The Power of Collaboration: Social-Media like Integrated Circuit Design

Cristinel Ababei
Milwaukee WI, USA, Feb. 2014

***Abstract*—We argue that given a social-media like framework in which design engineers from anywhere can participate collaboratively to create an integrated circuit, the final design would be better than any other realization of the same design done by only one organization alone. We envision this framework as the ideal set-up to unleash the power of collaboration fueled by the creativity of the single mind.**

I. THE POWER OF COLLABORATION – SOFTWARE

The power of collaboration is undeniable. It is especially so in the world of software. Since the birth of the Internet this concept has been proven numerous times. There are numerous examples where software products were developed in collaborative set-ups. Typically, a preliminary implementation is created by a single or small team of individuals. Then, once this implementation is made publicly available as open-source, the online community comes-in to take it to a whole new level of development with increased sophistication, diversified features, and enhanced performance. This is an iterative process, which in a way mimics a genetic algorithm in that newer generations of the software product get better not only by preserving good features from older versions but also by contributions of a larger and larger crowd participating in this process. It is interesting to note different levels and forms of collaboration achieving great things. For example, Apple's app store can be seen as a generic solution to address a huge variety of needs. Another example is Groupon's idea of teaming up people toward getting better deals on the Internet. Yet another example is Digg's reviewing or voting system. Examples of successful software products created in a collaborative process include Octave, R, Scilab, gEDA, Eclipse, etc. But perhaps one of the best examples is the Linux operating system itself.

II. COLLABORATION AND CREATIVITY OF THE SINGLE MIND – SOCIAL-MEDIA LIKE IC DESIGN

With the advent of the Internet, incipient forms of open-source hardware were represented by different designs posted online. Hardware intellectual property (IP) as Verilog or VHDL circuit descriptions have been available for many years in open-source form. An excellent example is the Opencores project [1], which represents an open-source hardware community developing open-source digital hardware including general purpose RISC CPU architectures. A similar example is the release in open-source form of the UltraSPARC microprocessors [2].

While these efforts are great open-source hardware ideas, their underlying principle has been primarily that of sharing. In this paper, we take the idea of open-source hardware to a new level by suggesting a paradigm shift. We argue that – similarly to software – hardware design too can benefit from the power of collaboration.

The main differences between the current open-source hardware philosophy and the proposed paradigm are:

- The current approach is to share a design in some form of register transfer level (RTL) language. Anyone can use the design and modify to suit particular needs. Instead, we propose a *collaborative development* of the design, which evolves elements of it being worked on concurrently by different designers. In addition, the design can have multiple incarnations such as RTL description, placed netlist, layout, etc. This process is done with a clearly defined objective such as performance and design constraints such as power and area.
- Current projects like Opencores and OpenSPARC offer primarily design repositories. Instead, we envision a collaborative framework, a design repository to *operate like a versioning system*. This system is similar to for example the well known concurrent versions system (CVS) [3]. While similar in philosophy, the proposed system is more than that, particularly with respect to the format of content, integrated simulation and validation tools, and its intended community of users.

A schematic representation of the proposed social-media like collaborative design framework is shown in Fig.1. While in the software world a project is stored in a CVS system for example simply as C++ code, in our system the design can be represented by different formats corresponding to different levels of abstraction. For example, while initially the design can be captured in HDL languages such as VHDL or Verilog, later it will be stored as a placed netlist, or finally as a layout in GDSII format. In addition, a new "snapshot" of the whole design can be composed of components that are at different stages of development. The simulation and verification of such a snapshot would be done by seamlessly integrating all these components.

While idea of collaborative hardware design is merely new, it has not been done at the scale proposed in this paper. The rationale behind the proposed framework roots from our attempt to answer the question: is innovation the result of the group or the individual mind? One popular opinion is that [4] *"Our species is the only creative species, and it has only one creative instrument, the individual mind and spirit of a man. Nothing was ever created by two men. There are no good collaborations, whether in art, in music, in poetry, in mathematics, in philosophy. Once the miracle of creation has taken place, the group can build and extend it, but the group never invents anything. The preciousness lies in the lonely mind of a man."* While the author is inclined towards the individual, however nurtured by the group, the proposed framework attempts to accommodate both:

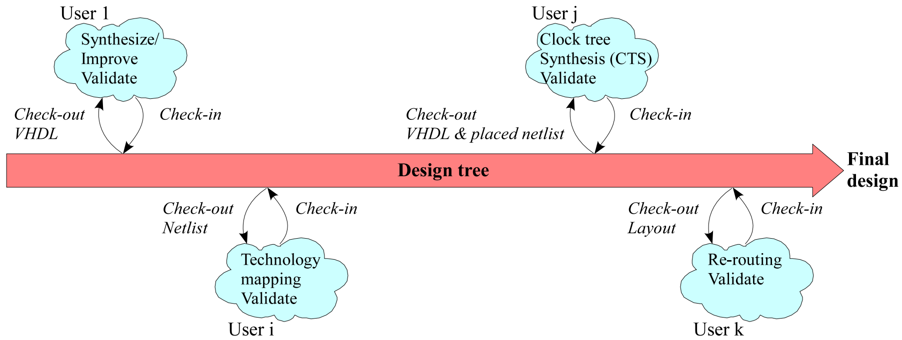

Figure 1. Illustration of social-media like IC design: engineers can check-out the design, make changes, and then check-in.

- Collaboration fosters improvement and innovation and enables tools diversification.
- The individual can contribute unique and novel ideas.

To operate properly, the proposed collaborative framework must follow at least the following rules or guidelines:

1. Initially, the functionality of a target design is specified using standard specification languages. In addition, the primary objective and constraints are identified. For example, clock period is to be minimized while total area should be less than some specified value and the temperature profile of the circuit floorplan should be within given bounds.
2. Any registered user can check-out parts or the entire design from any design stage. Checked-out design can be modified using any tool and design methodology available to the user. This is where designer experience and creativity is enabled to basically take a current snapshot of the design and improve it.
3. The user is allowed to check-in changes to the design only if the new snapshot improved the main objective function and/or on the satisfaction of constraints.

III. Discussion

In developing and deploying these ideas, we expect the following main challenges:

- The software platform to support a versioning system type operation is no trivial task. However, the computer science community is already working on some aspects related to this challenge [5]. Creating synthesis, simulation, and verification tools able to handle design components in different formats (or models of computation) from different levels of abstraction is also challenging. This is largely still an open problem nevertheless tackled by existing efforts [6], [7].
- It must be noted that while the proposed collaborative framework shares the same goals with OpenSPARC's initiative (nevertheless, via different implementations), we project that it will be ultimately a matter of economics [8] that will decide the practicality of these ideas. At the minimum, there must be some sort of a rewarding system to motivate designers to participate in such activities without loafing [9]. As a proof of concept, we plan to test these ideas within a graduate level course. Later, we will experiment within a larger ecosystem formed by students from courses at several different universities.

IV. Conclusion

We argued that hardware design in a framework that enables collaboration and unique contributions of the single mind can lead to better IC designs. This vision is a departure from the previous view of applying EDA knowledge to solving problems from social-networking [10]. Here, we want to emulate crowd sourcing via social-media type of frameworks to design better hardware for the greater good of humanity.

References

[1] http://opencores.org

[2] http://www.opensparc.net

[3] http://www.nongnu.org/cvs

[4] John Steinbeck, East of Eden, The Viking Press, 1952.

[5] Int. Conference on Collaboration Technologies and Systems, 2012, http://cts2012.cisedu.info

[6] Edward A. Lee, "Disciplined heterogeneous modeling," *ACM/IEEE Int. Conference on Model Driven Engineering, Languages, and Systems (MODELS)*, pp. 273-287, Oct. 2010.

[7] D.D. Gajski, S. Abdi, A. Gerstlauer, and G. Schirner, Embedded System Design: Modeling, Synthesis, Verification, Springer, July 2009.

[8] Gregory M. Pomerantz, Business Models for Open Source Hardware Design, 2000, https://files.nyu.edu/gmp216/public/papers/bmfosh-1.0.html

[9] S.J. Karau and K.P. Williams, "Social loafing: a meta-analytic review and theoretical integration," *Journal of Personality and Social Psychology*, vol. 65, no. 4, pp. 681-706, 1993.

[10] A. DeOrio and V. Bertacco, "Model checking social networks," *ACM/IEEE Design Automation Conference (DAC)*, 2010.