SOCGEN Project

Manifesto

The semiconductor industry is in the process of undergoing a complete metamorphosis that will change forever the way digital electronic products are designed. The last time this occurred was back in the 1990's when schematic capture was a popular tool for designing ICs. But designs grew to the point where you couldn't plop down a gate and and connect it fast enough to create and verify a net list in a reasonable amount of time. So schematic capture has been replaced in all but the simplest designs with a HDL/SYNTHESIS flow.

That happened about 15 years ago and despite constant predictions of the demise of Moore's Observation we now find ourselves back in the same situation, You cannot type fast enough to create and verify a net list in a reasonable amount of time.

The solution that is now emerging to meet this growing demand is Design For Reuse(DFR). This methodology recognizes that the vast majority of design work is simply a rehash of work that someone else has already done. If you could reuse and leverage preexisting code then the amount of new design work needed drops off dramatically.

The SOCGEN project was created to provide a free open-sourced Design for Reuse tool set for asic and fpga developers. It will also provide industry standard best practice guidelines that show how to create a reusable design as well as a series of IP modules and designs to demonstrate the proper way to make a design.

Everything in the SOCGEN project is released under GPL3 as free and open source. SOCGEN will partner with other opensource or free-to-use tools to form a complete embedded systems tool chain that includes IC design, PCB design, Firmware development and real time debugging.

Why am I doing this?

Theres an old saying that everyone has at least one good novel in them. For an engineer it's at least one good open source project. I spent my career as a electrical engineer working for Hewlett-Packard. Nothing in my years in R+D could prepare me for what we had to do to create deep submicron asic designs. The chip sizes and complexities grew every year and it was a struggle to keep up with all the new processes. But I had spent several years working as a production engineer mostly bringing new products up the production ramp and the techniques used in manufacturing were exactly what were needed to design DeepSubMicron asics.

R+D engineers tend to work like ye olde Yankee craftsmen. They go into their shops and spend a lot of time handcrafting a design and it is beautiful when finished. But it also takes a lot of time and the demand is such that you will soon need every adult male in New England working in the shops to meet production quotas. It's time to part with the traditional ways and create a high volume manufacturing line for asics.That is the goal of Design for Reuse.

I was fortunate to work at HP in a design group that was doing bleeding edge development of design for reuse techniques. The Inkjet Peripheral group had a IC design team that spanned four divisions and a support operation. We did 2-4 asic designs a year and did this year after year after year. We designed families of chips doing one for low-end, high volume , one for high end, high performance and if the vintage chart called for it , one in the middle.

We designed platform asics. There were to many products to do one chip per product so the chip that went into an All-In_One printer scanner could also go into a Photo printer or a stand alone printer. The extra silicon needed to support unused functions was nothing compared to the cost savings from combining product volumes into a single part. Pin's on the other hand were precious. The pins needed to drive the scanner interface were multiplexed with the Photo card pins as well as GPIO pins. The code would configure the chip for whatever product it happened to find itself in.

We also did multi-Vendor designs. Our volumes were to high to only have a single source. Plus we wanted the pricing that came from having multiple sources. One year we did a family of three asics with three different vendors plus a fourth chip that was the low end design second sourced from two of the vendors. On top of all this we breadboarded all three designs in a FPGA system.

I learned a lot about how to share and reuse IP and delivering a working net list on schedule. We were well ahead of the rest of the industry.

The recession ended my career after 28 years and 10 months. I was 57 and trying to figure out what I wanted to do now that I had grown up. When I started my career an asic mask set cost about $50,000. At the end of it they were quoting about $1,200,000. Asic design starts had been dropping year over year even before the recession. Fewer and fewer companies could afford to do asics. Staying with asics meant moving.

FPGA's one the other and were booming. Larger chips were opening up new areas where fpga only products were feasible and becoming more common. I also could see that FPGAs were following the same growth curve that asics took 20 years ago and that fpga designers were making the same tool mistakes that asic designers made. Some fpga designers still use schematic capture.

This became my mission. To create the tool set needed to generate System_on_Chip net lists for FPGA and asic designers. I know there are a lot of big guns in the EDA industry also trying to do this but I have some advantages. First of all I'm not in it for the money. The tools that I have tried out all have to squeeze in enough features to justify their price tags. They become bloated. Design for Reuse is not a complicated problem. The tool that we used at HP to preform most of the work was a simple macro preprocessor module that we downloaded from CPAN.

Second of all ,most of the commercial design for reuse tools didn't work. I would sit through the dog and pony shows telling me how these tools would do the job. Then we would start telling them about our design environment and their eyes would start glazing over. Their tools work fine if you have one design team in one location doing one chip with one vendor but anything beyond that is iffy.

Last of all they were addressing the wrong problem. They were trying to reuse IP that was never designed to be reused. That hard. SOCGEN tells you how to design IP so that it is reusable. That makes a big difference.

SOCGEN is a living project. I will be adding to it on a continuous basis from here on out. My legal constraints with HP have ended so I am sole owner of all the work that I create. I will not be adding any code here that was developed at HP. I now realize that we made a fundamental mistake in our tool set. We started by creating stand alone tools that ran under gmake and started working from the fringes of the design into the core. That was wrong. We should have started at the core and worked our way out to the fringes. The first SOCGEN tools will take that approach.

SOCGEN will also partner with and use other open source EDA tools when convenient. gEDA has a well developed PCB design flow and I intend to use gschem as part of my tool flow to feed parts to the PCB. Schematic capture is a useful tool. It's just that you do not want to be plopping down gates and connecting wires. You want to be plopping down components and connecting interfaces.

Till Later

John Eaton