FPGA based image processing system for electron beam welding facility

Mikhail Sizov, Alexandr Starostenko,
Kseniya Blokhina, Alexey Medvedev
Budker Institute of nuclear physics SB RAS

BINP Electron beam welding facility

Features:

- 4.5m long
- 0.98m diameter
- 60 kV accelerating voltage
- Up to 1A beam current
- 2D coordinate table
- 2D magnetic corrector
- Secondary emission detector

BINP Electron beam welding facility

- Long samples (up to 2m)
- Steel, copper, tantalum

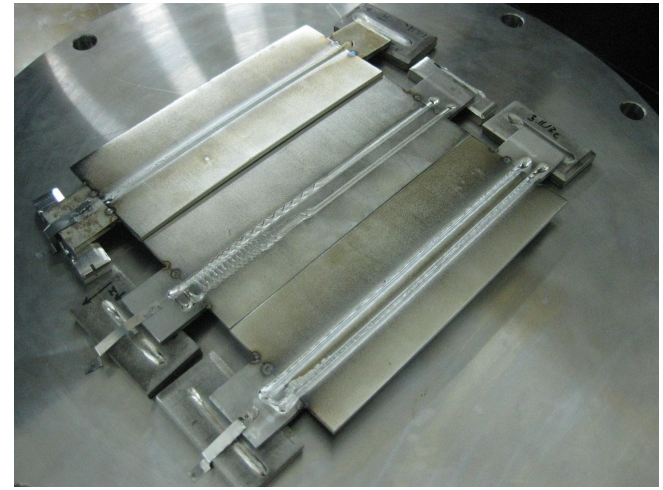
So, we can weld vacuum chambers inside vacuum (e.g. for FAIR HEBT project)

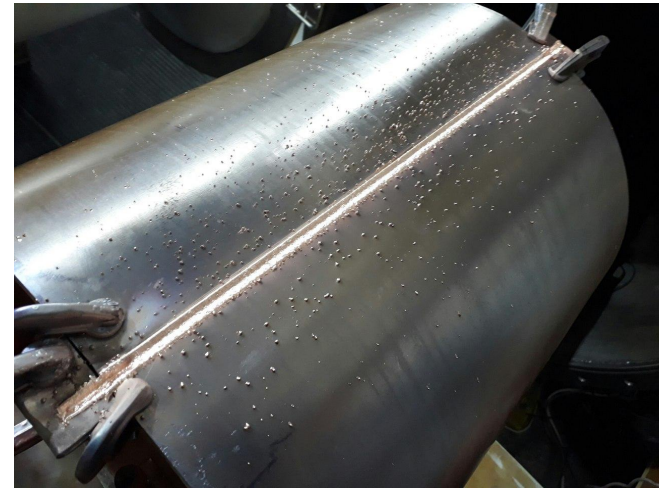
BINP Electron beam welding facility

And engrave...

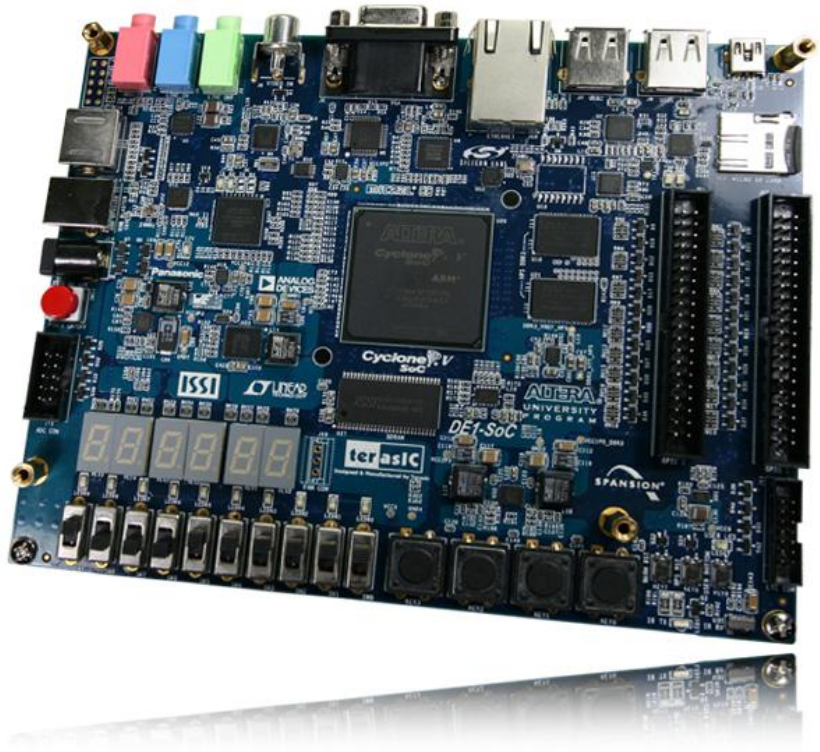
Problem statement

- Evaporated metal in chamber (=>difficult to use cameras)
- Thermal deformations
- Mechanical vibrations
- Precision of joint position estimation
- No interrupts
- Coordinate table and electron gun synchronization

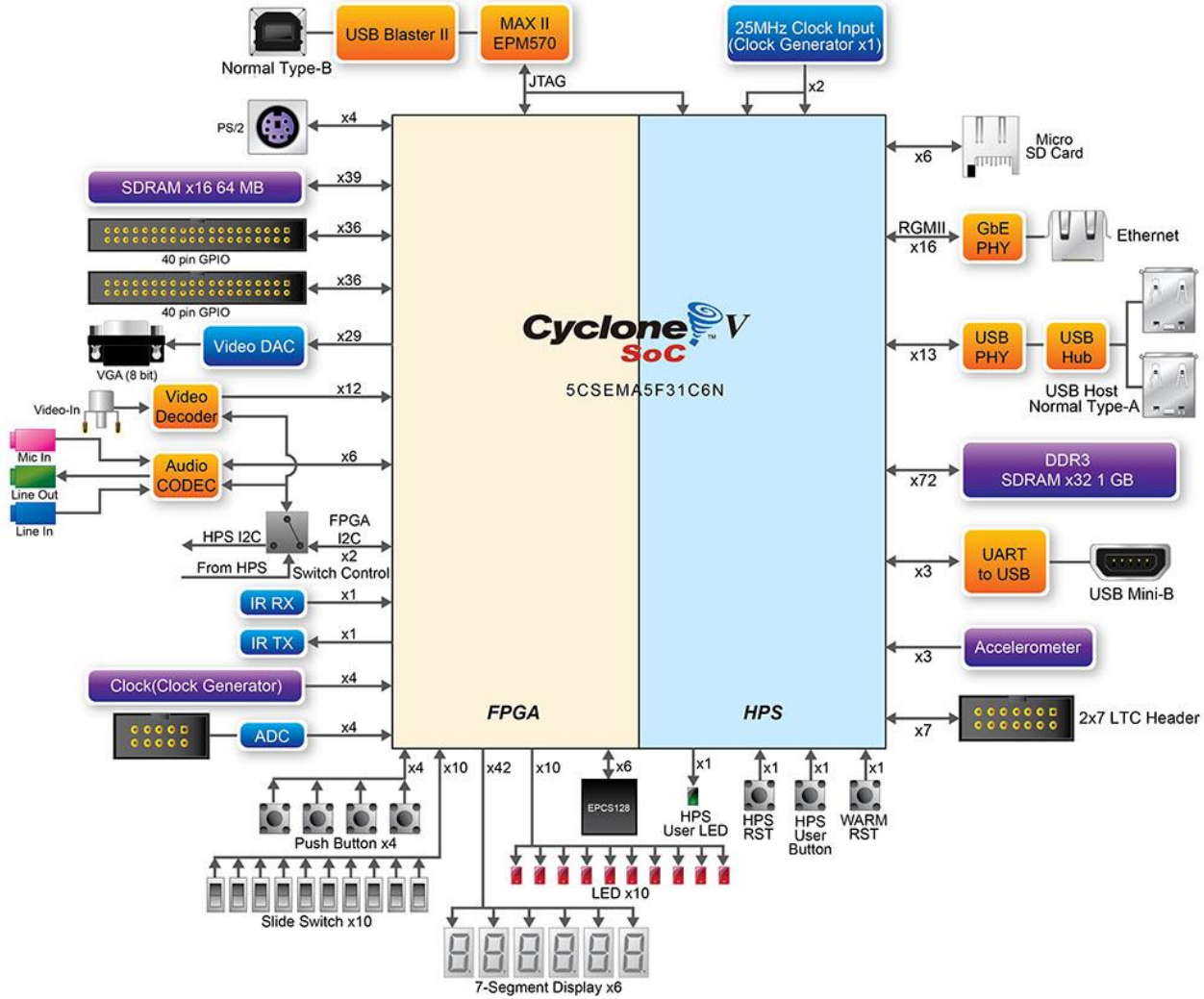




Hardware (FPGA + CPU) DE1-SoC/DE10-Nano



ALTERA
PREFERRED BOARD FOR
OpenCL™


Normal Type-B
USB Blaster II
MAX II EPM570
JTAG
25MHz Clock Input (Clock Generator x1)
x2
PS/2
x4
SDRAM x16 64 MB
x39
40 pin GPIO
x36
40 pin GPIO
x36
VGA (8 bit)
Video DAC
x29
Video-In
Video Decoder
x12
Mic In
Line Out
Line In
Audio CODEC
x6
HPS I2C
FPGA I2C
x2
From HPS
Switch Control
IR RX
x1
IR TX
x1
Clock(Clock Generator)
x4
ADC
x4
Cyclone V SoC
5CSEMA5F31C6N
FPGA
HPS
x6
Micro SD Card
RGMII x16
GbE PHY
Ethernet
x13
USB PHY
USB Hub
USB Host Normal Type-A
x72
DDR3 SDRAM x32 1 GB
x3
UART to USB
USB Mini-B
x3
Accelerometer
x7
2x7 LTC Header
x4
x10
x42
x10
x6
x1
x1
x1
x1
Push Button x4
EPCS128
HPS User LED
HPS RST
HPS User Button
WARM RST
LED x10
Slide Switch x10
7-Segment Display x6

Hardware (FPGA + CPU) DE1-SoC/DE10-Nano

Hard processor system:

- Standard Arm 1GHz CPU, 1Gb Ram
- Can run Linux
- Has Ethernet
- Lots of standard Linux tools/compilers available (C++,Python,Java …)
- **No need to standalone development**
- Good for high-level systems development

Hardware (FPGA + CPU) DE1-SoC/DE10-Nano

FPGA:

- ~100k LUT run in parallel with 50MHz
- DSP blocks
- Internal FPGA memory 1Mbit
- Has standard **memory-mapped interface** to CPU

Software stack

ERLANG

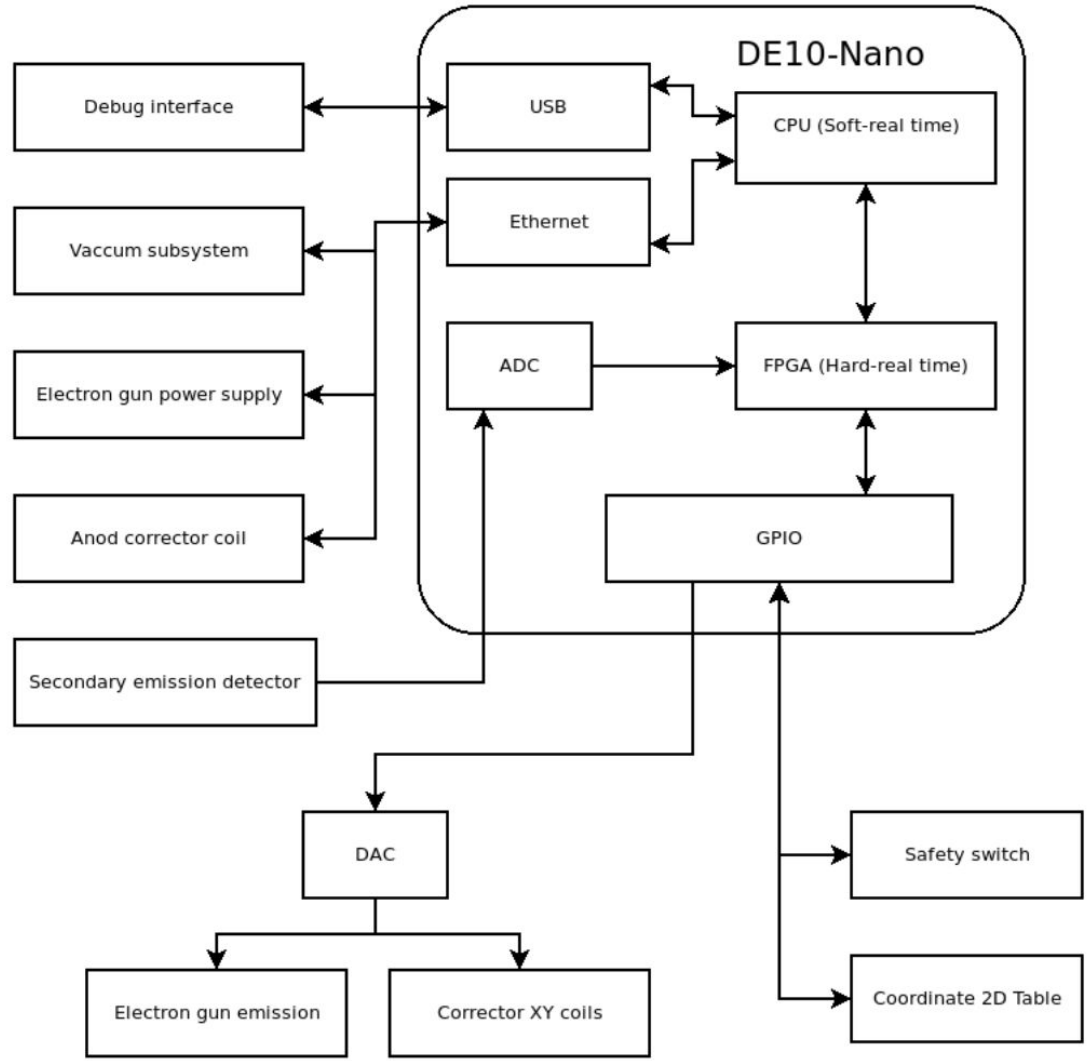
Linux

REACT

ELECTRON

SystemVerilog

CAPH



User interface

Automatic correction system

- Continuously scans surface while welding
- Must be real time
- Synchronous to correctors/coordinate table/scanline generator

Beam profiling system

- Measures beam spatial energy distribution
- Used to minimise beam diameter
- Must be real time
- Synchronises secondary emission, magnetic correctors and pinhole beam detector

Real time system (in our case)

- Field gate programmable arrays
  - Can be real time
  - True parallelism
  - Constructing pipelines (high throughput, but have latency)
  - Low level coding (VHDL/Verilog), gate level

  Other options have difficulties combining computational performance and low latency

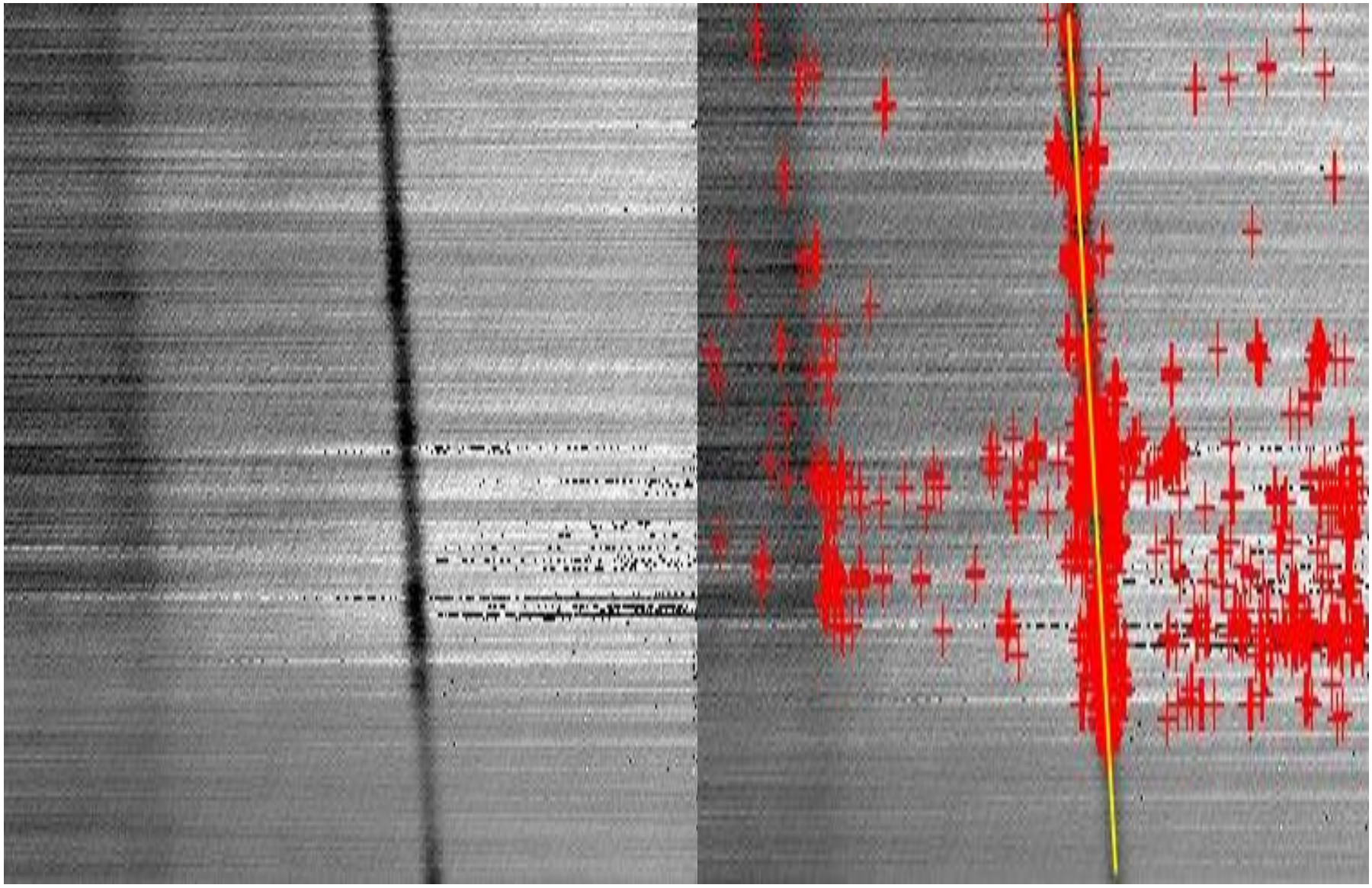
  In general RT systems are **difficult** to implement & debug

Caph

- Domain specific language
- Uses dataflow / functional paradigm (actors and nets)
- Has logical and clockwise simulators (used to fast debugging of algorithms)
- Translates code to VHDL

Results

- Enhances image by applying filters
- Calculates offset within 2ms (that is scanline period + latency)
- Latency is less that 20 FPGA cycles (400ns)
- System is limited by other pieces of hardware
  - Magnetic correctors
  - ADC and DAC curcuits



Conclusions

1. FPGA provide means to implement RT image processing system
2. Hybrid CPU/FPGA processors ease interaction between high and low level code by standard protocols/libraries
3. DSL usage speeds up development of RT system

Thank you for your attention!