Ilia   (greenblat@mac.com) CV

I design RTL and play with EDA.

Overview

Been in VLSI business since 1983. Started in Motorola Semiconductor. First years was what was called Circuit design. Transistors and gates. Main simulation was with spice. Then moved to first editions of synthesis. Designed programmable sequencers and many small modules. To aid design process, developed several tools and methodologies to augment Cadence/Synopsys offerings.
Took part in several tapeouts, among them chips that are produced till today. Had fun in Motorola for 17 years and two of those spent in Austin, Texas.

In 2000 moved to Virata. There designed switch fabric (rings) and threads processor that did the data movements. Did the integration of top levels.
The chip worked flowlessly (i had considerable stake in that). Perhaps even had customers. However, new owners ditched Israeli office.

Since then i work as a freelance. Partially because got fed with corporates. Three of us, toyed with start-up. For six months we prepared pitches, met with VCs, designed the product and basically went through crash course MBA.
One day we pitched our idea to IR. They got interested, but asked me to design a module for them.
I got an offer to work for International Rectifier as their digital design center. Worked there for 10 years (2004 - 2014). Income Tax authorities knew i was contractor, but everybody in Los-Angeles office thought of me as an employee.
IR needed a digital controller for their high voltage / high current motor drives. We designed programmable controller that mixed some analog (ADC, op-amps, bandgap) with pure digital content. It went through several generations and sold millions. Chinese, Japanese and some European washing machines may still use it. ACs too. Look for iMotion family in Infineon. The analog content came from Pavia, Italy office. We did all the flow, from architecture, through design, synthesis, place and route, DFT, signoff tools and upload of GSSII to TSMC.
Thus i know something about everything. The IR team were masters of selling the stuff. While our controller went for 50 cents a pop, IR augmented the offering with pre-drivers, drivers and IGBTs/MOSs to total of around 10 USD.
This arrangement ended when, another digital group was aquired by IR. They got rid of us, only to be fired when IR was bought by Infineon.

With IR, i had privilege to do all stages of digital and mixed level design and simulation. Oren (my partner) did the FPGA validation. Courtesy of IR we had real motors spinning in the lab.

Since then, i work mainly with Israely startups. Mainly designing RTL modules for their SOCs. Sometimes do special simulations they find hard to do.

Customers (partial list)

Pixim (now Sony)

- compression of black image. algorithm from the company.

Inuitive

- implemented SLAM engine, as DOG and FREAK , including floating point processing.
- several picture improvements vision algorithms (implementation of given algorithms)

Magic Leap

- designed programmable DMA (AXI to AXI) that is mainly used as a CPU that gets triggers from sensors, cameras and such and shapes and moves data.
- designed floating point processor that recognizes onset of speech. (Algorithm in "C" we got some Phd).
- implemented several vision algorithms (i am not an expert on algorithms, got the spec and built it).

Compass EOS

- Modeling of their switch chip at system level, to find bottlenecks in architecture.

recenty, Two american companies

various modules and fpga modeling. too recent to disclose.

recently, Israely company in area of vision sensors.

```
Simulation of netlist coming from spice. To prove connectivity is correct.
```

Autotalks

- Plug: is module that does encryption and ECC on AXI-to-AXI bridge. It is actually ours to re-sell.
- Filter module for autotalks.
- AXI firewall module.

Trieye and Ingonyama

- supplied and maintained the infrastructure for python driven verification.

Classiq

- they took my open-source schematic editor and paying me to adopt it to quantum computing.

Neologic

- implemented testing code for their speciality stuff. It includes debug-Uart, which talks to outside world and can on command exercise their stuff. It is a uart followed by state machine. No CPU involved.

Tools (all are MIT licensed open source)

verilog parser and manipulator

- reads verilog files (based on lex and yacc)
- gives the user option to add python processing stages
- optionally dumps out verilog after changes or python generated reports.
- used extensively by all my projects. Has numerous built in actions. Needs special doc.

Python driven verification

- methodology and implementation of replacement of SV-UVM and Specman. Not a CoCoTb.
- used extensively by all my projects.
- couple of companies adopted this flow.
- built library of VIPs for AXI, JTAG, APB, I2C, SPI and many more.
- has PDF and other docs in my gitHub.

translators

- Spice to verilog
- recognizers of transistor structures in verilog from spice (mos to gates).
- GDSII to text and back
- cadence DEF to verilog.
- basically any to any.

register file generator

- text to RTL generator. Adopted by at least one Chinese company. I used it in many projects and sold version of it to FPGA design vendor.

instruction set generator

- given description of instructions, produces rtl of the decoder, files to drive assmebler and documentatiom.

synopsys liberty parser

- translates liberty format to normal verilog and other formats.

network on a chip

- generates axi network on a chip.

lymphatic network on a chip

- generates serial network on a chip as a side channel for control and monitoring.

gitHub repository

- public repository with all the software described. Additionally library of rtl modules, floating point library and more.
- Not always keep documentation up to date. Ask and i will fix specific issues.
- git clone https://github.com/greenblat/vlsistuff.git

mentoring gitHub repository.

Intended to help recent graduates and especially former armed forces to get a grip on RTL design process.
The request came following shortage of rtl designers and verificators on one hand and hurdles for recent graduates to start working. My slogan was "Until Apple grabs you, do some design challenges and help me too."
It had limited success. But about 5 people got to work partly because of this. None in Apple though.
All designs are snippets from my real projects.

- git clone https://github.com/greenblat/vlsimentor.git

Open Source Tools

I try to use best open source tools that i can find.

- Yosys : synthesis, not so great, but usefull sometimes.
- OpenSTA : STA that kinda works
- iverilog : icarus verilog simulator. My main simulator.
- verilator : sometimes used for fast simulation and also as package for giving software teams a working model.
- gtkwave : wave viewer. just great. (sometimes use dinotrace)
- ngspice : sometimes i get nostalgic on spice and especially their add-on xspice.
- dot : graphviz : best for visualization and presentations.
- typora (.md) files - documentation.
- Python, C and all Linux / MacOS tools.