FIRMWARE DEVELOPMENT FOR SNS NEW TIMING MASTER*

R. Stefanic, J. Dedic, Cosylab, Ljubljana, Slovenia
D. Curry, D. H. Thompson, ORNL RAD, Oak Ridge, Tennessee, U.S.A.

Abstract

Implementation of a timing system master device is a complicated task, since a lot of details have to be taken into account even once the architecture decisions have been laid down. At SNS/ORNL [1] timing master controller is being upgraded in collaboration with Cosylab d.d and this paper focuses on some details of its implementation.

New timing system master device is based on agile FPGA circuitry and the main focus of this paper is its firmware implementation. Provided are implementation details for event distribution supporting multiple event sources and priorities.

Discussed are mechanisms, ensuring deterministic behaviour, different methods of encoding that have been employed, and host-independent distribution of time stamp frames. The concept of the super-cycle is explained and its implementation is laid down. Taken into account that implementation for such a complex device involves extensive testing, paper provides insight into verification it was applied. Advantages of the SystemC [2] based test-benches over traditional VHDL-only verification are discussed.

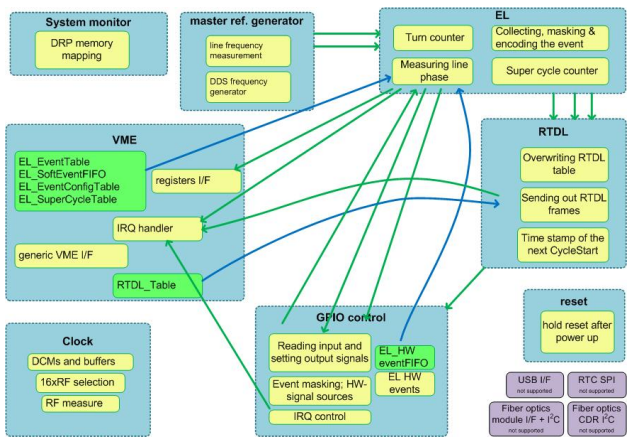

Figure 1: Timing system master architecture.

INTRODUCTION AND GENERAL DESCRIPTION

New timing system master device is an upgrade of the current device, based on outdated BNL (Brookhaven National Laboratory) hardware, used in the ORNL/SNS.

The SNS timing system consists of two data transmission links, which are distributed to all accelerator systems; Event Link and Real Time Data Link.

* Work supported by Oak Ridge National Laboratory for UT-Battelle, LLC, under contract DE-AC05-00OR22725 for the U.S. Department of Energy.

The Event Link (EL) transmits 8-bit timing event enumerations that define the SNS Machine Cycle. Events sent over the Event Link are synchronised with the accumulator RF frequency and are bi-phase encoded so timing receivers can lock their actions to the accumulator revolution period.

The Real Time Data Link (RTDL) transmits a series of 24-bit data frames, prior to the beginning of a machine cycle. Data frames contain information about the next machine cycle such as time of day - timestamps, ring revolution period, mode of operation, etc.

The device is based on a custom designed VME board with multiple interfaces – VME, RTDL, EL and GPIO. It is designed around Xilinx Virtex5 [3] FPGA integrated circuit which provides all of the functionality specified in the design.

GENERAL DESIGN OVERVIEW AND MODULES DESCRIPTION

New timing system master FPGA design consists out of the following major modules (Figure 1):

- Clock module
- GPIO module
- Master reference generator module
- Real Time Data Link (RTDL) module
- Event Link module (EL)

Other minor modules are:

- System monitor module
- I^2C module

Clock Module

Clock module is responsible for input clock signal conditioning, frequency stabilization and signal distribution to all other modules.

Four independent input clock signals from different sources, with different precision are available.

Major task of the module is to provide a stable internal RF clock signal comprised out of two input sources – primary RF clock from an external high precision clock source, and backup on-board clock source. In case that primary clock source is no longer available, the implemented logic switches its internally used RF clock line to the backup clock source, while trying not to disrupt the signal in the course of transition.

Additional functionality of the clock module is to provide period measurements for the two RF input sources, where period duration is measured in picoseconds.

GPIO Module

GPIO module mostly handles input and output signals and offers VME module to control various settings for signal handling and redistribution.

16 HW-signal sources of interrupts are provided, where all of them have configurable edge sensitivity, can be enabled or disabled and certain dedicated output signals can be controlled directly.

Module provides an option for events to be infiltrated into Event Link queue. For this purpose, a dedicated FIFO is implemented, to hold information about which events were triggered during the course of one machine cycle. FIFO content is then processed and redistributed by EL module.

Master Reference Generator Module

The purpose of the Master Reference Generator is to monitor the AC power line frequency and with the help from software, produce a stable reference pulse that does not change faster than the T0 neutron choppers can keep up with. Reference pulse also must not stray too far from the actual power line zero crossing. Module provides raw line-period and phase measurements and offers programmable frequency synthesis sub-module, where frequency correction step is as small as 50 μHz.

Host (VME, SW) is responsible for all calculations required to track line frequency and phase. Given the measurement of the line period and phase, host can calculate the desired target frequency which is achieved using direct digital synthesis (DDS) concept implemented in the FPGA.

Real Time Data Link

The SNS Real Time Data Link (RTDL) is modeled after the RHIC Real Time Data Link [4]. It is a bi-phase mark encoded, differential PECL serial link. An RTDL frame consists of 41 bits (Figure 2). Its payload is a 24 bit data word representing an SNS accelerator parameter pertaining to the upcoming machine cycle. The rest of the RTDL frame consists of a start bit, an 8-bit frame number, and 8-bit frame CRC.

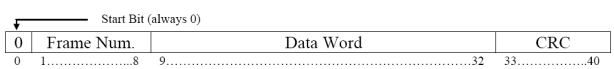

Figure 2: RTDL frame diagram.

Before the start of each machine cycle, a series of frames about the upcoming cycle is transmitted on the RTDL.

Implementation of the RTDL module consists out of three separate sub-modules:

- RTDL Timestamp module for tracking current time while it is synchronized with GPS PPS input signal from GPS receiver.
- RTDL Table Update module, for updating the values in the RTDL table
- RTDL Transmit module responsible for sending out data from RTDL table

Event Link

The SNS Event Link is modeled after the RHIC Beam Synchronous Link [4]. It is a differential PECL, bi-phase mark encoded, serial link that transmits 8-bit event codes synchronized with the ring revolution period. An SNS event actually contains 12 bits. These are a start bit, an 8-bit frame number, a parity bit, and two stop bits. The format of an event frame is show below (Figure 3):

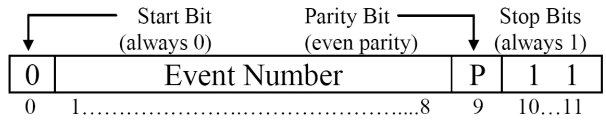

Figure 3: EL frame diagram.

Sending out of the Event Link events is orchestrated by the TurnCounter. Each turn one event is pulled out from the following three sources of EL events, listed from the highest to the lowest priority:

- EL Event Table
- EL HW Event FIFO
- EL SW Event FIFO

Module implementation consists out of separate sub-modules which are interconnected in a pipeline manner.

Major sub-modules are:

- EL Turn Counter module which keeps track of the current accelerator whole turn and also its sub-turns.
- EL Event Decision logic to prioritize event candidates which are eligible to be sent over EL.
- EL Event Encoding and Sending, which sends the given events over the EL.
- Logic which handles handshake with the RTDL module while issuing a request to transmit its data.

VERIFICATION AND TESTING

Primary simulations, during individual module development were performed by using Tcl scripts, which were used to define proper stimulators for signals, while simulation results were inspected and presented as waveforms. Since this principle is only suitable for sub-module and individual component testing, we have used SystemC platform for overall design testing and verification.

SystemC is an open source hardware design and verification language based on C++. It allows engineers to apply powerful, proven software techniques, such as object-orientated design, to the problems of system modeling and verification.

We have used SystemC because it provides the ability to merge test modules written in C++ with actual VHDL modules in a single simulation construct.

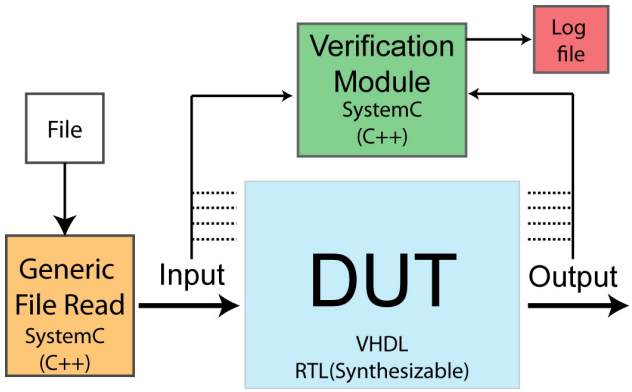

Figure 4: SystemC testing concept.

During the course of testing, we have evolved multiple testing concepts which consist out of a generic file reader module and multiple verification modules.

Generic file reader module accepts an input file with "command/signal value" pairs and asserts corresponding signals with the given value.

Below (Figure 5) is an example of an input file provided to generic file reader module for signal value assertion:

```
#Assert signals with initial values
sigMPS_AR_o              0
sigSpareP2_o             00000000

# Enable rising edge detection
sigHWirqEdgeConfig_o     1000000000000000
wait 50 ns

#Assert and deassert CycleStart signal
sigCycleStart_o          1
wait 50 ns
sigCycleStart_o          0
wait 500 ns
stop
```

Figure 5: Example of an input file with stimulators.

Verification modules obtain the same input values as VHDL DUT (Device Under Test), compute what desired output from the DUT should be like and compare them with the actual module result. Given below (Figure 6) is an example of verification function which is incorporated into its dedicated simulation thread:

```
void PeriodMeasurementTest::prc_PeriodMeasurementTest(){
  while(true){
    wait(DUT_LinePeriodReg.value_changed_event());
      if(DUT_LinePeriodReg.read()!=LatchedCounter){
        REPORT("Period value mismatch");
      }
  }
}
```

Figure 6: Example of a verification thread function.

Beside signal output values, timestamps of the actual result are also obtained and are taken into account, where correct timing boundaries of the result need to be verified. Single verification module can have multiple simulation threads obtaining different signal values at different times, so complex testing can be achieved with minimal code complexity and replication.

Verification results are logged in a report file, which can contain detailed description in case when failure was detected, where timestamp, actual input and output signal values and expected result can be provided.

Equivalent functionality could be done using standard VHDL testbenches, but by using C++ object oriented principle, libraries, streams and standard functions, it can be implemented in much more time and effort efficient manner.

CONCLUSIONS

Presently outdated system timing master hardware based on BNL is replaced by a new, flexible and powerful FPGA design.

Testing and verification plays a very important role in complex device development where usage of SystemC VHDL mixed language simulation greatly enhances usual HDL based verification.

REFERENCES

[1] Oak Ridge National Laboratory
http://www.ornl.gov/
http://neutrons.ornl.gov/aboutsns/aboutsns.shtml
[2] http://www.systemc.org/home
[3] http://www.xilinx.com/products/virtex5/
[4] H. Hartmann, "Specification for the RHIC Real Time Data Link Transmitter System", http://www.sns.bnl.gov/epics/timing/doc/RTDL/SNS-RTDL-specs.pdf