FPGA BASED IMAGE PROCESSING SYSTEM FOR ELECTRON BEAM WELDING FACILITY*

M. M. Sizov†, A. A. Starostenko, K. A. Blokhina, A. M. Medvedev
Budker Institute of Nuclear Physics of the Siberian Branch of the RAS, Novosibirsk, Russia

Abstract

Automated electron beam welding (EBW) facilities require real-time control systems. EBW facility at BINP has recently renewed its control unit. High-level part of control system runs as Linux user-space application written in Erlang, while real-time subsystem runs on Field Programmable Gate Array (FPGA) for signal processing and welding control. Time-sensitive algorithms such as joint finder algorithm and pipelined data filters implemented in VHDL and dataflow DSL language Caph. DSL eases implementation and testing of algorithms and translates code efficiently to HDL. Control system filters noise and calculates a joint location for weld autocorrection within 2 µs.

INTRODUCTION

Budker Institute of Nuclear Physics has production of electron beam welding facilities (EBW) for low series production (such as manufacturing of vacuum chambers for FAIR project [1]) and for materials experiments such making hafnium carbide [2]. The largest of all facilities built has cylindrical vacuum chamber 3.5 m long and 0.98 m diameter, it has a coordinate table inside with achie-vable area of 1970×300 mm. Electron gun's acceleration voltage is 60 kV and power of its beam is up to 60 kW. It has two fore-vacuum pumps and three turbomolecular pumps. The emitter is directly heated tantalum cathode and the magnetic coordinate system has two pairs of correctors, that can deflect the beam in two dimensions. Secondary emission electrode is located below electron gun and could be used for constructing of a height map of sample welded or as a joint finder.

Foundation of the control system is a commercially available single board computer DE1-SoC. On board is Intel Cyclon V system on chip (SoC) with dual-core ARM processor and field gate programmable array (FPGA), 1 Gb RAM, ADC, Ethernet, USB. Having standard ARM processor on board allows to easily reuse thousands of open source software packages ported to ARM architecture including operating system, programming languages build tools, message brokers.

FPGA can process data in parallel – different subsystems run simultaneously, each of subsystems constructs a pipeline. Thus, the time needed for data processing (and react to inputs) is the largest of pipeline latencies, which is fixed and could be easily calculated. FPGA's generally have 20-300 K of logic elements, 1-12 Mb of memory blocks, 25-300 DSP blocks which altogether allow implementation of complex real-time algorithms.

High-level control system part is written in Erlang functional language that has a number of features that ease implementation of the control system [3]. It has language concepts for binary data decomposition. And actor-model parallelism allows to write loosely coupled finite state machines. Erlang ships OTP templates library, that features patterns such as FSM and process monitors.

Although DE1-SoC is great in many aspects, it has limitations too. It doesn't have SATA connectors, so only possible options for storage are microSD and USB drives. It has VGA port, but its connected using FPGA fabric, so having user interface on system would take extra FPGA space. Memory and calculation intensive algorithms are not suited well for CPU processors as well as FPGA. To mitigate these limitations, we implemented "thin" client on regular desktop machine, that connects to control system that is able to store data and build surface heights approximation for the operator. Interface is built with ElectronJS framework and uses React for UI widgets and logic.

IMAGE ACQUISITION

EBW facility is able to weld long titanium, aluminum or steel samples. Electronic gun is stationary so coordinate table moves the sample. Having surface height map, operator enters sequence of coordinates and parameter values at key points such as current of heater, corrector coils or anode. To weld a joint operator needs to set two or more points on sample. Then control system sets high current on cathode heater, starts to move sample according to entered sequence and in case of long sample welding corrects deviation of joint position from program.

Figure 1 shows a block scheme of hardware setup. Between electronic gun and coordinate table we have a magnetic system, that is able to deflect the beam. To construct the surface map of the sample under the electron gun, user turns on electron beam with low heater current, then Erlang control system programs FPGA to repeatedly write periodic waveform signal to magnetic system. Magnetic system has corrector coils that deflect the beam in two dimensions. Electrons hit the surface and scatter. Fraction of reflected electrons hit secondary emission electrode. FPGA system synchronizes data from coordinate table, secondary emission electrode and deflection angle that was written into magnetic system. The principle is the same as in electron microscopes (but with much higher beam intensity and subsequently noises). FPGA preprocesses image scan lines with pipelined algorithms to reduce a noise. Prepared data is passed into joint finder FPGA [4] submodule that calculates joint position in the same units as magnetic deflection angle.

* work supported by Russian Science Foundation (project N14-50-00080)
† email address: sizov.m.m@gmail.com

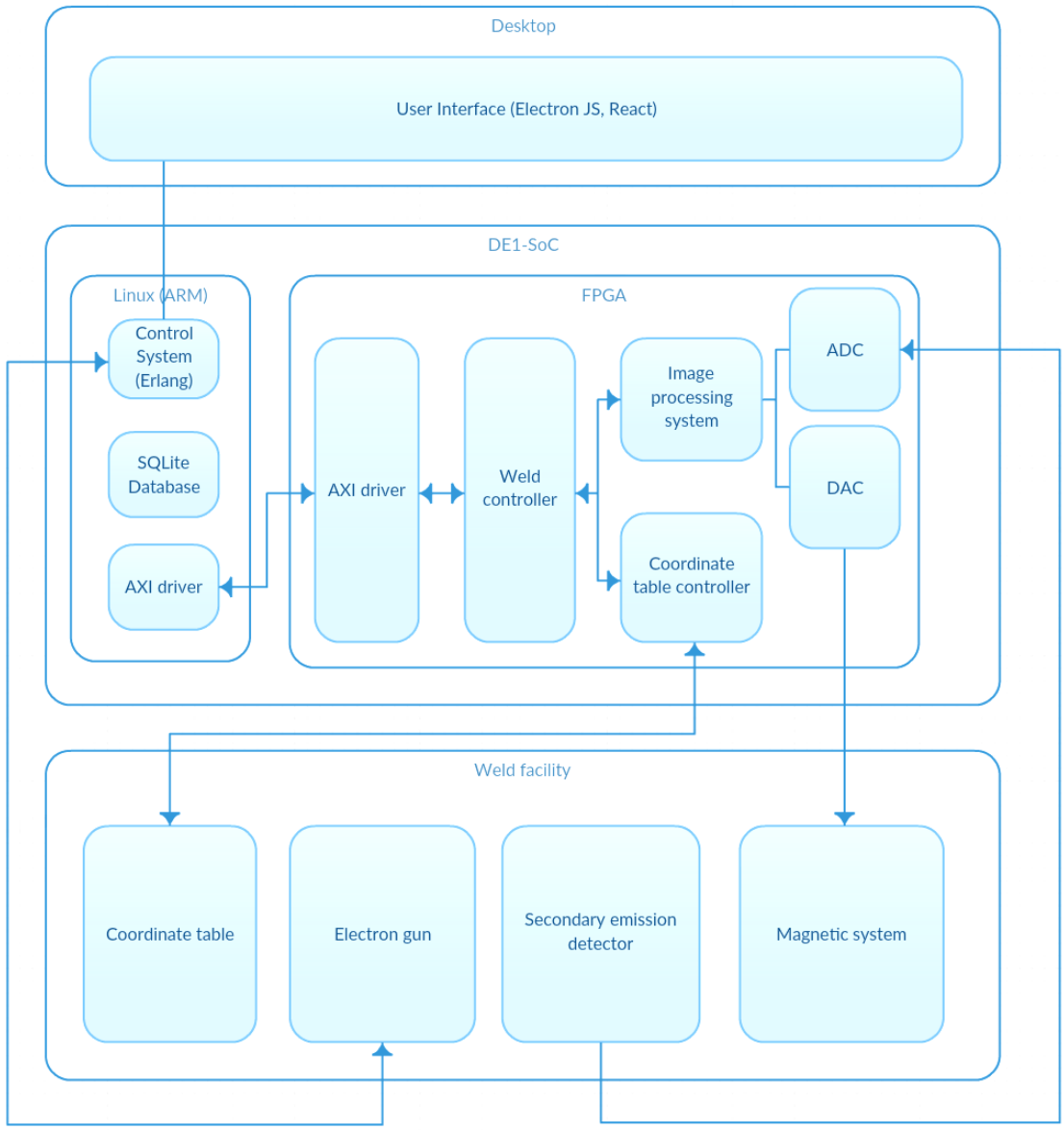

Figure 1: Block scheme of control system data flows.

Passing this position in a negative feedback loop to magnetic system compensates operator precision errors, vibrations and heating deformations. Preprocessed image line is sent to Erlang code, that stores data in DB and passes it through into ElectronJS interface.

Interface part accumulates image lines and binds height information to table coordinates. Current FPGA system has up to 500 of pixels per line with resolution from 60 µm per pixel up to 2 µm per pixel, although the minimal achievable beam spot is approximately 1 mm in diameter. 2 meters with 2 µm per pixel resolution would be a million of pixels which is not reasonable to work with. We artificially limit spatial resolution to 0.1 mm per pixel in long axis direction that makes working with picture feasible. As we have JavaScript as UI base, we had a need to speed up manipulations with image data which can be up to 16000×500 pixels. We have implemented WebGL shader for performance improvement. Shaders are parallel and executed in GPU processing units, algorithms implemented include zoom, pan and data filters such as FFT, median filter, and special filter that equalizes intensity of different rows between each other.

FPGA BEAM DIAGNOSTICS SYSTEM

FPGA is also used for the beam diagnostics system is based on pinhole probing method. The size of the pinhole is very small compared to the minimal beam diameter and to capture the beam characteristics, the beam is raster scanned over the hole. The beam current is collected by Faraday cup under the pinhole and through the resistor the ADC signals for each beam cross-section are obtained. In order to get better resolution the sets of signals are processed by filter method. This system is being developed for obtaining the beam power density distribution, which is the most important parameter of the beam used in welding applications.

The power density distribution maps are built from at least two beam projections. This scanning method allows obtaining a number of beam cross-sections in different projecting planes. The method of processing of data to be used for mapping the power density distribution is at the testing stage. The transversal cross-sections of the beam are shown on Figure 2. This scanning method with high accuracy allows characterizing the beam shape and transversal size. The measured beam size is 1.3 mm.

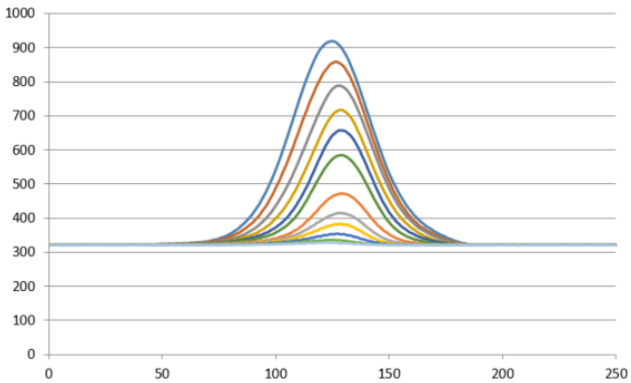

Figure 2: Beam cross-sections received by scanning with 0.015 mm per pixel (x axis corresponds to index in image line, and y axis shows the signal in mV).

FPGA IMAGE PROCESSING SYSTEM

Caph [5] is a high-level domain specific language which translates to VHDL. Dataflow paradigm utilized in this language exploits true parallelism of FPGA and ability to construct pipelines. Processing of sample heights starts with reducing of electromagnetic noise from 60 kV power source. Median filter removes spikes and a low-pass filter reduces high frequencies. Algorithms written on Caph describe algorithm as network of actors with inputs and outputs. On Figure 3 example of such graph is shown for median filter. ADC measurements are abstracted as stream of structured data that is either start of line, valid value or end of line. Values are consumed at actor inputs. Each actor is independent state machine and executes algorithms in parallel (as this code runs in FPGA), computed outputs are distributed to each input connected.

This computational model is declarative (as well as hardware description languages are) so it has a little overhead compared to hand-written code. As this language is functional, developer can execute and functionally test code and compute latencies on desktop PC without having to simulate hardware on register lever or compile FPGA firmware.

CONCLUSION

In this project, data is processed in each layer – FPGA, on Erlang control system and in user interface. Control system features ability to perform beam diagnostics, set arbitrary waveforms and provides relative heights of welded sample. Operator uses data acquired during scanlines to properly set welding points and weld samples. Arbitrary scanlines

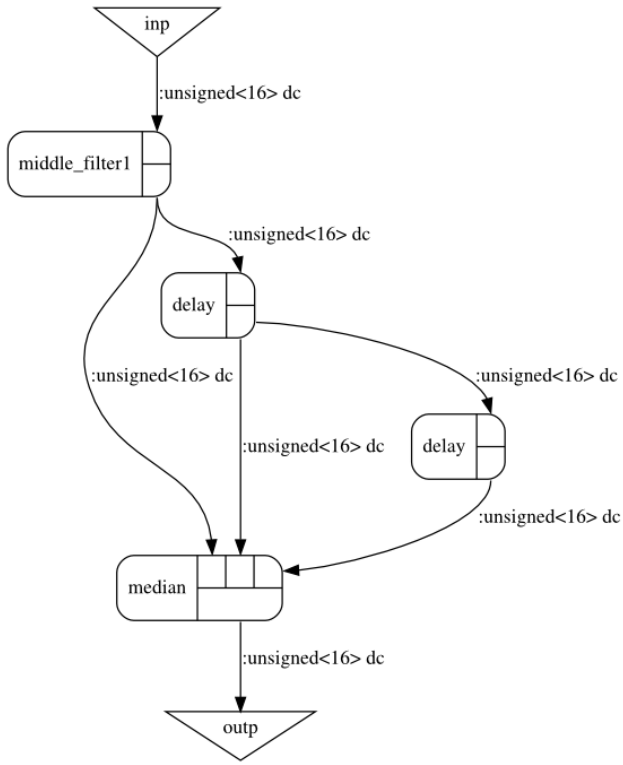

Figure 3: Calculation graph for median filter written in Caph.

support various experiments on quality of seams. In addition, joint-finder module was developed using domain-specific language Caph. Module compensates difference between joint position and center of welding scanline.

REFERENCES

[1] A. I. Ancharov, N. F. Grigoreva, P. V. Logachev, Iu. I. Semenov, A. A. Starostenko, B. P. Tolochko, "Possibility of application of hafnium and tantalum carbides as materials for additive manufacturing," The international seminar on interdisciplinary problems in additive technologies. 6-9 December, 2016, Tomsk.

[2] A. M. Medvedev, A. M. Semenov, Y. I. Semenov, A. A. Starostenko, M. M. Sizov, A. S. Tsygunov, "EBW application for the manufacture of HEBT dipole vacuum chambers, FAIR," *"E+E"*, 2016, p. 148–152.

[3] K. Kruger and A. Basson. "Erlang-based control implementation for a holonic manufacturing cell," *International Journal of Computer Integrated Manufacturing*, vol. 30.6, pp. 641–652, 2017.

[4] M. M. Sizov, A. A. Starostenko, A. S. Tsygunov, A. M. Medvedev, "Real time FPGA-based seam finder for EBW facility," *Electrotechnica & Electronica E+E*, The Union of Electronics, Electrical Engineering and Telecommunications /CEEC/, 2018, vol. 53, no. 5–6, pp. 145–148.

[5] J. Sérot and F. Berry, "High-Level Dataflow Programming for Reconfigurable Computing," *International Symposiumon Computer Architecture and High Performance Computing Workshop (SBAC-PADW)*, 2014.