**NASA Undergraduate Student Research Program (USRP)**

**Spring 2011 Project Plan**

Hieu Nguyen

Jet Propulsion Laboratory

California Institute of Technology

4800 Oak Grove Drive

Pasadena, CA 91109-8099

hnguyen913@gmail.com

1. **Introduction/Background**

In spring 2011, I will participate in a 15-week research program at NASA’s Jet Propulsion Laboratory (JPL) in Pasadena, CA. This is my first NASA internship, and I will be working under the mentorship of Dr. Yutao He, a Senior Member of Technical Staff at JPL. He is currently researching advanced avionics technology for future flight spacecraft and instrument missions, and has interests in FPGA-based reconfigurable computing. To support Yutao’s research interests, I will complete a project focusing on these embedded systems topics.

My research will primarily involve the development and implementation of digital signal processing (DSP) algorithms on ISAAC-based platforms. The ISAAC—**I**nstrument **S**h**A**red **A**rtifact for **C**omputing—project aims to provide a “highly capable, highly reusable, modular, and integrated FPGA-based common instrument control and computing platform that can be shared by multiple Earth Science and Planetary Exploration instruments [1].” Essentially, ISAAC allows instrument electronics designers to configure a complete instrument control and computing system by using its integrated framework.

The motivation for ISAAC technology is to provide a platform that can be *shared* by a wide range of instruments, each with its own various requirements. ISAAC’s reusable framework offers a combination of “adaptability, computation power, I/O bandwidth, digital interface standards, and data processing capability in a single common low mass/power and small form factor platform with significantly reduced non-recurring cost and risk to Earth Science instruments [2]”. As future instruments require more demanding on-board processing capabilities, ISAAC will allow instrument development to meet performance requirements in a cost-efficient manner.

A proof-of-concept prototype (the “*ISAAC I*”) has already been created, and has been utilized in the development of scientific instruments such as the SMAP L-band radar. The *ISAAC I* consists of a Xilinx Virtex-4 FPGA device with ISAAC-developed digital filter and function-fitting IP cores. *Although the ISAAC I has had successful application to the SMAP radar, there is more comprehensive development of ISAAC-planned capabilities to be completed.* My efforts at JPL will work to improve the ISAAC framework and increase functionality for the advancement of instrument development.

1. **Objectives**

My research project will involve using the ISAAC framework to perform system architecture exploration and design, algorithm development and simulation, FPGA implementation and testing, and integration and benchmarking. ISAAC consists of six key components: iBoard, iBus, iCore, iPackage, iBench, and iTool. I will be working to expand the ISAAC iCore library, which consists of “standard and parameterized IP cores that implement common computationally-intensive instrument control and computing functions [1]”. Existing IP cores include the Multi-Pass Wide Kernel Fast Fourier Transform (MPWK-FFT), the Multi-rate Finite Impulse Response (MRFIR), and quadrature demodulation. Using these IP cores as a starting point, my primary objective is to successfully develop and implement additional DSP algorithms and IP cores to expand the iCore software library for ISAAC-based technology.

Whether or not the algorithm design is successful will be determined using ISAAC’s iBench, a benchmark suite used to verify performance requirements. After the iCore library is configured in an FPGA-based data processing hardware, iBench can validate and fine-tune the overall performance of a specific ISAAC configuration. The benchmark suite consists of validated historical instrument data streams from different types of instruments (such as radars, radiometers, and imagers), and a common test-bench architecture that allows integrated functional testing and validation of ISAAC-based instrument digital electronics system. This simulation tool will greatly aid in debugging and optimizing my algorithms.

I would also like to accomplish other parts of the general project description if possible. One task is to develop a Java-based GUI software toolkit that streamlines the FPGA-based digital system development process from high-level modeling/simulation to RTL-level Firmware implementation. Another interesting task is to develop a flexible, modular, configurable Python-based software kit that allows control and monitoring of FPGA-based digital systems of various instruments, and acquisition and display of science and engineering telemetry of digital systems. These tasks would give me valuable insight and exposure to other embedded systems projects, which will be a strong field of study when I attend graduate school next year.

An overall objective of my USRP experience is to learn as much as I can—about research, engineering, and about the work-life at NASA. I hope that I can meet and get-to-know many other interns, scientists, and engineers at JPL. Beyond acquiring technical skills with hardware and software development tools, I would like to attain a memorable, life-changing experience as a USRP intern. This is a just a tentative list of objectives, as new goals may arise to replace old ones after the start of my research project.

1. **Approach**

* Outlines the approach you will take to carry out your task

I have 15 weeks to accomplish my project. Thus, I must prepare a schedule that will allow me to complete the assigned project and have significant time to write the final report. Principal steps include familiarizing myself with existing code and simulation/testing procedures, researching assigned DSP algorithms, implementing said algorithms at the Register-Transfer-Level in Verilog, and simulating/testing the design.

*Specifically, how will you reach your objective or produce your desired final product?*  
I have never worked with ISAAC before, thus I will need to learn how it works. Training… understanding existing software, design theory. I would like to spend the first two weeks getting situated with my project, to feel comfortable about moving forward with new developments.

In order to achieve my first objective of implementing DSP algorithms, I will have to understand the existing software algorithms, which are written in Verilog (initial design). I will be working with existing Verilog code for SMAP’s digital filter, and porting over a floating point representation to fixed point representation in Verilog. I will first get existing code compiled on the FPGA and understand the intricacies of how the code works. After successfully running the code, I will run tests to verify performance. Then I will start do develop new DSP algorithms in Verilog, while simulating my design using Xilinx ISE/EDK and Matlab. Using Xilinx ISE/EDK, I will… And then using Matlab, I will simulate the algorithms and put them on a custom JPL board.

To expand the iCore software library, I will first examine the iCore-dsp functions, which are data processing functions common in instruments such as digital filter, Fast Fourier Transform (FFT), Finite-Impulse Response (FIR), Radio-Frequency Interference (RFI) detection and mitigation, presum, block floating point quantizer, convolution, spatial correlation, data compression, and image feature extraction.

Simulation and testing

Use iBench (the suite of benchmark instrument data steams for performance validation and tuning of a completely-configured system) to develop for iCore

I will also be using iTool, the integrated tool-chain providing a familiar and end-to-end design flow for digital system designers

To measure the comparative advantage of TD FIR over polyphase decomposition, a performance test is devised using Xilinx-VR.

*What steps promise to be the most difficult, and how will you overcome the difficulties? What equipment or other resources will you need? Which of these are inherited, and which will you have to make or procure?*

The most difficult part will be figuring out how the algorithms are executed by the FPGA (writing the code) because I only have experience determining the DSP algorithms’ outputs using mathematical formulas, disregarding how the answer would be obtained from a computational standpoint. I will overcome my ignorance by going through various online tutorials and a lot of literature research. The equipment I will need is a computer for writing code and an FPGA for executing the code. I have my own laptop with the necessary software tools installed (Xilinx, Matlab), but it may not be powerful enough to run heavy computations in simulations. I do not have the FPGA device or other hardware; I will need to procure them from my mentor. From what I know, this project does not depend on results from other people in related projects, and does not require active collaboration in the design phase. However, it may involve groups in Earth Science (such as SMAP instrument) to perform testing and determine performance parameters.

I will document my work as I go along. If I get stuck, I will seek help from my mentor, colleagues, or peers. If I run into trouble, I will ask for help (opencores.org, mentor)

1. **Project Schedule**

* Provides a schedule or timeline for accomplishing the individual steps and overall goals of your project

Milestones:

* Get situated with ISAAC project
* Implement DSP algorithm(s)
* Test and simulate IP core(s)
* Paper and presentation
* Other embedded systems project(s)

*What are the principal steps or milestones along the path? How long will each take?*

Milestones… principal activities and events to show that I have taken a systematic approach to planning my work. This is a tentative (rough) schedule, as details uncovered during the course of the internship will bring about new tasks and goals that are more detailed.

At JPL, USRP is a 15-week program during the spring from January 24, 2011 to May 6, 2011.

|  |  |
| --- | --- |
| Week # | Goals |
| 1 | *Get acquainted w/ project, training* |
| 2 | *Learn software tools, hardware protocols, ISAAC, digital filter* |
| 3 | *Start DSP algorithm* |
| 4 | *Finish DSP algorithm* |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 |  |
| 9 |  |
| 10 |  |
| 11 |  |
| 12 |  |
| 13 | *Other embedded systemsexposure* |
| 14 | *Final presentation* |
| 15 | *Final report DUE* |

1. **References**

My mentor, Dr. Yutao He, has counseled me on this research project, providing a general project description and relevant papers on the ISAAC framework and its applications.

My literature sources include the following:

[1] Y. He, C. Le, J. Zheng, K. Nguyen, and D. Bekker, “ISAAC a case of highly-reusable, highly-capable computing and control platform for radar applications,” in *IEEE 2009 Radar Conference*, May 2009.

[2] K. Nguyen, J. Zheng, Y. He, and B. Shah, “A High-Throughput, Adaptive FFT Architecture for FPGA-Based Space-Borne Data Processors,” 2009.

[3] J. Zheng, K. Nguyen, and Y. He, “Optimized FPGA Implementation of Multi-Rate FIR Filters through Thread Decomposition,” 2009.

A prototype has already been made that is comprised of six key components—an FPGA-based hardware substrate, a hardware/software interface standard, software functions to implement common computationally-intensive and non-computationally intensive instrument control and computing functions, a benchmark suite, and an integrated tool-chain for system design and software development.

It is divided into three groups at the top level: *iCore-ctl* - implements control related functions such as commands handling, telemetry collection, and timekeeping; *iCore-ft* – implements fault tolerance functions such as detection of Single- Event-Upset (SEU)-induced errors;

ISACC Description

ISAAC’s unique technical innovations are embodied in its six key components: iBoard - the FPGA-based hardware substrate; iCore - the library of Register-Transfer-Level (RTL) Intellectual Property (IP) cores implementing common computationally-intensive instrument control and computing functions, such as the MPWK-FFT presented in this paper; iPackage - the collection of software functions that implements common non-computationally-intensive instrument control and computing functions; iBus - the standard and unified hardware/software interface; iBench - the suite of benchmark instrument data streams for performance validation and tuning of a completely-configured system; and iTool - the integrated tool-chain providing a familiar and end-to-end design flow for digital system designers

Based on the ISAAC I prototype

Hardware: iBoard (a single board computer (SBC) featuring a high-performance Xilinx Virtex 4 or Virtex 5 FPGA chip with two embedded PowerPC405 processors and over 8 million reprogrammable logic gates and provides the complete set of key instrument control and computing capabilities)

***iBoard***: is a Xilinx ML410 board featuring a Xilinx Virtex 4 FPGA device that contains two embedded PPC405 CPU hardcore and up to 8 millions logic gates programmable fabric. It also provides an IEEE 754 single-precision FPU softcore, 256 MB DDR2 RAM, up to 2GB SystemACE CompactFlash, two 10/100/1000 Mbps Ethernet ports and UART ports, one Interrupt controller, DMA engine, watchdog timer, and I2C RTC and temperature sensors.

* Xilinx FPA with 2 CPUs and 8 millions programmable fabric
* EDAC-protected NVM and SDRAM

Common instrument control/computing interfaces (LVDS/RS422/1553/SpaceWire/I2C)

WATCH ON YOUTUBE:

* Xilinxinc
  + HDL language tutorials
  + FPGA tutorials
  + Embedded systems tutorials
* FPGA tutorials
* ISE/EDK tutorials
* Opencores.org