

Lisa K. Barbay
3390 Vienna Dr.
Aptos, CA 95003
(831) 662-9829
(831)588-0191 (cell)
Email: lnelsonbarbay@sbcglobal.net
lisa@barbayconsult.com
Website <http://www.barbayconsult.com/>

PROFESSIONAL SUMMARY:

- Some FPGA experience (ACTEL, ALTERA). I am Currently working on ARM Cortex Altera Kit with Kiel debugger with my business partner who specializes in Verilog Design and I specialize in low-level software We are developing new product ideas and looking at the verification methodologies (VMM, OVM, OVL) and following current articles about the pluses and minuses of each methodology.
- Currently creating open source applications for Android (Eclipse SDK, Android SDK, API's) stepping through examples on-line. I have now written 5 applications in JAVA. I am currently writing a Notebook application.
- 20 + years written test plans, test benches, etc
- 20+ years written Firmware implementation Documents as an answer to the MRD.
- 20+ years ago writing regression test suites and tracking down bugs that were introduced into the system.
- 20 years board bring-up experience in the LAB. This is due to my methodology in written low-level functional APIs in Simulation and can then be used in first silicon testing.
- Currently stepping through each of VMM API C and Accelera 2.4 OVM APIs for white and black box testing.
- Bus-protocol testing (and random testing written in System Verilog. I am also looking at the VHDL version of OVL since VHDL was my first RTL that I learned and created models (I think of VHDL as the first Assertion Based Language. I am using these test API C to step through the System Verilog Specification and learn SVA.
- Currently learning Specman and the C language.
- RAL - I had to learn Register Abstraction Language (Synopsis) because my Python Register generation tool \ needed to output this format.
- Owner of Single Point Source Tool using Xerces XML Parser to define a chips register in XML and translate to many outputs (SPS written in C++).
- Member of the RDWG for (Register Description working group which tested IP-XACT 1.5) for compatibility with System RDL. - this has now been released.
- ARC Partner Program – now bought by Virage Technologies
- Worked on MEMS technology using the Cambridge Silicon Radio SDK (BlueLab 3.2 and Stereo-Headset-SDK)
- Spirit Consortium Reviewing Member for the Register Description Group (RWG--Spirit Schema, IP-XACT, and TGI (SOAP) experience)). I am using Python and it's TGI library.
- Strong XHTML/XML,XSLT Experience.
- Strong Experience programming in C and C++for engineering tools, behavioral models, & testing of various pieces of the designs (both ASIC and boards).
- Well versed in porting □ code across various hardware platforms and workarounds for various differences in compilers: GCC, Intel 80C196, NoHau 8051, Siemens C163, TI DSPs (included TI RTOS and Chip Support Library on the 67XX series), ARC, SparcLite, and ARM946ES., MIPs
- Experience with In-Circuit Emulators (Intel 80196, NoHau8051, Siemens C163, and TI DSP 2XX).
- Experience with VHDL/Verilog and VHDL/Verilog debuggers for behavioral models (BFM) and SystemVerilog.
- Programming Languages: C/C++; Java;Pascal; VHDL; Verilog; SystemVerilog; SystemC and various assembly languages.
- Writing APIs in C++ for embedded systems (with extremely small amounts of memory).
- Experiences with Serial Protocols RS-232, USB, and protocol stacks in RTL simulation and actual hardware.
- Experience with Serial interfaces SPI, I2C, I2S, S/PDIF .
- Scripting Languages: Python, C-shell, Bash, XML, XSLT, TCL, JavaScript, SOAP using Python, and Make.
- Some Perl Only for setting up scripting shells not developing applications in Perl.
- Linker Load Map Expert and ARM's Scatter Map.
- OS experience: Windows (NT, 2000, and XP), UNIX, and Linux.
- Editors/IDEs: MetroWerks (for both ARM and Pentium versions), Brief, VI, GVIM, TI Code Composer, Visual Dev Studio, Notepad++ and Eclipse for Java.
- 20 +years in bringing up ASICs end to end (from conception) to simulation (including) co-simulation and actually bringing up untested boards with the ASICs on them and writing the preliminary firmware.
- ASIC Simulators: VeriSim, ModelSim, Seamless (with VCS as the Verilog compiler, ARM tool chain, and

- XRAY as the debugger), and MetaSim (with ARC SeeCode Debugger, and ModelSim and the VHDL compiler and debugger) and DVE.
- Cambridge Silicon Radio (CSR) Blue-Lab, Stereo Headset VM and Kalimba DSP.
- I was a reviewing member of all technical working groups for Spirit Consortium which is now a standard with
- Accelera 2.4 which merged with IP-XACT 1.5 and contains SystemRDL by Denali. I have all this information – because of my Spirit Consortium membership.
- Strong Mentoring experience. I have taught almost every ASIC designer that I worked with. This frequently involves spending time helping them write their programs and use debuggers or teaching them the special pragmas for interrupt handling, packed structures, etc. I teach them debuggers. How to set break points, what a start-up file looks like, how to map file, etc.

EMPLOYMENT HISTORY:

January 2005-Present Barbay Consulting

- Worked for local video start-up (stealth mode) in the SW-Tools group:
 - Python Programming and using libraries like the matlab library and XML Parser library.
 - Took existing register generation tool written in **Python** and uses Dparser as the parser for the following using the free **Komodo Editor**). This is my third incarnation of writing the same tool:
 - Fix Bugs and learned PDB
 - New Features :
 - Added new output types: HTML Memory maps including SVG output, tables, RAL (Register Abstraction Language), dve_radix, Co-ware assembler output,
 - 'Cbit-field output, □ structure output.
 - Automated reset and read write tests (in □ for each module to be run on any of the environments (fast simulator, RTL, emulation box, and real silicon)
 - Many changes to Verilog output.
 - Added many new attributes to the register language to support many features.
 - Worked on FastSim Plug-Ins in C++ that simulated pieces of chip to be tested as a fast simulator of the chip for development of firmware. This particular chip contained 13 MIPS processor, custom Video DSP, and other features.
 - Worked on register files with engineers to improve feature set and help write more accurate register files (the output of which was used by the entire team).
 - Created the register files for all the in-house Synopsis IP (Design Ware) this included DDR registers, PCIe registers, I2c Registers, SPI registers, Flash Controller, and UART Registers. They needed the .VH, RALF and to write firmware drivers and test benches.
 - Updated System Verilog bus monitor that included an AXI Fabric, AHB, and APB bus to monitor my auto-generated tested and toggle GPIO lines appropriately that showed me the partial address of the failing register, the type (reset/rd-wr), and the passing condition. Modified to show values in Flash to further debug the test by evaluation the dve log file. These tests also have a second generation of write all writable bits address uniqueness value, the read them back. All these outputs and tested were written in Python.
 - Learned to run top-level simulations for DVE. Used dvrn for VCS running and saving of waveform files (vcdplus.vpd)
 - Use DVE waveform viewer to debug problems with the test/rtl.
- Converted DSP Kernels for video from Co-ware based assembler to gcc based assembler.
- Worked on video pipe for Local RTL Start-up company (Santa Cruz Imaging). The following tools are being used:
 - Cygwin/Bash Scripting/GCC modeling of pipe.
 - Visual Dev Studio for modeling of pipe.
 - ModelSim and VHDL for RTL development and modeling of the uP bus Interface (AMBA APB)
 - Used Modelsim Do files to emulate the amba bus and write tests through this interface.
- XML Parser for Register Description Language using Apache Xerces XML Parser and Dev-Studio
 Developed **SPS** tool for NuCore (Now Media Technologies – this is my own tool to sale. (all thought it can be compiled for Linux and Cygwin. This is a product that I sell and customize, <http://www.babayconsult.com/products.html>
 - Customized priority register description language in XML of my tool for a customer that had many outputs many other types of code (HTML, Memory maps, custom C++, R/W and Power on Testing writing in ARM assembly language). The very complex chip had 5 blocks, 58 modules and 2000+ registers.
 - Wrote papers about IP-XACT and became a reviewing member or Register Working Group.
 - Written presentations for customers about System on a chip development and Single Point Source.

- Consultant at Fullpower.com using MEMs Technology along with Cambridge Silicon Radio xIDE, VM, and the Kalimba DSP (BC55)..
 - Wrote 3 drivers for different accelerometers in I2C (in the MCU of CSR and the Kalimba DSP of CSR)
 - Worked on Bluetooth application for monitoring motion at Fullpower.
 - Converted C Motion Code using accelerometers into Kalimba DSP code. This included FIR filters, low/high pass filters.
 - Developed Hand-Shaking methodologies for FP on the CSR platform for message passing and interrupt handling.
 - Developed C like Data Structures to hold Kalimba information that I designed to closely match the original C code for easy understanding by a third-party.

September 2004 January 2005 Sabbatical

June 2001-September 2004 Senior Staff Engineer for **Creative Labs Advanced Technology Center (ATC)**, Scotts Valley, CA.

I served concurrently as a member of both the ASIC team(s) and Product Engineering. I worked on many audio projects that included S/PDIF, I2S, USB, priority DMA, SPI, I2C, various RTOS platforms, UARTS, and timers. I wrote many specifications, including an introductory class on C for the ASIC designers. This job included co-simulation, working with many IDE platforms, the ability to port tool chains across platforms, and bring up both firmware and un-tested boards.

All code is doxygenated (this is freeware-program which uses tags to produce online documentation, it is also was CVS (Source Control) aware, a take on JavaDoc). I was very instrumental in bringing many teams together who had not previously worked together (Audio Research DSP Engineering, Product Engineering, ASIC Engineering, and Firmware).

- Worked on Audio Streaming project on chip project which included the following:
 - ARM tool chain on UNIX and MetroWerks (for the API and board testing PC platform) in simulation.
 - Writing API C in Embedded C++ for Xdif (Creative □ S/PDIF), I2S, and a Stream Manager to use DMA to hook-up endpoints.
 - Helped create a Perl tool that generated HAL header files (which included bit definitions and register access functions in C++ and some Verilog defines).
 - Setup Seamless/VCS co-simulation and helped all the software people to use it (and a few Asic Designers).
 - Brought up much of firmware with co-simulation and then used same firmware on the board.
 - Continued to write new tests and correct problems with board. Brought up most of the chip in 4 days.
- Worked on Audio System project (proprietary) Most of this code was done in 'C' because TI67XX C++ compiler was very inefficient for the DSP-in place algorithms. This used a Creative developed FPGA and TI 67XX as both a DSP and system controller.
 - Worked with TI 67XX Code Composer and its Chip Support Library and RTOS.
 - Learned to use RTOS concepts (like semaphores, tasks, and mailboxes).
 - Created the initial repository (under CVS).
 - Developed the code for programming the McBsp (HI-speed serial bus on TI) and programming its DMA engine to get the raw-code from the I2S. Since the codec board TI gave us never worked, I developed a scheme to use the previous project C ARM based board and modify its firmware for testing the FPGA developed and created to get the raw audio data.
 - Developed ARM code for the previous project- this would generate various sine-wave data to test various features of the FPGA (soft-clipper, rate-control, etc).Developed the soft-clipper interface in C
 - Worked on initial part of the state machine for the push-button interface (then was moved to the next project), so a fellow engineer completed it.
 - Wrote a JELscript (TI scripting language) to simulate a button.
 - Also wrote a JELscript to program the EMIF (external memory interface) prior to another engineer writing a flash burner.
 - Worked on the message handling (message created by other team) which uses the commands to the ATC part of the project and created a dispatcher through a table of pointers to functions. To accomplish this, I created a Dev Studio project that would issue UART commands.
 - Modified Perl script to support the new register syntax and gives us auto-generated HAL files for the FPGA registers
 - Setup all SW, DSP, and Hardware Engineer's boards and JTAG interface in the early stages of the project
- Worked on Audio Streaming project with different ASIC team that used different methodologies- their basic means of testing was with TCL, I helped refresh their C skills (this was more of a mentoring process than a class).
 - Developed a single-point source system with XML/XSL and Xselerator tool. This included making C

HAL files that were Doxygenated, VHDL signal definitions, TCL access functions, and HTML register descriptions.

- Helped develop simulation to run USB drivers with a third-party USB host. This included many of the enumeration sequences.
- Worked on a flash-burner for the board (embedded C for flash) and Dev-Studio(on the PC) for the user-interface as the secondary bootstrap loader.
- Created a make file to run on all platforms and found a Saxon tool (this is XML->XSL conversion) for each platform to create the various outputs. Also made the make file flexible so that it builds for all the hardware and OS platforms needed (Simulation, Quickturn-box, and the actual board).
- Created a web page with register definitions . This grew into a full-blown website that has all the documentation we needed to program the chip, including compiler documentation and other 3rd party IP. Everyone was involved and each person thought of something new to add to the website.
- Co-Developed a nightly regression script to run the regressions in c-shell script. A SourceForge mailing list delivers the results every night.
- Wrote many APIs and HALs in embedded C++ specific to our chip.
- Got the ARC compiler working on Unix, Linux, and the PC
- Got the Co-Simulation environment working the ARC, ARC SeeCode Debugger, MetaSim (co-simulation tool using FLI on Windows XP and PLI on Linux) and ModelSim.
- Set up 5 people with boards when the chip came in and helped them with setup. Most of chip is functional in 3 days.
- Wrote a primary bootstrap loader that was turned into gates. Tested this in simulation and on the Quickturn box with a UART tool, IVT (freeware).

1996 to Jun001 Senior Staff Software Engineer for Sierra Imaging/Conexant. Member of ASIC team.

Sierra Imaging was a digital camera company that offers a complete digital camera solution. Responsibilities included system design and development of ASIC (not implementation), testing of ASIC that included test bench modeling, functional testing, Reference Design Kit (RDK) board testing to validate ASIC hardware. I developed image-processing firmware for various implementations of digital cameras. Worked on various sections of the chip which provided the following functionality:

- Serial transfer protocols (USB, RS-232, Appletalk, IRDA).
- Image Processing Chain (Capturing from CCD/CMOS sensor, interpolation, JPEG compression, decompression).
- Dram testing.
- Sleep Management.
- Interrupt Controls.
- Memory Controller Testing.
- Video playing for various formats: LCD, PAL, and NTSC.
- Developed the following simulations in C for validation of the Chip - "Frame Capture Processor, DSP (below), Buffer Simulators, etc.
- Developed a CYCLE and BIT accurate model for our in-house SIMD. This was very complex and I developed an algorithm which allowed Bayer input through the system to the DSP (setting up DMA, the DSP, etc) and in the final output was a JPEG image. We took a quarter of this image and ran it overnight in Simulation on the real RTL and it compared exactly and we could see the image!!!. This was later developed into a debugger since there was no visibility in the DSP.
- Developed VHDL model for various transfer protocols: RS-232, Appletalk, and IRDA (slow and fast).
 - I Implemented physical transfer for various protocols.
 - Developed various behavioral models to validate ASIC behavior (simulators). These were written in □
 - Developed tools to prototype image processing. These were written in □
- Developed all software support needed to embed the ARC microprocessor. This included all start-up code, and interrupt vectors. This was used in both simulation and the actual hardware board.
- Wrote tests in C that ran during simulation of the ASIC. ModelSim simulator was used.
- Board Bring-Up when new Silicon came from the Foundry:
 - Used ARC SeeCode debugger to bring up new RDK boards and validate ASIC functionality.
 - Ported Camera firmware to run with new ASIC DSP hardware and provide new feature set.
 - Developed image processing firmware for various camera projects.
 - Developed middle ware movie playing code (the chip was never intended to play Movies. We used QuickTime. I worked on record and playback.
 - Developed image disc allocation for the different size levels.
 - Developed Decimation for LCD.
 - Developed Pixel Correction Code.
 - Developed Gamma Curve and validation routines in the DSP and for the 32 point programmable curve.

- Developed Bayer Interpolation Code.
- White Balance Support.
- Alpha Blending (part of the SoC) and I implemented it on the RDK with Software programming the Alpha Blending.

Sept6-1996- Barbay Consulting

- **Sierra Imaging** - Tested custom ASIC to use in digital cameras. Wrote tests in C and used ModelSim for VHDL simulation & debug.
- **Seagate Technologies**- Tested custom controller chip for SCSI drive. Tests focused on memory devices.
 - Wrote tests in C and used ModelSim for VHDL simulation & debug.

Jun '90 to Sept6 - Senior Staff Software Engineer for **Seagate Technologies**, Scotts Valley, CA. Member of ASIC test and integration team. The ASICs developed are the controller chips on Seagate Drives (both AT and SCSI). Highlights of projects include:

- Developed VHDL model of an 80196 processor for test of ASIC design that takes in .hex data generated by the 80196 C compiler.
- Wrote diagnostics written in C that are run during simulation to test the ASIC. Used Mentor Graphics QuickSimII to run simulations. Used ModelSim for the PC to debug the VHDL model. It used variables and procedures – which at the time were for behavioral models only.
- AT drive firmware written in C (for 80196, Siemens C163 and DSP TI 320 processors). This firmware tested features of the chip, which included Reads and Writes to disc, based on AT Host commands.
- Tools written in C with MS-DOS Windowing Interface. Tools include control of disc emulator and paddle board with RS-232 interface protocol implemented.
- Tools written in C which included a suite of assemblers, a pre-processor, 2 pass assemblers .
- Tools written with a GUI for Headerless registers based on Timings. This helped people to calculate the register contents (and produced a header file for programming the registers.)
- Tools written in C that runs both on UNIX and MS-DOS. These include: a sequencer assembler, dram translation modeling, error correction modeling.
- Real-time firmware for error correction used on both SCSI and AT drives in many different assembly languages.
- Wrote a C document and gave a 10 session training class to the ASIC designers. This also involved a lot of mentoring and helping people write tests for their modules.

Jun '89 to Jun '90 - Software Engineer for **Epson Technology Center**, Santa Clara, CA. Developed Source Control Tools in C using PVCS as backbone and specified and modeled an on-line time card and project schedule tool.

Sep8 to Jun'89 --Firmware Engineer for **Phoenix Technologies Ltd.**, San Jose, CA, a leading vendor of IBM PC compatibility firmware (BIOS). Developed a WD3600 chip set BIOS.

- Wrote the specification for developing the firmware in-house model for all specifications of OEM BIOS implementations.
- Team leader of development project for WD3600 chip set, which included writing 80286 assembly code and use of Intel 286 ICE.

Aug '86 to Sep '88 -- Technical Response Engineer for **Zenith Data Systems**, St. Joseph Michigan.

- Developed trainings for the Field Applications Engineers. This included bound documentation that I wrote about a new machine, OS, etc.
- Gave Trainings to FAE and larger OEMs.
- Extracted the Strings from many of Zenith products so that the the programs could be developed for other spoken languages. Prior to this, error and other type messages were printed from the code. Now they could change out the new header file for whatever language was being developed.

CONTINUING EDUCATION CLASSES:

- 3-day Vera Training class taught at Synopsis, and Introduction to Object Oriented Programming and Languages UC Extension Class.
- VHDL / UC Extension Class.
- C++ / UC Extension Class.

EDUCATION: 1986 -- Bachelors Degree, Computer Science, Louisiana State University