedure
- Flexible order of the execution of the tests

Goal

Be able to provide a HW Testing Framework not only for the WR Switch, but also for all the WR Node form factors.
What’s Next?

**WR Production Test**
- Cosmetic changes
- Integration with hw Testing Framework
- Development of more tests?

**WR Compliance and Test-Bencher**
- We should start working on it
- Who wants to ”chip in”?

**Hw Testing Framework**
- Release by February a version for the WR Switch
- Keep working!
Wiki of the Project

Information about this Project in the OHWR Wiki, so far there is:

- WR Production Tests and How To
- WR.v2 802.1Q Functional Test Cases Report
- Torture Report
- Testing Hardware Framework, not released yet
- Test Methodology Doc
- Proposal for the WR tester.