Marek Funtowicz

Electronics and Communications Engineer

Personal Details

Address: Henryka Sienkiewicza 6/14
44-310 Radlin

Phone: +48576146398

E-mail: ice.marek@yahoo.com

Web: icenet.online

Git: github.com/IceDefcon

Education

2014 – 2017 **The University of Sheffield**
Electronics and Communication Engineering
Master of Engineering

2013 – 2014 **Open Study College**
Pure Mathematics
A-Level

2012 – 2014 **Central College Nottingham**
Electrical and Electronic Engineering
Higher National Diploma

Skills

FPGA: Intel, Xilinx

CPU: x86, x64, ARM, PowerPC

RTL: Verilog, VHDL

Code: ASM, C, C++, Python, Fortran

OS: FreeRTOS, VxWorks, Linux

Comms: USB, UART, SPI, I2C, JTAG, GPIO, Ethernet, CAN, SRIO

Circuit: Mentor graphics, Spice, Multisim, Cadstar

Test: Scope, Logic Analyser

Matrix: Matlab, Octave

Web: HTML, PHP, CSS, Java script, Bootstrap

Lang: Polish (Native), English (Fluent)

Personal Research and Development

Research and Design Computer Network Analyzer driven by ARM Cortex-M4F Microcontroller, Cyclone II Intel FPGA and x86 platform (ASM, C and VHDL)

- x86 → Linux application for overall master control
- ARM → Driven by FreeRTOS or Bare Metal
- RTOS → TCP/IP Stack → Communication with x86 terminal
- RTOS → SPI interface for ARM request → FPGA
- FPGA → Driven by Nios II Firmware
- FPGA → SPI interface for FPGA response → ARM
- FPGA → Fast Fourier transform and digital filtering
- FPGA → External 640Mbps and 800Mbps LVDS channels for Rx and Tx respectively to communicate with the Ethernet target network

Research Linux Kernel for the x86, arm and arm64 hardware platforms and Booting procedures (ASM and C)

- BIOS, MBR and UEFI (x86)
- PMUFW, FSBL, ATF (arm, arm64)
- BOOTLOADER (U-Boot, GRUB)
- Kernel x86 Boot protocol
- Kernel Initialization
- Linux Drivers
- Makefile
- Linker Script

Research STM32 Cortex-M3 (ASM and C)

- Research STM32F103C8T6 startup codes
- Makefile and Linker script to compile application
- Flash and launch test application

Tronel — October 2019 → Present

FPGA Software Engineer

Research and design the components of LTE communication protocol within CPU and FPGA

- PowerPC T4240 and Intel Nehalem CPUs
- Kintex and Virtex FPGAs
- Symetric multiprocessing @ VxWorks RTOS
- Thread Safe Coding (C/C++)
- Low level debug (PPC and Intel ASM)

Turbo Power Systems — August 2019 → October 2019

Embedded Systems Engineer

Design Firmware and RTL for Power Electronic Device using AURIX 32-bit Tri-Core Microcontroller and AURIX 7 Xilinx FPGA (Soft Power Bridge)

- Research the Infineon Low Level Drivers for the Tri-Core
- Develop Tri-Core Firmware drivers for ASCLIN (UART), QSPI and CAN peripherals to communicate with devices installed on PCB
- Research Tri-Core Firmware for TCP/IP Stack (LWIP)
- Research ARTIX 7 RTL to communicate CPU and PCB over Fiber Optic channels
- Research Microblaze firmware in ARTIX 7 FPGA to communicate with the CPU over QSPI
- Investigate and analyze circuit schematics and PCB layout
- Test the behavior of the Firmware on the Tri-Core development board and the actual product PCB (Soft Power Bridge)

Junior Hardware Engineer

Design computer hardware for capturing, processing and streaming live video using Intel Altera FPGA technology and firmware control

- Design FPGA hardware modules in VHDL and Verilog
- Design QSYS networks for Nios II processing
- On-chip memory management for Nios II processing
- Test RTL modules with Modelsim
- Test and debug RTL modules with signal tap
- Test and debug RTL modules with live PCB
- Test functionality of live PCB with measurement equipment
- Design Nios II firmware for graphical signal processing in FPGA
- Design Nios II firmware for peripheral devices mounted on PCB
- Test and debug Nios II firmware on live FPGA usign Eclipse
- Integrate firmware with BSP and HAL drivers for Intel IP definitions
- Investigate booting codes for embedded Intel Nios II processor
- FPGA pin planning → IO Banks, Voltages, Transceivers
- FPGA chip planning → Modules connectivity and Logic congestion
- Timing analysis for RTL → Clock Cross Domain , False paths
- Investigate and analyze circuit schematics and PCB layout
- Integrate FPGA hardware code and CPU firmware with the circuit schematic and PCB Layout

Prototyping embedded graphical processing system using Xilinx FPGA technology and Linux firmware control

- Develop booting sequence for Zynq Ultrascale MPSoC in QSPI → hand over FPGA control to Linux kernel via U-Boot
- Research bare metal codes → PMUFW, FSBL, ATF
- Research and modify U-boot source → Configure MAC addressing
- Research bare metal XEN Hypervisor → Register XEN watchdog
- Develop UART communication for Zynq Ultrascale MPSoC → register UARTLITE driver in Linux to communicate with external chip for TMDS processing and HDCP control
- Research kernel → Version control
- Research device tree → Petalinux overlay
- Research root file system → Network configuration
- Customize Petalinux → Kernel, Root file system and device tree
- Debug kernel in OS awareness mode → Eclipse environment

Capacitor discharge interface using Intel 8051 microprocessor (ASM)

- Design interface to discharge bank of capacitors installed in the hybrid racing car during crash simulation
- Develop and simulate analogue circuit, digital hardware and firmware (Multisim, ASM)

Vending coffee machine interface using Freescale KL25Z (C++ and ASM)

- Develop (C++) and debug (ASM) testing code procedure for vending machine in Processor Expert and KEIL µVision software

5-bit instruction microprocessor using Xilinx ISE Design Suit (Verilog)

- Design CPU to compute addition, subtraction, multiplication and division of binary I/O signals using pipeline architecture (Verilog)
- Design 5-bit instruction set stored in ROM for CPU control
- Test RTL on Modelsim and live Spartan FPGA Hardware

Using higher order basis functions to model electromagnetic scattering from the flat surfaces (FORTRAN)

- Compute impedance matrix of any arbitrary shape of the scattering object in order to predict the behavior of electromagnetic radiation
- Develop User interface (FORTRAN) to control matrix impedance

Data Acquisition System to monitor power consumption by the wide band three phase 25kW industrial microwave oven (C++ and Python)

- Design and build analogue network for voltage and current sensing, signal conditioning and power GRID isolation
- Design and build digital network to convert analogue signals into digital samples at Nyquist frequency of 200kSPS to detect the harmonics up to 100kHz imposed on the power GRID
- Develop Freescale K64F board Firmware (C++) to receive the acquired digital samples from ADC through the SPI channels
- Develop server (Python) at x86 terminal to receive captured ADC data over Ethernet interface using TCP/IP IPv4 protocol
- Data processing: Compute Fourier Transform on x86 and prepare results for the web server end point (Python)
- Configure Apache Linux server to host the End User website and develop website to display results for power monitoring within the analysis of higher harmonics imposed on the electrical GRID system