**EUROPEAN ORGANISATION FOR NUCLEAR RESEARCH**
**ORGANISATION EUROPEENNE POUR LA RECHERCHE NUCLEAIRE**

Tomasz WŁOSTOWSKI
CERN AB-CO-HT
Tel : +41 (0)22 76 79262
Bat : 864-1-A07

# *White Rabbit* switch

## Preliminary functional specification

Geneva, Switzerland
14 October 2008

## 1. Introduction

*White Rabbit* switch is the key component in new CERN timing system, allowing for multiplexing of high-precision timing and control data in single fiber connection. Unlike most of current timing electronics, its standalone device (not a VME card). Its main features are:

- gigabit Ethernet fiber optic switch with 8 downlink ports and 1 or 2 uplink ports and boundary clock (timing repeater) capability
- compatible with IEEE802.1 specification (e.g. standard Ethernet)
- synchronous operation for low-latency data and high precision timing transmission
- timing protocol (eg. IEEE1588) support by hardware packet inspection and timestamping
- less than 0.5 ns of master-to-slave timing skew and < 1 ps RMS jitter
- support of both single (WDM) and two-fiber links
- transparent, plug&play timing transmission (no configuration/calibration required)
- remote configuration and management

## 2. Inputs and outputs

| **Name** | **Type** | **Count** |
|---|---|---|
| **Downlink ports** | **SFP module socket** | **8** |
| Standard fiber-optic GbE ports for connecting timing slaves or lower-layer switches. | | |
| **Uplink ports** | **SFP module socket** | **1 or 2** |
| Standard fiber-optic GbE ports for connecting to timing master or upper-layer switch. The only difference between uplink and downlink ports is the presence of phase shifter and clock multiplexer. This makes it possible to measure link phase shift and use recovered RX clock as a master clock for downlink ports. There may be one or more uplink ports. Using multiple uplink ports can make timing network fault-tolerant. | | |
| **Exteral clock inputs** | **LEMO connector, PECL levels** | **1** |
| External 10 MHz / 1PPS inputs used to supply reference clock for the whole network when the switch acts as system timing master | | |
| **Local digital outputs** | **Multi-pin (e.g. DB25) connector, TTL/LVTTL levels** | **16** |
| General-purpose digital outputs capable of producing digital sequences (waveforms, single pulses) synchronized with master clock. May be used for testing purposes or triggering devices without using additional timing receivers. | | |
| **RS232 port** | **DB9 female connector** | **1** |
| Local RS232 port with „dumb terminal” for initial configuration/service purposes | | |

## 3. System diagram

Simplified block diagram of the switch is shown on **figure 1**. It can be separated into following blocks:

1. **Uplink and downlink CPBs** (Common Port Logic Blocks). Structure of CPB is shown on **figure 2**. Each CPB consists of:
   - simple GbE Media Access Controller with hardware timestamping which acts as frame parser/composer
   - packet inspector which detects packet sources/destinations/types and tells about them to routing controller
   - scheduler which performs data ordering and decides when to send awaiting standard frames. It also maintains high-priority traffic.
   - FIFOs, which are used to buffer incoming/outgoing data. For high-priority data FIFOs are bypassed.
2. **Routing busses,** interconnecting CPBs. There are 5 16-bit bidirectional busses clocked at 62.5 MHz (half byte rate). 4 busses are used to route standard-priority data between downlink CPBs only, and one bus (marked green) is used to route standard data from uplink to downlink ports or to route broadcast high-priority data. Busses are controlled by special arbiter module which assigns them to CPBs by setting up proper MUXes.
3. **Routing controller** decides to which port incoming frame shall be sent. It associates source addresses identified by packet inspector with appropriate switch ports. Also it maintains high-priority mode operation by pausing transmission of standard frames when high-priority frame arrives.
4. **Clocking system** is crucial for precise timing transmission. The basic idea of Sync-E is to propagate master clock embedded in downlink data stream from System Timing Master to all devices in the network. In our switch master 125 MHz clock is recovered by CDR circuit in uplink port's PHY, filtered by DPLL and then fed to downlink ports' PHYs TX clock inputs. DPLL also has ability to sustain stable master clock for short time (eg. 1 millisecond) when PHY CDR de-locks itself due to transmission error. The uplink port TX clock can be phase shifted, allowing for very precise phase shift measurement (refer to *Fiber delay compensation* document for details). Master clock can be also provided by external reference (e.g. GPS or cesium clock). Every switch contains local clock reconstruction unit, which can generate up to 16 synchronized programmable signals (e.g. clocks or trigger pulses) as well as 8 kHz Sync-E time slot clock and other necessary internal timing signals.
5. **CPU** handles all high-level stuff required for switch operation. Please note that routing is done entirely in FPGA hardware, and CPU is used only for management and routing tables optimization. The main tasks of CPU are:
   - scanning the network structure and calculating optimal routing table entries by using RSTP (Rapid Spanning Tree Protocol)
   - measurement and compensation of link delays for timing transmission
   - remote switch management via SNMP protocol.
   - clock source switching (to secondary uplink port) in case of failure of primary uplink port

The CPU is interfaced with network infrastructure via separate Ethernet controller built into routing bus arbiter. This controller has assigned separate MAC address and its visible from the whole network like typical Ethernet card.

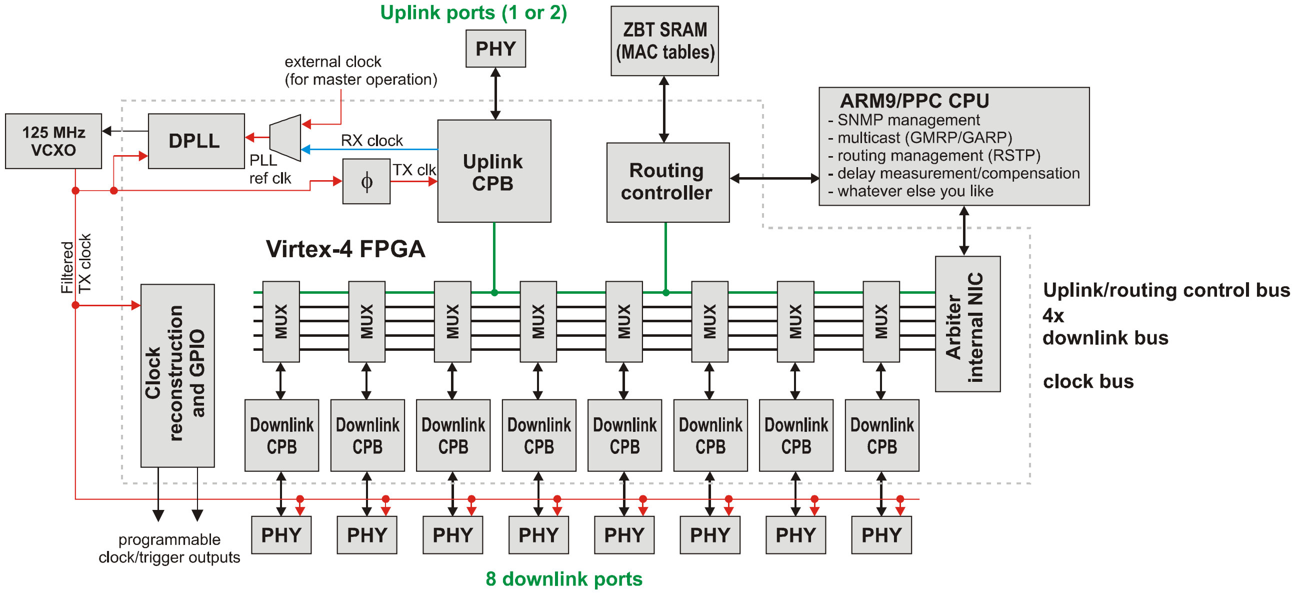

**Fig. 1.** Block diagram of the switch

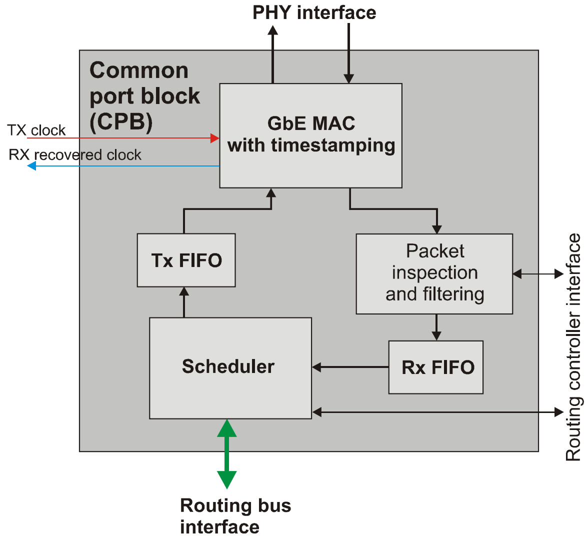

**Fig. 2.** Structure of Common Port Logic Block

## 4. Device operation

The goal of this project is to provide network layer comatible with Ethernet, but also providing deterministic timing model and low transmission latency. To achieve it we decided to introduce two traffic classes in the network:

- **standard traffic (SP) –** consisting of normal unicast or multicast Ethernet frames. For this kind of traffic, switches operate in store-then-transmit mode (like standard Ethernet gear). In case of collision, data is buffered. This type of traffic is non-deterministic.
- **high-priority traffic (HP)** – broadcast ethernet frames with unique value of *EtherType* field (to allow network gear to detect such frames reliably). Upon reception of such frame, switch immediately pauses all standard traffic transmissions and broadcasts high-priority packet (introducing only small and **constant** delay). To prevent collisions, devices are allowed to send such frames only in response to the master request.

### HP and SP packet handling

As it was said before, HP frames have absolute priority over SP frames. Typical situation when HP frame arrives during transmission of SP frame is illustrated on **figure 3**.

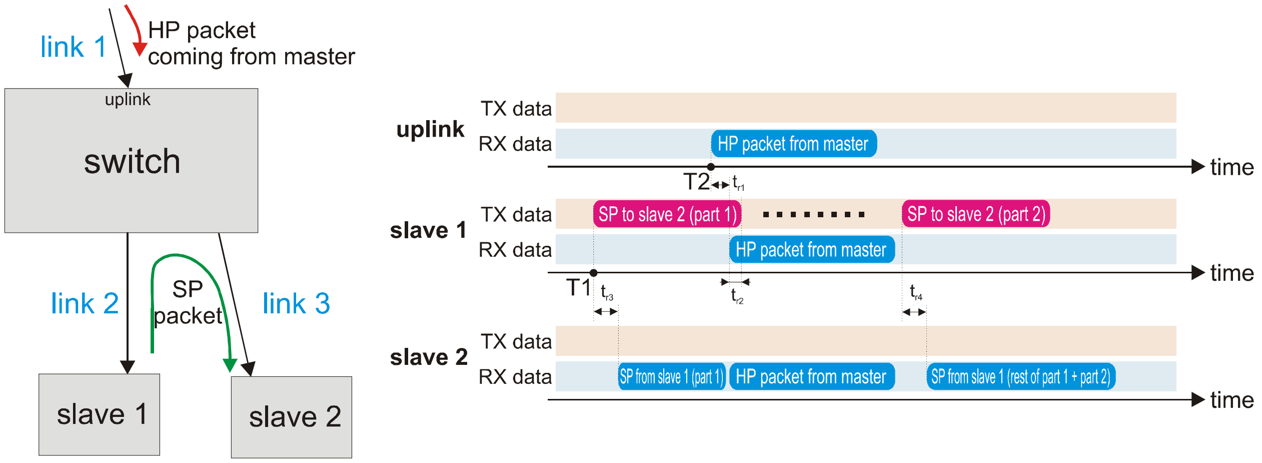

**Fig. 3.** Routing HP packet

Below is the order of data transfers:

1. Slave 1 starts transmitting unicast SP packet to slave 2 at time T1.
2. After nondeterministic time $t_{r3}$ SP packet reaches slave 2.
3. At time T2 HP packet is detected by uplink port in the switch. Switch immediately pauses transmission of SP packet between slaves and starts broadcasting HP packet
4. After deterministic time $t_{r1}$ slave 1 starts receiving HP packet data.
5. Upon detection of HP header (after time $t_{r2}$) slave 1 pauses transmission of SP packet to prevent from buffer overflow in switch. Because switch paused transmission of SP before slave 1 did it, some data of SP is buffered by FIFO in switch.
6. HP packet is received by both slaves
7. When HP packet is received, slave 1 continues transmission of SP packet
8. After nondeterministic time $t_{r4}$, slave 2 receives the rest of the SP frame (partially stored in switch FIFO).

The only requirement for this method is that devices are never allowed to talk in HP mode independently. A slave device may send HP packet **only** in response to request sent by master. Similar scheme is used in Powerlink or WorldFIP.

### Time-critical traffic

There are HP frames (e.g. timing data) which need to be delivered to all slaves before some certain point in time. As the collisions are not allowed in HP mode, each device must always wait for all devices to receive the current HP frame before transmitting next frame. This delay can be:

- fixed to value greater than maximal possible delay between any devices in network. This scheme is simple, but it can cause bandwidth loss.
- computed dynamically from delay compensation measurements. In this mode, all switches are regularly reporting measured link delays to the master. The master computes maximal routing delay (in whole network) and broadcasts it to all devices in the network. This scheme ensures best bandwith allocation.

As the device works on 2nd network layer (MAC bridge), no data loops in network should exist.They are automatically detected using RSTP algorithm (refer to 802.1 specification for details) and removed by disabling ports which create a loop. These ports can be automatically enabled to create alternate data path in case of primary link failure.

## Timing routing

As we mentioned before, timing is multiplexed with data in the same medium. Proposed timing network topology is shown on **figure 4.** Primary timing connections are red, secondary ones (alternate) are green. In such network uplink ports never share the same timing path - if one of switches or masters fails, network still remains fully functional. Because the plug&play version of the switch uses only WDM SFP modules (single-fiber), such network can be built using existing non-multiplexed fibers (no additional cabling is required).

For the delay compensation, we use method described in *Fiber delay compensation* report. For delay measurement, PTP version 2 protocol can be used, as (according to preliminary IEEE1588 v2 spec) it can work transparently on 2nd network layer and has TLV fields (type-length-value) which can be used to measure phase shifts.

Clocks from both uplink ports are received and compensated simultaneously, allowing for seamless switching. Therefore every switch has in-phase master clock and acts as a boundary clock (timing repeater) for lower-layer switches or receivers. In proposed network topology, slave devices synchronize themselves only to the closest switch. This approach greatly simplifies the hardware (because we can use slow DCM-based measurement procedure) and reduces network/timing master load. As the timestamps are recorded before buffering we can use SP packets for delay compensation, saving precious HP bandwidth. In contrary to data routing, no timing loops are possible as we are using dedicated uplink ports for receiving timing.

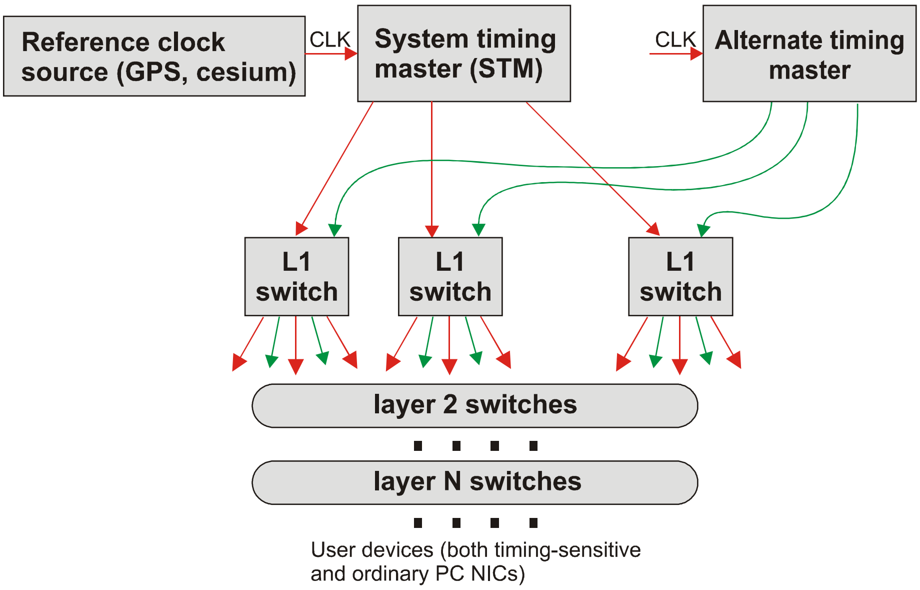
Reference clock source (GPS, cesium)
CLK
System timing master (STM)
CLK
Alternate timing master
L1 switch
L1 switch
L1 switch
layer 2 switches
layer N switches
User devices (both timing-sensitive and ordinary PC NICs)

**Figure 4.** Timing network topology

## Transmission reliability

Gigabit interfaces are susceptible to random transmission errors. As the switch operates on 2nd layer (MAC bridge), there is no guarantee that packets will reach its destination without errors, even in HP mode. Therefore some error correction algorithm should be implemented for important control/timing messages.

Let's assume that master wants to send 1024-byte broadcast HP frame. Because we can't use handshaking for every slave (it costs too much bandwidth and time), we must be sure that even if frame is corrupted, the original data can be recovered. Proposed algorithm for error detection/correction is described below.

1. We divide the frame into 128-byte blocks. For 1024-byte frame there will be 8 blocks. Lets call them *A1..A8.*

2. We calculate 2*8 equations by XORing randomly chosen blocks from *A1...A8*. All equations must be different. For example:
   *X1 = A1 xor A3 xor A7 xor A8*
   *X2 = A2 xor A4 xor A6 xor A8*
   *X16 = ...*
3. We transmit *X1..X16* in subsequent HP frames, each frame contains:
   - 32-bit frame ID (allowing to identify to which frame each block belongs)
   - 32-bit integer with equation coefficients. For example, value of *100101* means that *Xn = A1 xor A4 xor A6.*
   - payload (*Xn*)
   - MD5/SHA1 hash value

With this algorithm, receiver can reconstruct original frame having any 8 of 16 equations. This means that 50% randomly chosen frames may be corrupted and we will be still able to reconstruct original data. The method does not require high processing power and its easy to implement in hardware. Also, the *X1..X16* frames can be sent subsequently (without waiting for recepion of each frame), so the bandwidth loss caused by data redundancy is minimal.