EXTREMELY LOW-JITTER FPGA BASED SYNCHRONIZATION TIMING SYSTEM

J. Dedič*, A. Hasanović, D. Golob, M. Pleško, Cosylab, Ljubljana, Slovenia

Abstract

Injection-involved synchronization timing system must provide synchronization triggers and clocks with the jitter values in the range of few tens of ps. A well-thought-out system-level design approach was necessary, splitting a design into several sub-modules, each addressing the specific synchronization issue. Tight synchronization between the unrelated RF signal and external trigger is based on a PLL phase-shifted over-sampling technique. Beam-monitoring instrumentation synchronization is also handled. An emphasis was put into a design, offering an installation without calibration. Utilizing state-of-the art FPGA circuits we designed a purely digital system, without analogue components (i.e. delay lines) that would require a time-consuming calibration and lead to increasing jitter for long delay ranges. Finally, regardless of its complexity the timing solution has to provide seamless integration into the accelerator facility. To leverage the performance, offered by a dedicated state-of-the-art HW, with flexibility, offered by a SW solution, we used a standard device for peripheral CS integration, based on an embedded processor running OS - a part of a microIOC family of products.

TIMING SYSTEM REQUIREMENTS

In addition to numerous subsystems also a large number of diagnostic equipment is involved in the operation of the accelerator [1], [2]. Usually these subsystems and the diagnostic equipment must be synchronized to bunch transition, which is typically in the range of 2 ns. It is the role of the timing system to provide trigger signals for starting various events across the accelerator. In other words, the accelerator requires just-in-time orchestration of its subparts, which if triggered inappropriately can deteriorate its operation.

When handling repetitious events, it is important that jitter is low enough not to cause unwanted artefacts, e.g. satellite bunches at single bunch top-up procedure. When defining the jitter requirements of the timing system it is important to take into account the fact that all the components – involved in the "trigger-to-execution" path – contribute to the resulting jitter. For example, if we look at the top-up procedure again, one can find that the jitter of the electron gun should not be overlooked. From this it is clear that when adaptations are required, this has to be paid with resulting jitter deterioration. To keep the jitter low-enough, a system with flexible interface is required to minimise the count of additional adaptation components.

To be able to provide bunch-by-bunch information the beam diagnostic equipment has to be triggered synchronously with the radio-frequency (RF) clock. This means that the timing system must have its internal operation synchronized with the RF clock. Of course, not limited to synchronous RF operation, a "standalone" asynchronous operation mode is also required for other modes of operation.

The process of installing a new device is usually not an easy job and therefore as much as possible work should be done in advance. This means avoiding installation-time calibration and control system (CS) integration issues. To avoid calibration, a purely digital system should be used. And to provide flawless integration into the CS, one should choose flexible and proven SW-platform.

All of this was our motivation to design a purely-digital and fully programmable timing system, which requires no extra adaptation to use it to trigger accelerators' hardware equipment. The entire system can be divided into two parts. The first is responsible for timing functionality (shown in Fig. 1) and the second is responsible for the integration into the CS (shown in Fig. 3).

HARDWARE IMPLEMENTATION OF THE TIMING FUNCTIONALITY

To keep the jitter low and reliability of the system high the total number of the components had to be minimised. A field programmable gate array (FPGA) platform was chosen that is a good compromise between performance and flexibility it offers (easily reconfigurable hardware and plenty of resources capable of truly parallel operation at over 500 MHz). In the core of the system there is a Xilinx Virtex-4 [3] and for it we have chosen commercial FX12 mini module from Memec [4], which greatly simplified the base board printed circuit board (PCB) design.

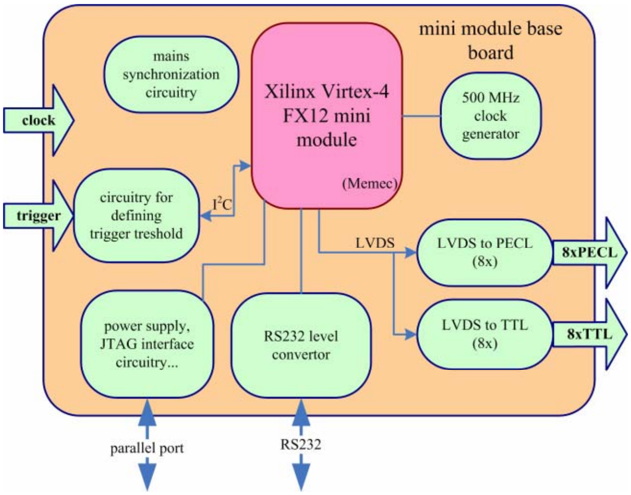

Figure 1: Timing part of the system.

*joze.dedic@cosylab.com

The design of the FPGA is straightforward. Programmable delay is implemented by means of counters. Clock signal for these counters is user selectable. As required, one can either decrement them using on-board low-jitter 500 MHz clock or external RF-clock. When the later is chosen an in-FPGA digital phase locked loop (PLL) circuitry is used that guarantees in-phase synchronous operation. User can check the status of the successful locking of the PLL through status bit. This is needed at start-up or external clock fluctuations. If PLL unlock status is detected a SW initialization procedure is required for aligning phases.

The start event for decrementing the counters is programmable. User can either select external trigger event or in-FPGA numerically controlled oscillator (NCO). For even more flexible external trigger control the user can programmably define the voltage at which the trigger event is recognized. This functionality is implemented using I2C analogue/digital converter and level comparator (placed on a base board). I2C master module is implemented as a part of the FPGA functionality. If user selects NCO as a source for the trigger, mini module's 100 MHz oscillator is used and user directly programmes the values for the NCO. Any on-board trigger repetition rate can be used, as long as it fits within programmed delay scheme. If for example user defines the triggering that is more frequent than the longest delay setting, this can be determined through device status bits. As it is often required that the triggering is synchronized with mains (for example the gun), main board implements a zero-detection circuitry and FPGA provides a support for phase shifting.

The entire functionality, such as selecting trigger and clock source, defining trigger threshold level, setting per-channel delay, defining the polarity and with of the output, reading the status information, etc., is programmed serially. The base board implements only the required signal level-translation, but the complete UART functionality is once again implemented in FPGA.

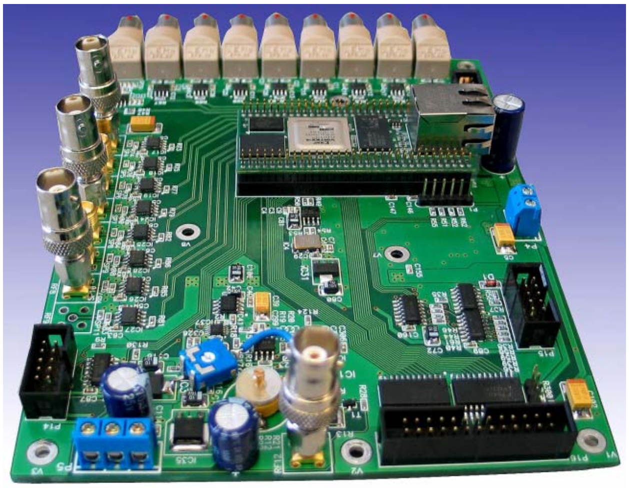

Figure 2: An FPGA base board providing support functionality (power supply and level translation).

Base board implements also a JTAG (joint test action group) interface circuitry. Using this interface the HW functionality of the FPGA can be easily reconfigured through parallel port of the single board computer (presented in the next section). One could even decide to reconfigure the system on-the-fly through the CS if new mode of the operation would require that.

To bring high-speed per-channel signals out of the FPGA and still keep the jitter low, a low voltage differential signalling (LVDS) output buffering was chosen for the FPGA outputs. This LVDS signals are then converted into eight PECL (Positive Emitter Coupled Logic) and eight TTL signals for convenient triggering of different devices. Low-jitter level translators are chosen.

Figure 2 shows the base board, providing all the support functionality required by an FPGA. When routing a board, a special care was required to make appropriate PCB physical layout in order to avoid signal reflections, which could impair the net performance of the timing system.

High-frequency Design of the FPGA

To achieve the final delay step resolution of 1 ns, a two step-approach was required. First, to halve the clock resolution, all the output modules (16 output channels) inside FPGA have selectable operating edge. Using this approach, the clock resolution is improved twofold. The same goes for the trigger detection module, which detects the trigger on both clock edges. Here the double data rate (DDR) module – as the part of the HW functionality of the Virtex-4 FPGA family – is used. For this, each channel has a special circuitry that correlates output edge triggering (+/- half of the clock) with detected input trigger edge and programmed delay.

Second, to be able to provide synchronous operation at frequencies as high as 500 MHz, a well thought design was required. RF frequencies are too high to expect the entire FPGA to operate at them. We decided for a design where only a portion of the design is high-frequency, the rest of the design is operating at eight times lower frequency. By doing this, only a portion of the routing had to be done under strict timing requirements. Doing this, the entire delay is defined by the means of high-frequency counter decrement (clock resolution), low-frequency counter decrement (clock/8 resolution) and output edge (2*clock frequency resolution).

The logic for setting the required delay is handled on the microIOC platform (next section).

INTEGRATION INTO THE CONTROL SYSTEM

We wanted to design a timing system that would provide flexible means of timing control straight out-of-the-box. For this, we expanded the previous hardware, responsible for performance, with the microIOC platform, offering flexibility [5]. This platform is based around single board computer, especially designed for use in stressed and rough environments. It provides the complete

1-4244-0917-9/07/$25.00 ©2007 IEEE

functionality of the personal computer, but sized down to the quarter of letter-paper size.

The operating system we have chosen is Linux, Debian distribution and is booted from the compact flash card. Single board computer offers direct connection of the VGA monitor and keyboard, which enables quick and low level access to the memory mapped settings (registers) in the FPGA through RS-232 port (using Linux minicom). Our timing system was further upgraded by the EPICS control system, which enables comfortable control system integration (using one of the two available Ethernet ports).

Figure 3 shows an example of microIOC platform in a compact demo case having all the needed components for the operation, including power supply and LCD for basic information display. The example in Fig. 3 is extended with eight-port serial communication.

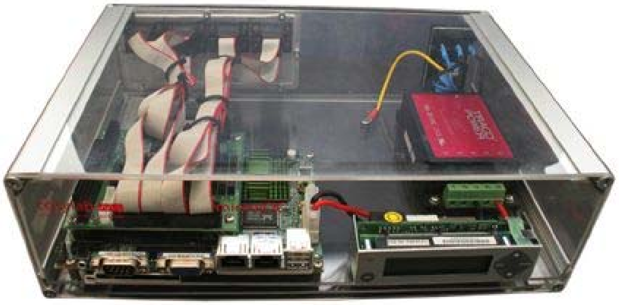

Figure 3: A microIOC platform, providing flexible CS integration.

RESULTS

We have built and successfully tested four prototype timing systems (named microIOC-Delay generator). The resultant output jitter is within expected 50 ps RMS value and the FPGA design is able to operate up to 550 MHz, providing all the mentioned functionality. Above the frequency of 300 MHz the FPGA is overheating, so we had to add a small heat sink (passive cooling is adequate).

Insertion delay for the system (the delay from the trigger event to the reference channel output) is constant for every system and is in the range of 100 ns.

Figure 4 shows an oscilloscope screenshot of output signals. To enable the observation of various output settings and their jitter values, persistence is turned on. The screenshot is made with the selected clock of 500 MHz; it can be seen, the achieved delay resolution step is half the provided clock (i.e. 1 ns).

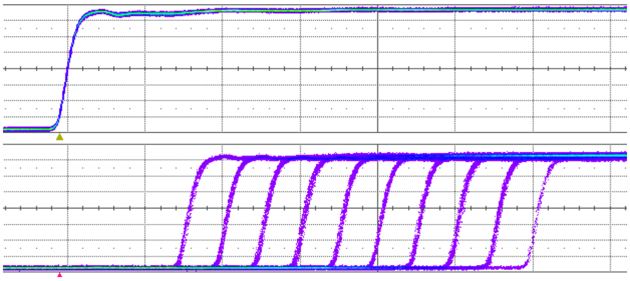

Figure 4: Persistent snapshots of reference vs. output channel (channel delay is incremented by 1 ns steps, oscilloscope time base is 2 ns/div).

REFERENCES

[1] M. Conte, W. W. Mackay, *An Introduction to the Physics of Particle Accelerators*, World Scientific Publishing Company, 1991, ISBN-10: 981020812X.

[2] R. Bakker, R. Goergen, P. Kuske, J. Kuszynski, Measuring Beam Intensity and Lifetime in BESSY II, Proceedings DIPAC 1999 – Chester, UK.

[3] Xilinx homepage: www.xilinx.com

[4] Avnet Memec homepage: www.memec.com

[5] microIOC homepage: www.microioc.com

1-4244-0917-9/07/$25.00 ©2007 IEEE