David Riley

Philadelphia, PA
T 443-824-2111

david@the-rileys.net
http://david.the-rileys.net

PROFILE

Embedded hardware/software/FPGA engineer seeks a challenging and engaging position developing products and architectures for embedded devices, desktop and server systems. Has been involved in many highly sophisticated (and many simple) software and hardware projects in a personal and professional capacity. Very quick to read code and learn new systems. Successfully blends a fanatical passion for electronics and computing devices with an outstanding work ethic and obsessive attention to detail to produce reliable, high-quality products.

EXPERIENCE

Technical Advisor to the CEO, Stream TV Networks; Philadelphia, PA 2014 - present
Performed many high- and low-level technology tasks for a glasses-free 3D technology development company. Responsibilities included: defining and developing new product and system architectures based on an assortment of original and licensed technology; developing interfaces, standards and protocols for connecting system components from multiple divisions, developers and contractors; introducing new hardware to manufacturing; managing and coordinating technology development and manufacturing teams in the US, the Netherlands, and Taiwan; updating marketing and executive teams on project status and working with them to define new directions for projects and products; and defining overall product direction and design within the overall company. Significant experience in reverse-engineering sparsely-documented inherited code in C and VHDL in order to create new documentation and models.

Chief Technology Officer, Anrim Technologies; Philadelphia, PA 2014 - 2016
Developed hardware for wireless vehicle telemetry system connected to vehicle OBD-II bus. Design was targeted for size and cost while still supporting legacy OBD-II buses. Contributed to design of overall system architecture, including user application, server backend, cryptography structures and database design.

Developed FPGA and embedded microcontroller code for video processing FPGA board. Embedded microcontroller was a small (32k), low-power ARM Cortex-M0 device which required a custom operating system and command execution structure to fit all functionality into the available code footprint while allowing for brick-proof updates. Worked with software team to develop specifications for functionality from customer specs, and directly implemented much low-level code (including operating system, command dispatch system, some device drivers, software update protocol/framework, and linker scripts. Managed port of existing customer FPGA code to new device along with a skilled FPGA team.

Operated servers for company including information wiki (Atlassian Confluence), issue tracking (Atlassian JIRA), continuous integration (Jenkins), file serving (NFS/SMB), virtual machine management, license server (FlexLM), VPN endpoint (StrongSwan), build servers and others.

Principal Hardware Engineer, Mantaro Networks; Germantown, MD 2006 - 2015
Developed many FPGA cores and complete systems-on-chip. Most FPGAs were interfaced to external processors through standard buses such as PCI, PC-104 (ISA), and Serial RapidIO as well as generic SRAM-style processor external buses. Cores included standard communications cores (such as RS232-style UART, SPI, I^2C), high-speed ADC capture cores, a scatter-gather DMA engine, and many adaptors for

memory interfaces. Worked mainly with Xilinx and Altera, including their system-on-chip development environments (Xilinx Platform Studio/SDK and Nios EDK, respectively).

Worked on many embedded microcontrollers and DSPs, including ARM, PowerPC, Motorola/Freescale DSP56k, Texas Instruments TMS320 DSP, Atmel AVR, and Microchip PIC. Most work was on bare metal (bootloader and lightweight OS/RTOS development). Performed significant work with U-Boot bootloader on PowerPC. Lots of experience with embedded microcontroller hardware, including communications interfaces, memory controller configuration for DDR and standard RAM, timers, and interrupt controllers. Lots of experience with reverse-engineering software from machine code on PowerPC and x86 using interactive disassembly techniques. Built small Forth kernel for wireless sensor node firmware stack.

Worked on many desktop/server software projects, including PCI and USB device drivers and end-user libraries for Windows and Linux. Developed mainly with UNIX make and Microsoft MSBuild, but had significant experience with Visual Studio as well. Performed 32- to 64-bit porting on Windows and Linux (including device drivers). Developed multi-platform build and unit/regression test system based on SCons and Python which included a web- and command-line-based dashboard for managing test runs and results.

Worked on a number of board design projects, including a high-speed ADC capture board (test equipment for satellite communications), embedded software reverse-engineering, automotive transportation management units, high-speed DSP AMC modules and wireless sensor networks (hardware and software) and military communications and air traffic control hardware and software. Key skills included power supply design, embedded and desktop software development, FPGA design, parts selection given many constraints, high-speed board design and low-power system design. Developed high degree of competency for parts selection and schematic entry. Participated in countless design reviews.

Contract work 2009 - present
Developed waveform generator FPGA to generate arbitrary waveforms for a 4 Gsps DAC. Project challenges included high-speed LVDS outputs to the DAC (48x 1 Gbps) and implementation of a parallelized hardware waveform generator producing 4 Gsps at 250 MHz. Additional, unexpected challenges included lack of direct connection to processor (forcing development of a hardware UART peek/poke debug unit and an intelligent inter-FPGA communications bus) and power supply synchronization with the output to the DAC. Most FPGA development was conducted in simulation, as only one instance of hardware existed and was not available for testing; this required good communication with the client team operating the hardware as well as a good simulation environment.

Many other smaller contracting projects, mainly involving small embedded software projects for tasks such as environment monitoring, laboratory instrumentation, and wireless sensor networks.

Lead Engineer, Pop'N'Ko Music and Entertainment; Baltimore, MD 2005-2006
Led software design (reuse and modification of an existing open-source engine) and electronic hardware design of an arcade dance game with a small startup. Hardware design included selection of computing resources to run the game and development of a capacitive touch-sensing foot sensor to detect dance steps. Initial hardware worked, but work on the game was discontinued when the distributor backed out under pressure from our competition in Japan (a much larger corporation).

EDUCATION

University of Maryland, Baltimore County (Baltimore, MD) BS in Computer Engineering, 2006

University of Maryland, Baltimore County (Baltimore, MD) MS in Computer Science, 2012

SKILLS

FPGA: Significant experience with Xilinx and Altera FPGAs in both standalone (system-on-chip) and peripheral capacities. Developed many standard IP cores (16550, smaller UARTs, SPI, I²C, SRAM

controller, NAND flash controller with ECC) interfaced to several SoC buses (including AMBA AXI and Wishbone). Developed scatter-gather DMA controller for use in a PC using bus-mastering PCI core from OpenCores. High-speed ADC and DAC interfaces (including 4 Gsps DAC interface with accompanying waveform generator). Highly comfortable with both VHDL and Verilog. Developed many Bus Functional Models for testbenching, including UARTs, Wishbone masters/slaves and CPU peripheral buses (particularly for the Blackfin). Primarily familiar with Mentor Graphics ModelSim, but some familiarity with Icarus Verilog. Considerable experience developing unit/regression tests in VHDL and Verilog/SystemVerilog using constrained random and data-driven test harnesses and scoreboards.

Embedded software: Considerable bare-metal development with ARM, PowerPC, M68000 series, Motorola/Freescale DSP56k, Atmel AVR, 6502, 8086 and Microchip PIC processors. Some experience with FPGA-embedded processors, including PowerPC, MicroBlaze and Nios II using FPGA vendor-supplied tools. Bootloader and lightweight OS (simple RTOS for instant-on devices) development. Significant experience with U-Boot bootloader and embedded Linux. Deep knowledge of embedded microcontroller hardware (UART/SPI/I^2C communication interfaces, timers, DMA units, interrupt controllers, memory controllers, cache controllers, etc.) Assembly-level (and occasionally hand-generated machine code) programming experience with ARM, PowerPC, 68K, DSP56k, 6502, 6809, x86, AVR, PIC, PDP-11 and MSP430. Familiarity with computer architectures ranging from PDP-11 and early 8-bit microcomputers (primarily 6502-based) through modern PCs, ATCA chassis, wireless sensor nodes and single-board computers. Significant low-power software experience utilizing the sleep modes of modern low-power processors. Primary embedded development languages: C, C++, assembler; familiar with GNU ld linker scripts and other esoteric elements of the GCC toolchain necessary for specialized embedded software design.

Desktop software: Windows and Linux kernel device drivers (PCI and USB) and end-user software/libraries for distribution to clients. Windows software packaging. Significant experience with Python for many tasks, including a fully-featured unit/regression test framework utilizing a web-based run management blackboard. Visual Studio, MSBuild, UNIX Make and SCons build environments, among others. Linux server management and application development (embedded, server-side and desktop). Significant amounts of Classic MacOS (pre-OS X) development, including porting classic programs to OS X and other platforms. Experience with threading and multiprocessing concepts and problems. Some experience with OpenVMS (VAX and Alpha) and earlier PDP-11 operating systems. Significant experience with operating system design at kernel and driver level, including developing kernel code on Linux, NetBSD, OpenBSD and Mac OS X as well as bare-metal operating system development for custom systems. Highly conversant with modern development cycle tools such as bug tracking systems (JIRA, Bugzilla, etc.) and continuous integration (mainly Jenkins).

Electronic/board design: Experience with high-level design taking low-level constraints into account. Experience with schematic capture using Cadence tools (Allegro Capture, Orcad), Mentor Design Architect, Altium, and KiCad. Very competent in component selection taking many parameters into account (e.g. op amps, ADCs, low-power processors, discretes). Familiar with high-speed (SERDES up to 10 Gbps and 10 Gbps Ethernet) and low-power design techniques. Familiar with many switch-mode power supply topologies and their design techniques; passing familiarity with magnetics design. Experience designing vacuum tube amplifiers. Considerable experience using SPICE to simulate analog circuits. Some VLSI design experience.

Other skills: Mobile development on iOS hardware (iPhone, iPad). Significant skill with OpenGL (including engine and shader development for both desktop OpenGL and OpenGL ES for mobile devices). Significant experience building networked software (primarily over TCP/IP using IETF Zeroconf for service discovery, but also using AppleTalk in the past). Some mechanical design experience (currently developing ballscrew-driven 3D printer controlled by PDP-11 and custom logic for Vintage Computer Festival exhibition).