

A PULSE-PATTERN GENERATOR USING LABVIEW FPGA

D. Beck, H. Brand, H. Hahn, F. Herfurth, S. Koszudowski, GSI Helmholtzzentrum für Schwerionenforschung GmbH, Planckstraße 1, D-64291 Darmstadt, Germany
 G. Marx, L. Schweikhard, F. Ziegler, Institut für Physik, Ernst-Moritz-Arndt-Universität, D-17487 Greifswald, Germany

Abstract

A pulse-pattern generator produces bit patterns at user specified times. It can be used to control the timing of experimental procedures - each bit is used as a trigger line for external devices like a switch-able power supply. The development was initiated by the need of ion trap facilities [1]. Typically, such a facility has about three traps. The experimental procedure requires repeating many times a complex sequence of a few seconds duration with about 30 steps with a precision of 100 ns. The sequence must be synchronized to external events like the timing structure of an accelerator. As a solution, a FPGA card from National Instruments is used. The LabVIEW FPGA module translates the graphical code to VHDL, which is processed further by the tool chain of the FPGA manufacturer Xilinx. Presently, this solution is used at six different experiments at four institutes.

INTRODUCTION

Typical experiments in nuclear and high energy physics have sophisticated data acquisition systems with a large number of channels and event rates on the one hand. On the other hand, the requirements on the slow control system to adjust experimental parameters like high voltages or gas flow are fairly relaxed. For experiments using ion traps, like SHIPTRAP [2], HITRAP [3], ISOLTRAP [4] or ClusterTrap [5], the situation is reversed. While data acquisition is fairly simple, the use of single ions in vacuum requires active manipulation of electromagnetic fields in fast real-time. A simplified sketch of a part of the experimental procedure is depicted in Fig. 1. Shown is a sequence of steps that is called a *cycle*. First, the ions to be investigated are produced by nuclear reactions or, in case of stable species, might be obtained by using pulsed laser beams.

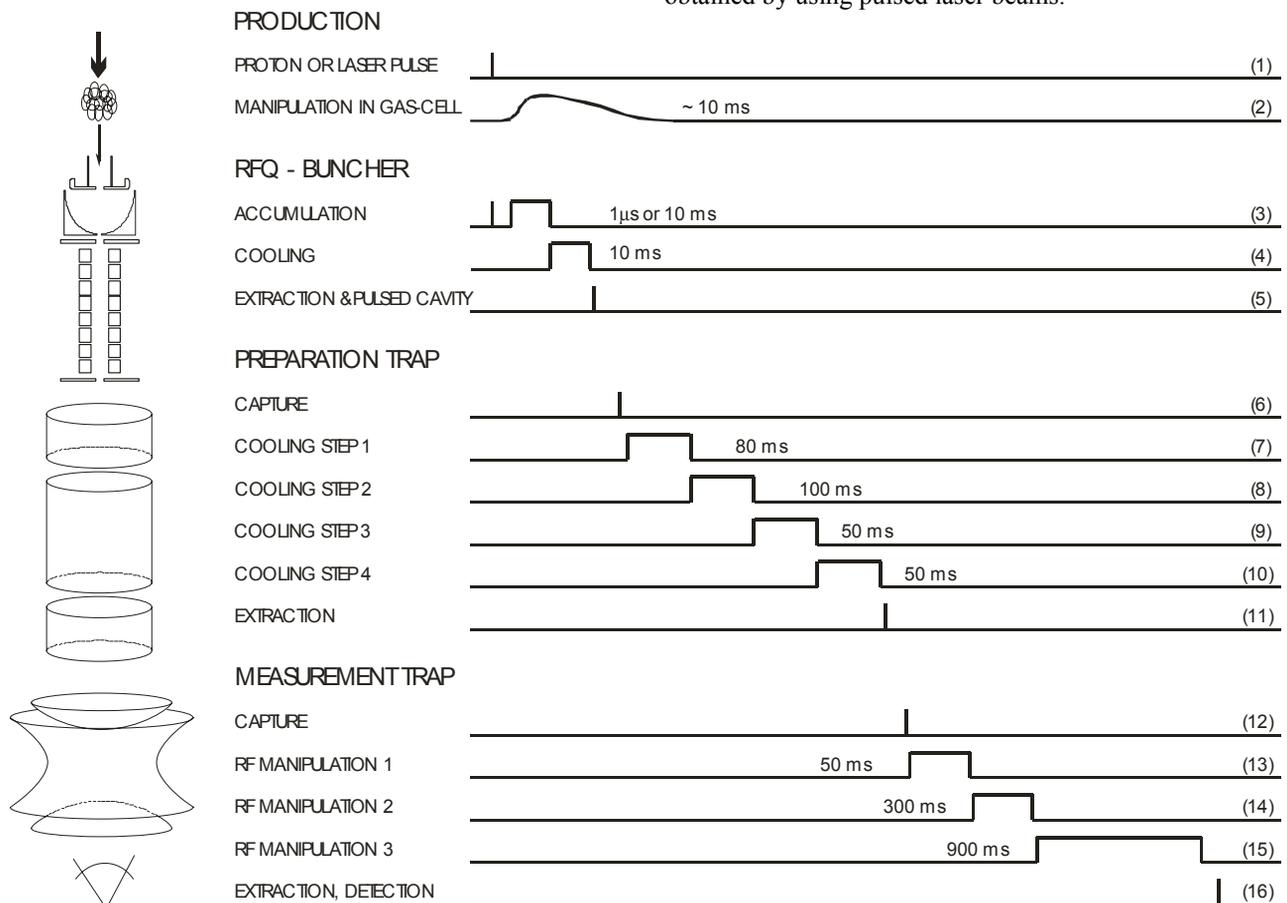


Figure 1: Simplified sketch of an experimental cycle (see text) of a typical trap experiment. The different steps must be synchronized with sub-microsecond precision.

In some cases, the ions are produced at high energies and must be slowed down using a gas cell [6], before they can be caught in a gas-filled radio-frequency quadrupole (RFQ) buncher. Besides being a first cooling stage, the prime task of such a buncher is to produce a well defined bunch of ions with a phase-space volume of a few eV μ s. Such a bunch can be transferred into a preparation trap that serves for further cooling. Typically, a mass selective cooling technique, requiring damping and radio-frequency (rf)-manipulation, is applied to isolate a few ions of interest from a background of unwanted species. The ions are transferred as a cooled bunch to the measurement trap. Depending on the experiment, such a cycle may take a few hundred ms to a few seconds. During an experiment, a cycle is repeated while changing a parameter between cycles. An example is a mass determination that can be performed by measuring the true cyclotron frequency of stored ions with ISOLTRAP [4]. This is accomplished by measuring an ion signal as a function of the frequency of a rf-field applied in the measurement trap, see Fig. 2.

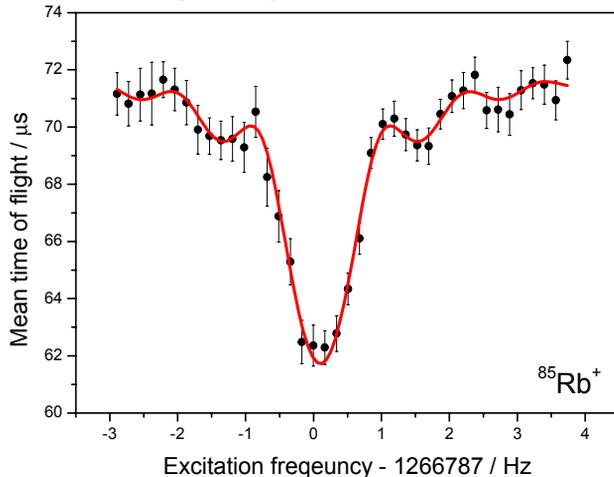


Figure 2: Ion cyclotron resonance using a time-of-flight detection method [4]. The solid line represents a fit of the expected line-shape to the data points.

The steps of a cycle might be changed frequently. Moreover, some of them must be synchronized with sub-microsecond precision either consecutively or triggered by external signals. This is the task of a pulse-pattern generator that produces trigger signals for switch-able power supplies and rf-generators.

REQUIREMENTS

For typical trap experiments 64 digital outputs for device triggering are sufficient. Thus, the pulse-pattern generator should produce bit patterns with a width of 64 bit. Each pattern is applied for a time specified by the experimentalist. When transferring ion bunches between traps, it is required to control the timing with a precision in the order of 10 ns. The same precision is required when ions need to be ejected or captured with a fixed phase correlation to rf-fields that have frequencies up to a few MHz, depending on the mass-to-charge ratio of the ions.

Reconfigurable Hardware

Conditional execution of pattern generation must be possible at every moment depending on the experimental procedure. This is important for synchronization to external signals like laser-pulses or the timing structure of the accelerator that is involved in the ion production. Up to eight digital inputs are sufficient and allow up to 255 different trigger conditions. Another feature is the repetition of certain steps within one cycle in a loop. As an example, this allows accumulating many bunches from the RFQ-buncher in the preparation trap prior to the actual preparation. Furthermore, it must be possible to change the time of individual patterns without re-loading a whole sequence of patterns. This feature allows measuring experimental parameters as a function of time, which is especially important for tuning the experimental-setup during the preparatory phase of a beam-time. Precision experiments often compare a property of a particle to another particle that is well known. As an example, the mass of a radioactive nuclide is measured relative to the mass of ^{12}C , the atomic mass standard. Different nuclides may require different timing schemes. Thus, it must be possible to quickly change the experimental procedure between cycles. This is similar to the *fast context switching* applied at the GSI accelerator (virtual accelerator concept).

The pulse-pattern generator is intended for trap and similar experiments. These experiments are small and not distributed over buildings or whole sites. This has two consequences. First, a distributed timing system is not required, i.e. a single pattern generator is sufficient. Second, such experiments rarely have staff specialized for the control system only. Thus, the solution must be simple and should be based on hard- and software commercially available.

SOLUTION PATH

The solution presented here is based on a reconfigurable input/output (RIO) 7811R card from National Instruments. This card has a Field Programmable Gate Array (FPGA), embedded RAM as well as a PCI or a PCI eXtensions for Instrumentation (PXI) interface. This card can be integrated into a LabVIEW environment. The FPGA, a Virtex-II V1000, can be programmed by using the LabVIEW FPGA-module, which creates VHDL code from the graphical code programmed by the developer. The VHDL code is then transferred to the tool chain of the FPGA manufacturer Xilinx, which creates a bitfile that is finally uploaded to the FPGA. The FPGA-module allows communication between normal LabVIEW programs on a host PC and the FPGA code during run-time. Out of 160 digital I/O lines, 64 lines are used for pattern output. Eight lines are used for trigger inputs. A few additional output lines allow monitoring and debugging.

FPGA

The main idea is the following: Commands and their parameters are uploaded to the on-board memory of the

FPGA card via Direct Memory Access (DMA) transfer. Four different commands are implemented.

- *\$time*: A 64-bit pattern is applied to the output lines for a specified time by counting FPGA-clock-ticks.
- *\$wait*: A 64-bit pattern is written to the output lines. No further processing is done until a specified trigger condition is detected on the eight input lines.
- *\$jump*: Command used for implementing loops.
- *\$stop*: Last command for a sequence of patterns. This can be used to separate multiple sequences of patterns and is required for *fast context switching* (see Sect. REQUIREMENTS).

Two different clock domains are used. Some part of the FPGA is clocked with a 40 MHz reference clock. However, the actual pattern generation is performed within a so-called single-cycle loop that is operated at 80 MHz. A FIFO is required to transfer data between the two clock domains. When the FPGA starts execution, the commands are copied from the onboard memory to a first-in-first-out (FIFO) buffer. The FIFO is emptied by the single-cycle loop that does the final processing of the commands.

The operation of the FPGA is controlled by a simple state machine supporting the following states.

- *idle*: Default state, waiting for commands from the host PC.
- *reset*: Clear onboard memory.
- *query*: Query information from the onboard memory.
- *load*: Load new data to the onboard memory.
- *run*: Start copying of data from the onboard memory to the FIFO and start processing the commands in the single-cycle loop.

The states *reset*, *query*, *load* and *run* can only be accessed from the *idle* state and return to the *idle* state after executing the entry, do, and exit actions of a state. Error conditions are handled by the *idle* state, which then transfers error messages to the host PC.

Host PC

Dedicated LabVIEW code that is part of the FPGA-module allows communicating with the FPGA target from a host PC. Although using that code is in principle straight forward, implementation details are hidden by encapsulating the direct communication with the FPGA target. This is achieved by the implementation of an instrument driver according to the *Instrument Driver Guidelines* [7]. An instrument driver defines a well defined Application Programming Interface (API) that can be used from an application program without any knowledge about FPGA programming.

STATUS

A first version of both the FPGA and LabVIEW code on the host has been developed by two authors of this paper, F. Ziegler and S. Koszudowski. It is in production since about three years at six different experiments [2-5,8,9]. The original requirements have been refined further as presented in this paper, and small parts of the

code have been adapted accordingly. The code is GPL licensed. The solution presented here has been developed on the "smallest" RIO card produced by National Instruments. The code can easily be compiled for larger or faster FPGAs and is not restricted to the 7811R card. If required, the FPGA clock can be linked to an external timebase, which is available if the RIO card is used on a PXI platform.

APPENDIX

Note that the 7811R provides only TTL signals for output. However, some laboratory devices require higher currents for their trigger inputs and a TTL line-driver has been developed, see Fig. 3. The line-driver module is built as 19" crate providing up to 160 output lines. Each output line provides a 100 mA output current at 5 V with a rise-time of 15 ns.

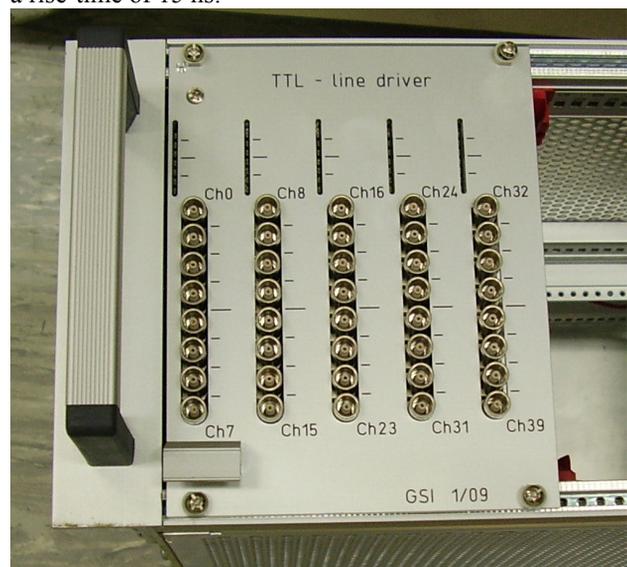


Figure 3: TTL line-driver module. Channels 0-31 are used for 32 output lines. Channel 32-39 are eight input lines for conditional triggering. A second module is required for using all 64 output lines and monitor signals.

REFERENCES

- [1] F. Herfurth, *Int. J. Mod. Phys. E* 18, 2 (2009) 392-404.
- [2] M. Block et al., *EPJ D* 45 (2007) 39-45.
- [3] H.-J. Kluge et al., *Adv. Quant. Chem.* 53 (2008) 83-98.
- [4] M. Mukherjee et al., *Eur. Phys. J. A* 35 (2008) 1-29.
- [5] L. Schweikhard et al., *Eur. Phys. J. D* 9 (1999) 15-20.
- [6] J.B. Neumayr et al., *Nucl. Instrum. Meth.* 244 (2006) 489-500.
- [7] National Instruments, *Instrument Driver Guidelines*, <http://www.ni.com/idnet>.
- [8] V.S. Kolhinen et al., *Nucl. Instrum. Meth. B* 266 (2008) 4547-4550.
- [9] J. Ketelaer et al., *Nucl. Instrum. Meth. A* 594 (2008) 162-177.