SOHEIL SHABABI

www.shababi.us | shababi@usc.edu | (323)449-9044 | 3025 Royal St. #Apt 344B, Los Angeles, CA 90007

Objective: Seeking for fulltime Software/Hardware positions

EDUCATION

University of Southern California, Master of Science in Computer Engineering. Los Angeles, CA May 2019
Courses: Digital System Design (EE560), Advanced Computer Architecture (EE557), VLSI System Design (EE577b, EE577a), Diagnosis and Design of Reliable Digital Systems (EE658), Computer Aided Design (EE681), Mathematical Foundations for System Design (EE599), Computer System Organization (EE457)

Shahid Beheshti University, Bachelor of Science, Computer Engineering, Tehran, Iran May 2017
B.Sc. Project: Hardware Acceleration of Image Edge Detection on FPGA-SoC platform using OpenCL, demo: http://www-scf.usc.edu/~shababi/OpenCL.html

WORK EXPERIENCE

AI Processor Prototyping Intern, Deep Vision Corporation, Los Altos, CA May 2019 - Aug 2019
- Verification of a deep learning processor and building system level solutions for automotive applications
- FPGA prototyping and emulation of a multicore AI chip
- Mapping deep learning workloads to the FPGA platform to analyze performance bottlenecks

Hardware Design Engineer, Naminic Corporation, Tehran, Iran May 2015 - Sep 2016
- Designed and implemented a CNC Controller on FPGA Platforms, using NIOS2 ipcore
- Developed application layer of Network Management Software which connects a network of ARM Processors together for Laser-Tag products, using Qt platform

TECHNICAL SKILLS

Processors Knowledge: Out of Order (Tomasulo), Multicore (CMP), VLIW, Superscalar, SMT(HTT), Vector Processors
Verification Knowledge: UVM, Functional Coverage, Constrain Random Verification, System Verilog Assertions, Formal Verification
Test Knowledge: DFT, Boundary Scan, ATPG, Fault Simulation

Data Transmission Protocols:	**Cache Coherency Protocols**		
PCIe, AXI, APB, DDRx	MOESI (Snooping Based), ccNUMA 3-Leg (Directory Based)		
Hardware Description Languages:	**Scripting Languages**:	**Software Development Languages**	**Version Controls:**
System Verilog, Verilog, VHDL, OpenCL	Python, Bash, Make	C/C++, Java	Git, Github

Tools:
ASIC 45nm design flow: Design Compiler(Synthesis), Innovus(PnR), PrimeTime(STA), Cadence Virtuoso(Layout), ABC(Technology Mapping, Logic Optimization)
Simulation: NCSim, QuestaSim, Modelsim, Isim, HSpice FPGA design flow: Vivado, Xilinx ISE(Nexys4), Altera Qsys/Quartus (DE2, DE1SoC)
Architecture Design Tools: SimpleScalar, Cacti, PIN

PUBLICATION

Deep Learning-based Circuit Recognition Framework using Sparse Mapping and Level Dependent Decaying Sum Circuit Representation
Arash Fayyazi, Soheil Shababi, Pierluigi Nuzzo, Shahin Nazarian, Masoud Pedram – Accepted for DATE 2019 Conference, Florence, Italy

ACADEMIC PROJECTS

- **NoC: A Chip Multi-Processor(CMP) in a Ring Router NoC from RTL Design to Layout**. Connected 4 processor cores in a ring topology network on chip environment by implementing router and network interface component (NIC). Performed RTL coding (Verilog), RTL simulation (NCSim), synthesize (DC Compiler), post synthesize simulation (NCSim), place and route (Innovous), post PnR simulation (NCsim), and Static Timing Analysis (PrimeTime) on top level Chips Multi Processor(CMP). Achieved clock period of **5.12(ns)**, **11.4 (ns)** and **15(ns)** for pre-synthesis, post-synthesis and post-PnR phases, respectively, and layout area of **686000 (um2)** in 45nm technology.
- **PCIe: A 2 lane-2 link PCIe architecture on Xilinx Artix-7 FPGA**. RTL design and synthesis of "Elastic Buffer" to detect ordered sets (SOS), "Deskew Buffer" to adjust delays between lanes, an "Decoder 10b/8b" to adjust running disparity and avoid DC coupling noise. FPGA prototyping verification by measuring relative skew between lanes and adding trigger conditions in Chipscope ILA.
- **SRAM: 1Kb 6T SRAM with 2 Banks in Cadence Virtuoso.** Applied Super Buffer to adjust input capacitance of SRAM address decoders. Utilized pre-decoders to accelerate address/column decoders. Designed Sense Amplifier, Precharge circuit, SRAM cell, SRAM Banks, Address/Column Decoders, Read/Write Circuitry. Archived frequency of 1 GHz in 90nm CMOS technology.
- **UVM: UVM based Constrained Random Verification of AMBA APB Slave.** Designed Sequence Generator to produce TLM Sequence Items. Designed Driver to convert TLM sequence items to APB pins signals. Designed Monitors to convert APB pins to transactions.
- **AXI: Advanced eXtensible Interface (AXI) protocol with five channels.** Developed AXI master/interface modules for connecting 4 processors to 4 memories in a 2x4 mesh interconnect with deterministic routing logic "x first, y next". Managed out of order packets using reorder buffers. Implemented TCL scripts to verify functionality of AXI protocol by comparing output result with golden files.
- **CAD: Designed and implemented a VLSI Placement CAD Tool** optimized for low-power and high-speed technologies such as AQFP, using C++ and shell scripts. Developed a detailed placer for level-based placement. Solved a Linear Assignment Problem by Hungarian Algorithm to find optimal placement for each cell. Used global placer to minimize wire length, and legalization to remove overlaps between cells.
- **OoO: Out of Order (OoO) Tomasulo processor** which performs speculative instruction execution beyond branches by using dynamic instruction scheduling and dynamic branch prediction using Branch Prediction Buffer (BPB) and Return Address Stack (RAS). RTL design, synthesis and FPGA implementation of modules like reservation tables (to avoid CDB conflict), dispatch unit, FRAT, RRAT, FRL, PRF, and CFCs.
- **CMP: Chip Multiprocessor (CMP) with a 7-stage 16-threaded CPU** (4 Thread 4 Core – 4x4), L1 private cache and L2 shared cache. Maintained Cache Coherency and sequential consistency in L1 cache of 4 cores by implementing non-blocking cache with MOESI protocol, SCU, CCU, MSHR,WSHR,FSHR, SC, LL, and Cache Maintenance instructions. Accelerated matrix multiplication by 16X by utilizing 4 cores.
- **CDC: Cross Domain Clocking (CDC) in a dual clock FIFO** with double synchronization FFs and gray codes to control metastability in a CDC environment. Applied a dual ported Flow Through and Pipelined SSRAM for FIFO memory, and a small register based memory as internal FIFO of consumer with FWFT (Fast Forward Fall Through) technique. Performed FIFO width and depth expansion, using Ping Pong method.

CERTIFICATIONS

- **TCL Programming from Novice to Expert**, Udemy, Certification Link: https://www.udemy.com/certificate/UC-3DYSCLRW/
- **Learn to build OVM & UVM Testbenches from scratch:** https://www.udemy.com/learn-ovm-uvm/
- **GNU Programmer (with focusing on makefile, gdb debug)**, Udemy, Certification Link: https://www.udemy.com/certificate/UC-8VYAWMCQ/