A Lean Fabless Semiconductor Business Model

Andreas Olofsson, Adapteva, Inc.

There is currently a general consensus among industry analysts, investors and company executives that leading-edge digital chip development projects cost $30 million to $100 million to develop. This negative view of chip development costs has led to severely decimated levels of VC investments and has even led to certain companies scrapping whole product lines[1]. Today's chips have a bigger impact on the world economy than ever and reduced investments in chip product development will eventually have a very negative impact on the economy. China, for one, has realized this and is pouring billions of dollars into their local semiconductor economy[2]. This article will go through some of the reported data and show that present assumptions regarding chip development costs are very much exaggerated, and that it is possible to design state of the art chips at a fraction of the cost previously reported.

Why is Chip Development so Expensive?

Chip development has never been cheap and there is no question that it's getting more expensive. The million dollar question is: "how much more expensive?" The consensus among industry pundits is that there is no way to do a leading-edge chip with less than $30 million to $100 million[3, 4]. Let's take a look at some of the most common arguments for the "exponential" increase in chip development costs.

"Mask Costs are Increasing Exponentially!"

It used to be that you could design a product with production mask costs in the range of $100 thousand. Today, production mask costs at leading nodes can cost $1 million to $2 million. This has the effect of increasing the minimum market size that is practical to address with a leading-edge chip design. Still, given a decent size market or high chip selling prices, mask costs are certainly not show stoppers, and don't explain the $100 million price tag.

"EDA Tools are Getting Really Expensive!"

While electronic design automation (EDA) tools are certainly not cheap, aggressive volume purchasing agreements, increased feature integration and increased efficiency in the tools have made today's tools far more cost effective than those of ten years ago. Today many leading chips can be taped out using vanilla flows derived from reference scripts provided by the EDA vendor. Thus, per transistor and per project, the cost of EDA tools has actually gone down in the last ten years.

"Deep Sub-Micron Design is Really Hard!"

Chip design is certainly getting more difficult as we move to finer process geometries. At 0.35um and above, life was simple. Gates were slow, wires were fast and there was virtually no leakage to speak of. The only problem was that chips were slow, power hungry and big! Since then, the design constraints and difficulties have been piling on with every process node. In order of introduction, designers had to learn and deal with: wire delays, voltage drops, signal integrity, leakage, process engineering effects like stress, and most recently on-chip variability and an explosion in the number and complexity of device design rules. The introduction of each one of these effects was painful, but within one process generation, the EDA industry always seemed to find automated methods of dealing with them, thus keeping engineering team growth to a minimum. Deep sub-micron chip design is definitely not for hobbyists, but for a small expert team equipped with state of the art tools, it is today possible to accomplish in weeks what used to take a generously sized team six to nine months[5].

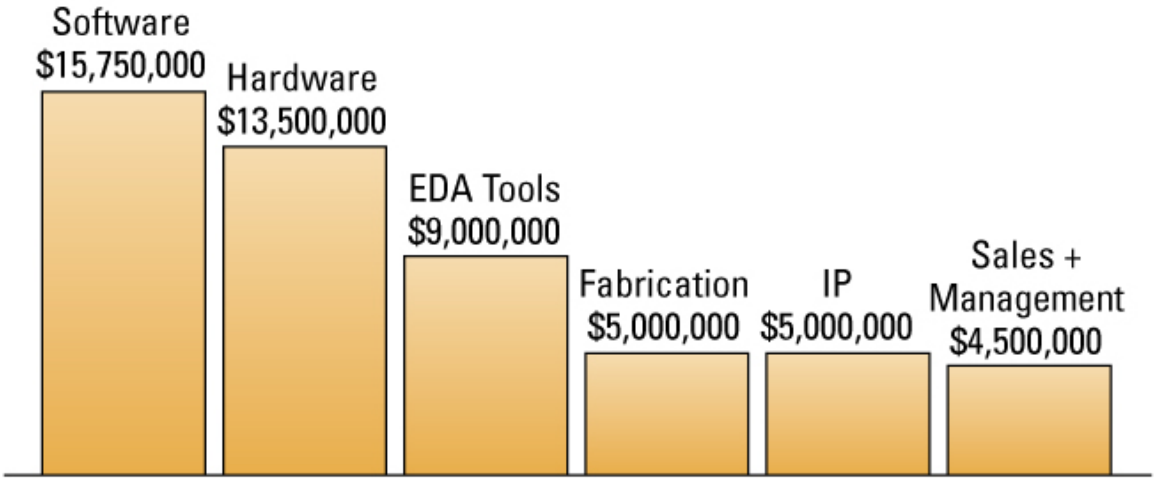
"Chips are Getting Incredibly Complex!"

Bingo! The simple reason for the explosive growth in chip design costs is that design complexities are growing exponentially thanks to Moore's Law. At leading-edge nodes, it is now possible to integrate complete multi-billion transistor computer systems on a single chip. The more complicated the system, the more complicated the chip. Complicated systems-on-chip (SOCs) are indeed very expensive to develop and there are few if any examples of complete products that have cost less than $10 million from start to finish. An application-specific IC (ASIC) on the other hand tends to solve a very specific problem and be a key part of a larger system. The limited generality of ASICs can lead to much higher performance levels and generally results in reduced complexity and development. For a great discussion on the relationship between project complexity, team sizes and project costs, I highly recommend reading "*The Mythical Man-Month*", by computer legend Fred Brooks. Unfortunately, there aren't a lot of short cuts for saving costs when projects are complex and need to be completed in a short time. Note that complexity drives up development costs of any project, regardless of the underlying implementation fabric. As many designers are discovering, large field programmable gate arrays (FPGAs) have all of the complexity problems of large SOC design, minus the long turnaround time and costs of manufacturing.

The Status-Quo Fabless Semi Startup Model

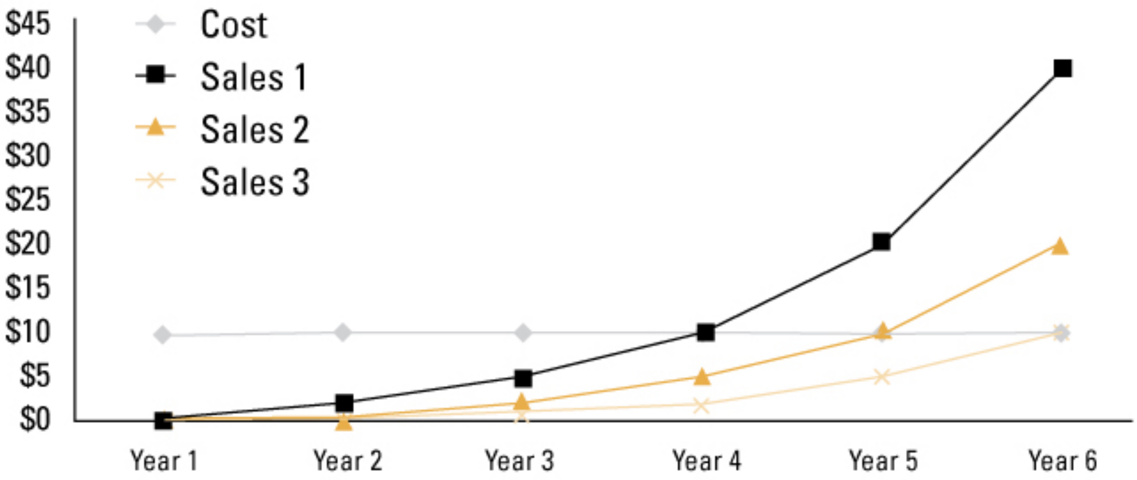
Figure 1 shows the typical costs of reaching production for a state-of-the-art SOC for a fabless semiconductor company. The data is based on personal experiences and numerous informal interviews with executives at chip startup companies who spent up to $50 million without reaching break-even.

Figure 1: Typical SOC Costs (Assumptions: Three years from start to break even for product, $150 thousand/engineer, 75 person company)



Clearly, despite the large EDA tool component and cmask building costs, by far the largest project cost is engineering. Figure 2 illustrates the biggest issue with the large burn rate business model shown above. It’s incredibly sensitive to delays in “time to money”, whether the delays are due to design slippage or the market materializing late. The figure shows three potential sales models with different degrees of eternal optimism built in. Anyone involved with semiconductor product sales knows that it’s incredibly hard to get to $10 million in profits for a chip product per year. The good news is that if you do, the potential upside is huge. The downside for startup companies without deep pockets is that if the company runs out of money before profitability is reached, it is again at the mercy of investors and/or macro-economic conditions. Many companies on the verge of greatness went out of business in 2008 because they ran out of money at the wrong time. Since time to sales carries the most amount of uncertainty, reducing the R&D expenses though the initial sales period can significantly increase the likelihood of success, given a fixed investment size.

Figure 2: High Burn Rate Semiconductor Business Model



Choose your Market

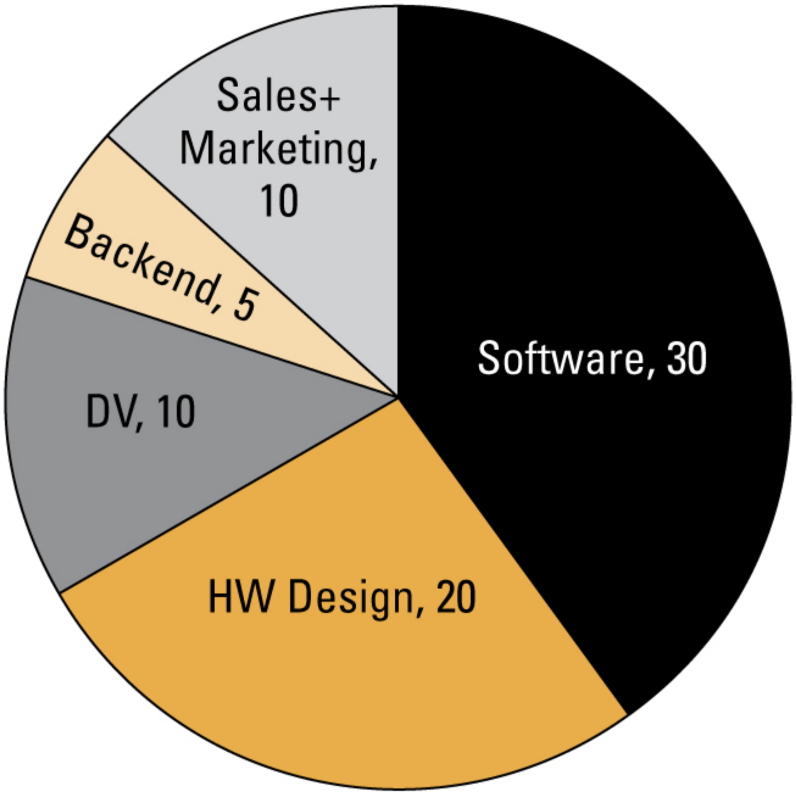
As we discussed, the biggest contributor to semiconductor development cost is design complexity. The easiest way to reduce design complexity is by picking the right market. The market will drive feature sets and chip selling prices. By far, the most extreme example is the SOC application processor in a cellular phone. Half a billion transistors, every input/output (I/O) interface one can think of, and yet the average selling price is often less than a large latte at your local café. This is not a good space for a startup, as the total investment in software and hardware could actually reach $1 billion at the end of the day!

Markets that are not quite as cost and space sensitive have a huge advantage in terms of development costs savings, because they allow companies to leverage existing chips in the market to make chip set solutions that satisfy the system requirements. For example, it is possible to buy off the shelf FPGAs and/or microcontrollers for $1 to $10 that will give a system a complete host computer and every connectivity standard in the book. High-end I/O Internet protocol (IP) such as PCI express (PCIe), DDR, universal serial bus (USB) and Ethernet are very expensive to design into SOCs and choosing all of them on a single chip adds a lot of cost and risk to a project. It's not just the purchasing costs, it's also the integration, validation and testing costs, in addition to the sticker price. The trend towards inexpensive mid-end FPGAs is especially encouraging, as it allows the fabless startup company to go after a large number of markets with a single chip platform. Software development costs also have a huge impact on product development. Some markets demand that chip vendors develop the whole software stack and give it away for pennies. Unless you are designing a system solution, and get a good return on investment, always focus on "teaching your customer to fish, rather than giving him the fish for free".

Further Costs Cutting Measures

Figure 3 shows the manpower breakdown for a typical modern fabless semiconductor startup company. On average, the SOC startup team seems to ramp up to approximately 75 engineers quickly, and stay at that team size until eventually ramping up towards an exit or going out of business.

Figure 3: SOC Design Team Composition



If we turn the fabless semiconductor business model on its head and ask the question: “What could be done about spending if we are willing to give up time and some down the road profits, instead of trying to minimize time to product at all costs?” Let’s go through each one of the expense categories, and look at some things that could be done to reduce costs.

- Software: Many chips today are sold with a complete accompanying software stack. The big problem is that the chip vendors rarely get properly compensated properly for the software by original equipment manufacturers (OEMs). The question posed is often: “Why should we pay for the software? We can’t use your chip without it and we really aren’t interested in learning the inner workings of your chip to be able to program it ourselves”. Two potential solutions to reducing software R&D include: 1.) Design your architecture so that you can leverage existing open source software packages (e.g. make it ANSI-C programmable in the case of processors) 2.) Make the chip attractive and easy enough to use that the customer can effectively do the programming himself.

- Hardware: Quoting the slogan of the green movement. “Reduce, Reuse and Recycle”. The way to reduce hardware development costs is by reducing the feature set that needs to be implemented, maximizing the reuse of validated blocks[6] and recycling all the great chip ideas of the last 70 years (no not-invented-here nonsense allowed). Reuse is one of the key competitive advantages in the chip industry. An SOC company that does not embrace reuse as a basic design imperative will not be profitable in the long run, because they will miss market windows and have R&D costs that are far too high.

- Design Verification: Reduced design complexity again comes to the rescue and the key here is to not be “too clever”. The prevailing method of chip development usually involves an experienced system architect writing a set of specifications that are then given to a team to design and verify. Without close interaction from day one between the architects and the complete design team, this almost certainly causes expensive problems downstream. In the words of Richie Kernigan. “Debugging is twice as hard as writing the code in the first place. Therefore, if you write the code as cleverly as possible, you are, by definition, not smart enough to debug it.” Designing architectures that can easily be verified with a small team is an art form that requires very tight integration between the architecture, design and verification teams.

- Back-End: There is a rich ecosystem of support companies that can take over as much, or as little, of the chip business the client requests. Opportunities for outsourcing include: manufacturing (a must!), RTL design, synthesis, verification, package design, board design, testing and logistics. If your core competency is chip design and computer architecture, then outsourcing software, board design, sales, marketing and logistics makes sense. If your core competency is system architecture, outsourcing a large part of the chip design makes sense, but make sure you find the right implementation team.

- Sales/Marketing: Selling chips with power point presentations has never worked and really just wastes money and burns out the sales and marketing team. The bigger the claim of the chip, the smaller the chance that the customer will do anything before he gets his hands on a board level development system. If you have a low R&D burn rate, you can be less aggressive with the pre-product sales and marketing effort and subsequently waste less money.

Conclusion

It is possible to spend $100 million (or even $1 billion) on chip development, but it’s also possible to spend as little as $2 million to reach profitability as a fabless semiconductor company.

About the Author

Andreas Olofsson founded Adapteva in February 2008. Prior to Adapteva, he was a key contributor to a number of successful products at Analog Devices, including the groundbreaking TigerSHARC digital signal processor (DSP) architecture. The TigerSHARC was the first processor to enable software programmable 3G and WiMax base station platforms and was the most energy efficient floating point microprocessor in the world at the time of release. While working at Analog Devices, Andreas also architected and productized a line of CCD controllers for digital still cameras that ended up in products with the fastest sales ramp in ADI history and gross revenue of over $100 million. Starting in 1996, Andreas has performed work in areas ranging from silicon process flow development all the way up to system level definition. Andreas' breadth of experience and holistic hands on approach to computer design has been a key ingredient in the development of a multicore architecture with unparalleled energy efficiency and ease of deployment. Andreas received his BS in physics and electrical engineering from the University of Pennsylvania in 1996 and his Masters in electrical engineering from the same school in 1997. Andreas has over 20 patents issued and/or pending.

References

[1] “Silicon Startups Get the Squeeze” http://www.eetimes.com/electronics-news/4083599/Silicon- startups-get-the-squeeze

[2] “The Rise of China's Fabless Industry “ http://www.eetimes.com/electronics-news/4216994/China- fabless-sector

[3] “Are SoC Development Costs Significantly Underestimated?” http://www.cadence.com/Community/blogs/ii/archive/2009/09/24/are-soc-development-costs- significantly-underestimated.aspx

[4] “Commentary: Stakeholders See Turning Point for Chip Industry“ http://www.eetimes.com/electronics-news/4210698/Commentary--Stakeholders-see-turning-point-for- chip-industry-semiconductor

[5] “From RTL to GDSII in Just Six Weeks”, http://www.eetimes.com/electronics-blogs/other/4211089/From-RTL-to-GDSII-in-Just-Six-Weeks-

[6] “Heard at DAC: Is IP Integration the Real High-level Design?” http://www.edn.com/blog/Practical_Chip_Design/39344-Heard_at"_DAC_is_IP_integration_the_real_high_level_design_.php