

DesignCon 2006

Innovation: Learning from the EDA Industry

Brian Bailey, Brian Bailey Consulting

email: brian_bailey@acm.org

web: <http://brianbailey.us>

Abstract

Innovation happens all of the time in just about every industry. Most of that innovation improves existing products, allowing them to do what they did before only faster, cheaper, better. These are what are referred to as sustaining innovations. Occasionally a second type of innovation occurs, called disruptive innovations, which can have significant effects on the industry and can change the market leaders. The EDA industry has seen a number of disruptions in its history and can learn a lot from looking at past changes. This paper will also identify a different type of disruption not identified in the business press as it relates to the equivalence checking market. Finally the paper looks ahead at the big disruption that is coming.

Author(s) Biography

Brian Bailey is an industry and management consultant specializing in the functional verification of electronic systems. He has implemented verification tools and solutions for companies of all sizes, and has been involved in some of the most significant advances in the field. He has recently been proclaimed as one of the visionaries of the industry and today spends most of his time helping start-up companies turn their ideas into practical realities. He is active in the standards community, chairing a committee within Accellera, had two books published in 2005 and is a popular contributor to the press and conferences. He graduated from Brunel University in England with a 1st class honours degree in Electrical and Electronic Engineering.

Introduction

"It is better for you to make your own products redundant than to allow someone else to do it for you."

Innovation has been studied by many people with many theories put forward about how it happens, how to control it and how to predict the next change. Perhaps the most famous of these business writers is Clayton Christensen of the Harvard Business School who has become required reading for most executives and popularized the term "disruptive innovation". At the same time I have seen managers and executives in EDA companies attempt to control the innovation process or to attempt to 'enable' innovation. It rarely works that way, as I believe at the end of the day that there is also an element of luck that comes into the success process, as well as having the right people, and the confluence of many influencing factors. However, it is very possible to stifle innovation and many companies are either directly or indirectly guilty of this.

In this paper I will not only draw on the work of Christensen but also add some of my own views of the innovation process and use one example in the EDA industry to highlight a form of innovation that I do not believe is covered by Christensen's analysis. The more we understand about the past, the clearer picture we can get into some of the possible outcomes in the future.

Sustaining Innovation

Innovation happens all the time in just about every industry. Most of that innovation improves existing products, allowing them to do what they did before only faster, cheaper and/or better. These are what are referred to as sustaining innovations and history shows us that the established players in a market love these types of innovation. They have the brand names, the distribution channels, the budgets to get these innovations wrapped into their products and to get them into the hands of their customers. New entrants in the market may have something better, but they struggle to have anyone hear their voice, or see their product. Most of the time, people are somewhat reluctant to buy from a company they have never heard of. In some markets this is more so than others.

If we were talking about a bar of soap, most people just stick to the brand they are used to, and do not even look at the other products that line the shelf. Those who do may look at the new entrant, even though it is likely to be put in a place where most people would not see it as they cannot command the premium shelf space, are still unlikely to buy. How do they know it is better? Is it cheaper or have a distinct stated advantage? What would make them give up the security and comfort of their favorite brand? If we assume that they believe company marketing and that they are tempted to give the product a try, they may buy a single bar. What is the downside? They use it, find that they don't like it, throw it away and they are out of pocket a small amount of money. They are willing to risk that amount of money because there is the possibility that they really like the new product. But what if the product is not on the shelf the next time they go into the store? Well, they go back to the old brand and probably don't think to search out the new product elsewhere. It is probably not worth the extra effort to go to another store that may

still stock it. Incidentally this happens to most new products sold in a grocery store. A typical grocery store carries 30,000 items and each year places 20,000 new products on their shelves. Within a year, only 20% of those new products are still available and even fewer 3 to 5 years down the road.

Innovation is also alive and well in the high-tech world. In fact companies that do not innovate become irrelevant, as other companies will quickly take market share from them. There are three primary groups of sustaining innovations: technological, market and process. Technological innovations make improvements to the products that already exist. This may mean additional features, better battery life, smaller physical size or many other types of improvements. Market innovations find new ways to appeal to a set of customers, or to make a product cool. While they often rely on the right technology as well, color, shapes, associations with popular figures can have enormous impacts on the desirability of a product. And finally process innovations. These innovations make it faster or cheaper to produce a product. It may mean streamlined manufacturing, a reduction in parts count, or using external resources to do a job better than would be possible within the company. Of course really successful companies combine all three kinds of innovation together to make remarkable products that we all want, and cheaply enough that we can afford to buy them.

Many of the innovations surrounding the Internet were not as radical as some people thought. For example, in the retail industry, pundits claimed that brick and mortar stores were going to be completely replaced by on-line retailers. The problem with this argument is that many retailers already had catalog businesses and the migration from catalog to Internet was a fairly small sustaining innovation step for them. So the 'old guard' was quite able to show the 'new economy' companies exactly how to do it, and often already had everything in place such as distributions centers, layout staff etc.

At the same time, innovation becomes difficult for many companies, especially large companies with secure markets. They see little need or advantage to innovate and only start to look at their own internal processes for innovation when they feel threatened by a competitor or a new emerging technology. Many of these problems come from the management structures that are put in place to ensure an efficient operation for their existing product lines. Often the processes do not encourage innovation, or when they do, restrict the types of innovation to those that directly affect the current line-up of products. More progressive companies actually encourage innovation by giving employees a certain percentage of their time to spend on any project that they like or establishing on-line resources to help encourage employees to share ideas.

Sustaining Innovation in the EDA industry

Buying an EDA tool is very different from a bar of soap, although it seems that the EDA industry uses sales and marketing types or even CEO's who might as well be selling soap, or machine tools, or indeed hotel rooms. When talking about products that cost a lot more and can have a significant impact on a company's existence or health, many factors have to be considered. In the EDA industry, startup companies face enormous pressures bringing a new product to market. For example, a large semiconductor company, will not be comfortable buying a tool from a startup when they cannot be sure that they will be in existence next week, or when they need support from them, or to fix a critical bug that is

stopping them from taping out their latest design. That is the bad news. The good news is that the creation of an EDA company requires very low startup capital, the cost of a compute, a compiler and a certain amount of human time. Often within a year a new idea or tool can be developed by a few smart people, who probably can afford a year or two of no salary. In most cases they sell their product by comparing it against those available in the market, a benchmark - often doing the comparison themselves. Many new products start with 'an AE in the box', meaning that the new company will commit the time and people necessary to make their customer successful. While this strategy works when they only have one or two customers, it is not a scalable business model. The market for EDA is not localized to one city, state, country or even continent. Design is a worldwide business. EDA must also be worldwide. The first customer may be in the US, second in Japan, third in Europe. So to scale the marketing sales and field support of EDA products requires building a worldwide organization fairly quickly. The distributor model does not work well for most EDA products – few can be ‘shrink-wrapped’ and supported by distributors. Many have very leading edge and complex functions that need considerable technical smarts to appreciate and try out. Therefore scaling any new EDA product can be quite expensive due to this world scope.

The normal life expectancy of an EDA startup is short. The goal for most startups is to produce a product, demonstrate the viability of the idea by selling the product to a few customers, and then looking for an exit strategy, often one of the major EDA companies acquiring them. While no startup should think of this as their goal without also considering how they would make the product good enough that they could exist on their own, few can afford the expansion costs to take those concepts to the next level. A large company will likely acquire them and can integrate the new and exciting parts of that technology into their existing tool suite, and everyone is happy. The startup founders get some reward for the time and effort they put into it, the large company gets access to the new ideas and hopefully sells more of their products because of it, the customer gains a stable company to stand behind and support the product so they feel more comfortable using it in a production process or products critical to their company. This rapid creation and verification of new ideas and the assimilation of the successful ones into the large EDA supplier's product offerings is part of the reason why there is little innovation within the large EDA companies themselves. Given the risks associated with the development of new products, it can be faster, cheaper and less risky to allow the new companies to do the innovation for them. Larger companies have the worldwide marketing, sales and field support teams that are very expensive for a startup to build and thus offer an economy of scale that can be hard to match for a small startup. So selling out as an exit strategy may actually be the only viable one for a small startup.

Disruptive Innovation

Every so often, there is an innovation that does not fit into this mold. These innovations do not improve on the existing tools, instead they create complete new categories of tools, with new value chains and can often stimulate the most interest from a new class of customer. The tool companies with the old products are not interested in them because they fail to see the value, they cannot integrate the offerings into their existing products and they do not have the contacts with the people necessary to enhance sales. The theory

goes something like this: Companies with a product in a particular market listen to their customers in general. They learn from them what is important and what new features or capabilities they would like to see in the product. From this feedback and the constant pressure to improve, the product is enhanced on an on-going basis, with every release providing additional capability. However, the customers for this product are not able to use all of the new capabilities that are being added. In fact it may start to diminish the overall value of the product because it becomes too complicated, too slow because of all of the new features, or too expensive for the same reason. This is called product overshoot.

As an example, think about the software product Microsoft Word, in which this paper has been authored. It has existed for many years and over that time more and more features have been added. How many of those features do you use? How much performance or additional memory usage are you paying for those unused features? For most people, there are probably quite a lot which represents a large amount of wastage, but the reasoning goes that by adding additional features, it makes it more difficult for a competitor to come in and take your market away because they cannot match your feature set, and a slimmed down product may be missing your favorite feature. In this particular case there are many other reasons why we continue to use Word even though most people who use it also curse at it on a regular basis. Other examples of product overshoot include desktop processor performance, DVD players and more recently in WiFi routers.

At the same time, imagine that a new product becomes available, which may cost more, have less capability initially and potentially have fewer features. It appears to be inferior to the existing product. Historical examples of this include the introduction of voice telephony, when at the time, long distance data communications was the dominant technology "Morse Code". Who would want to use a telephone to talk to people, was the question of the day in the 1870's. It could only be used over very short distances and was thus nothing but a toy, was the thought of then dominant supplier Western Union, who passed up the chance to acquire the patent for the technology. Other examples have happened in the disk drive industry with each new platter size becoming one of the enablers for a different type of computer, from mainframe, to mini computer, to personal computer, to lap top and most recently to iPod. Today ironically we can see the reverse of the 1870's problem happening. Today it is the dominant voice telephony companies that are being squeezed by the long distance data communications market with VoIP threatening to undermine their sales. Would Western Union have been right all along to have continued with digital communications over a long distance infrastructure? Perhaps they were just let down by the quality of the technology at the time.

These innovations are examples of what are termed disruptive innovations. While not the original creator of the theory or term, Clayton Christensen, of the Harvard Business School, was the person who brought this to the attention of the high tech industry in his award winning book "The Innovator's Dilemma". This has become such a classic book in the industry that it is difficult today to go into a company who is looking at their innovation process and to not talk about the theory presented in this book and its later derivatives "The Innovator's Solution: Creating and Sustaining Successful Growth " and "Seeing What's Next: Using Theories of Innovation to Predict Industry Change". Companies such as Hewlett Packard, DuPont, Eli Lilly, Lockheed Martin and Quantum are just a few of the companies using this to guide their innovation process.

Other notable business writers, such as Geoffrey Moore, have also incorporated Christensen's thinking into their models typified by his books, "Crossing the Chasm", "Inside the Tornado : Marketing Strategies from Silicon Valley's Cutting Edge" and "Living on the Fault Line: Managing for Shareholder Value in Any Economy". This most recent book from Moore has extended Christensen's thinking outside of the high technology arena and made it part of the more general company innovation process.

Disruptive Innovation in the EDA industry

There have clearly been some big disruptions in the EDA industry in the past. The emergence of the EDA industry was in itself a disruption from the large internal CAD departments that the electronics companies of the day used to have. Even since then, one only needs to think about who the dominant suppliers have been over time. In the earliest days of the industry were Calma, Applicon and other companies that enabled polygons to be placed and edited. These were replaced by Mentor, Daisy and Valid which supplied systems that could help users design at the gate level using schematics. At the same time, the emergence of standard cell design and automated place and route was emerging and places like Bell Labs and other large systems companies continued to do their own tools often on mainframes before moving to Mentor, Daisy, Valid. The gate level tools could not create chips that were as efficient in terms of area or even overall quality of results, but they were good enough for most people and allowed them to be a lot more productive by enabling them to work at a higher level of abstraction. It also allowed the market to grow, because the model provided by these EDA tools was somewhat simpler and allowed a great number of engineers to become proficient in using them. Think today of the differences between analog designers compared to digital designers. There must be 100 or more digital designers for every analog designer, and most digital designers seem to think that analog design is close to black magic.

These companies in turn were replaced by Cadence who concentrated on the place and route problem and Synopsys with the birth of RTL - another increase in abstraction that again produced less optimal solutions than was possible by the tedious gate level work at lower levels of abstraction, but with higher productivity. Again, the market grew because of the simpler programming model associated with RTL. Indeed, one of the disruptive aspects was that it was a programming model, akin to writing SW, rather than a HW gate or cell design model, which is really logic design. While Mentor managed to survive this shift, it did not come out of it the powerhouse that it once was and was relegated to the role of point tool supplier in a restricted number of areas. Mentor tried with synthesis – Autologic which at one point had 10-15% of the market, but eventually was terminated when sales could not be ramped sufficiently. Cadence completely missed the synthesis boat with their internally developed tool never getting beyond a 3-5% market share. This demonstrates the difficulty in displacing a clear leader in a field even for the large established companies who have an effective sales and marketing force in place.

Each of these companies have been looking for and expecting the next disruptive change. Grant Martin, Chief Scientist at Tensilica has stated that they act like turtles, or ostriches. They occasionally put their heads out into the open air, look around, try to see what is different, and then retreat back into their shells or bury their heads back into the sand. This is the retrenchment philosophy that they periodically go through when their

mainstream business goes sour for a quarter or two, and they need to find a rational explanation for it. The turtle or ostrich just might never see that 10 ton truck heading straight for them!!! In a DesignCon East keynote Raul Camposano, the CTO of Synopsys also used a similar analogy when talking about deep-sub-micron effects, but quite humorously pointed out, that no-one has actually observed an ostrich with their head in the sand. The key point here is that they have all been trying to predict when it will happen with the hope of being able to survive the shift. Those who do and get it right stand to be the big companies of the next round of EDA evolution. So far, while they have all stated that they have seen the future and that they were ready for it, the migration has failed to happen.

Cadence tried with its VCC effort, based on years of research with Polis, and over a hundred man-years of effort. Synopsys released its behavioral compiler and withdrew it after a few years of failing to gain any traction with it. Mentor has tried on a number of occasions to release high level synthesis products which were withdrawn and re-engineered. It is currently trying again and the product, Catapult-C is receiving some encouraging reviews, but is it just a better synthesis product, or is it the beginning of a move to a higher level of abstraction? Only time will answer that question and with hindsight may appear to have been obvious all along. But as with most disruptive changes they tend to happen at unexpected times and not in the ways that anyone thought they would happen. Are they all trying for the obvious extensions to the technology they already have, and thus trying to create a disruption out of sustaining innovations? Quite possibly.

This should not imply that disruption in the classic sense does not go on within the EDA industry, and that tools cannot be successfully implemented by the major player. One such example is with the adoption of processor based design. Processors were typically not fast enough for many jobs but their performance and power profiles have now become sufficient to allow what was once custom logic to be replaced by software running on the processor. This makes it possible for designs to become more modular with better defined interfaces, and to be able to utilize libraries of components ready made to bolt together in ways that produce much higher productivity. To strengthen this point further Clayton Christensen provided the forward to Chris Rowen's new book entitled "Engineering the Complex SoC". This book talks about the emergence of processors in the SoC and in particular reconfigurable processors.

A different type of Disruption

There is a second kind of disruption that happens in the EDA industry, which may be somewhat unique to this industry. Moore's law predicts the rate of growth of chip sizes and thus complexity. This rate of a doubling every 12 to 18 months puts enormous pressures on the EDA industry to improve tools to keep up with this demand. Sometimes the algorithms that are in use just fail to keep up, even though the companies are continually looking for ways to make them more efficient. New algorithms or improvements cannot be found and it is only by having a paradigm shift that a better solution can be found. So unlike the classic disruptions, these are not caused by over capable tools, they are caused by the inability of tools to satisfy the needs any more due to the increase in the demands placed on them. The process is similar to simulated

annealing. In the annealing process a metal or other substance is heated and then can be cooled in various ways. Depending on the rate of cooling a specific crystalline structure starts to form. Once the form of the structure is defined, then the cooling process continues and the substance condenses into the most compact form for that structure. But if a different rate of cooling is used initially, a completely different structure may be adopted by the substance which can have very different properties. Just think about the differences between diamond and graphite, or the tensile strengths of different kinds of steel. The EDA industry is good at refining and optimizing an approach once it has been seeded by someone else, but then gets stuck in these local minima and fails to see other solutions.

While there are differences between this type of disruption compared to a classic disruption, they also have some similarities. Very much like the classic disruptors, the new class of tools does often start with much lower levels of capability and only over time do they overtake the old technologies in terms of performance or capabilities. Consider as an example, the case of combinatorial equivalence checking.

A case example

In the 1980s almost all simulation performed was at the gate level. The number of gate types is fairly limited (and, nand, or, nor, xor, nxor) plus a few special types for handling pass transistors and tri-state drivers. Also at that time, delays were fairly well defined and well behaved compared to today. In most cases a simple rise and fall time was associated with the output of the device. While designs were a few thousand gates or even a few 10's of thousands, the gate level simulators of the day had reasonable performance and with the increases in tool performance through better simulation algorithms and faster computers, managed to keep up with the demands placed on them. The tool vendors of the day were obviously always being pushed for more performance, and this they gladly provided.

While designs were getting bigger, the geometries of the chips became smaller, and the simple gate delay models started to become insufficient. Wire capacitance started to become important, so the delay models were improved to take this into account. Delays were associated with gate inputs as well as outputs, but when this proved insufficient, point to point timing was introduced so that complete path lengths could be taken into account. This added amount of detail had two major impacts. First and not surprisingly, the simulators had a lot more work to do, and thus became slower. Secondly, the wide variations that could now exist with the timing within a chip made it more and more difficult to develop the test vectors that would exercise the most challenging timing paths within the design. In other words a lot more test cases were needed and each of these would take longer to run. One of the problems with this approach is that the test vectors were both redundant, in that many of them would keep exercising the same paths in the design many times over, and at the same time they were incomplete, such that they would miss the problems that existed.

Rather than solving the problem, the difficulty continued to increase with ever more detailed timing models being used and design sizes reaching into the millions of gates. Even though by this time, the last major disruption in the industry had already happened

and designers were now using RTL entry and synthesis to create even bigger designs, most fabrication companies still required gate level simulations for sign-off, so companies were forced to continue down this path. Tests that were developed to run on the RTL models and could take hours to run, would take days at the gate level. Standards groups spent many hours working on ways to improve the accuracy and portability of the timing information. The situation had reached a crisis point, and there was just no more performance that could be wringed out of the simulation algorithms. Emulation, parallel processing and other techniques were attempted to alleviate the problem, but none of them provided more than a band-aid for the problem.

An emerging disruptive innovation

In the early 90's an alternative to gate level simulation started to emerge, called formal equivalence checking. The essence of this technology attempted to answer the following question: is this design functionally equivalent to an earlier reference design. These tools worked at the gate level and would compare two gate level netlists, a somewhat small niche, but it was useful for ensuring that after test insertion or other post synthesis changes had been made, that unintended changes had not affected the system's functionality. It first became accepted as a means of managing ECO's that were conducted on the netlist as it made it possible to avoid a re-run of the complete regression suite after every change. These regression suites would often run into the days or weeks, and the elimination of these could be seen as a positive return on investment for the tool.

Part of the value of equivalence checking tools is that they do not require a test bench. Instead all logical behaviors are considered. This led to a perceived problem of these early tools which said that they could never handle the largest designs because of the huge amounts of memory necessary to perform the analysis. Instead they would be relegated to doing small pieces of the circuit at a time and thus would never replace simulation. In the early days of equivalence checking this was somewhat true, but many of these concerns were based on misunderstanding of the technology and the early stage in development of the tools themselves. In addition, equivalence checking must be regarded as a 'garbage in - garbage out' tool. If the earlier design has a bug in it, equivalence checking will prove that the new design also has the same bug in it. It cannot tell you anything about the correct functionality of the circuit. It was thus an add-on tool and not a replacement tool because it still required that the gate level simulation had been run first and verified to be correct. Additional capital expenditures are never easy for a company to accept, especially for unproven technology. This limited the uptake of the tools.

Another factor is that equivalence checking is only half of the problem. It proved that two designs were functionally equivalent, but completely ignored the timing aspects of the design. Timing was one of the main parts of the problem that was giving the simulators so much trouble. The acceptance of equivalence checking thus depended on the success of a second tool, static timing analysis, for the complete value to become apparent. It was the maturing of both tools that enabled the disruption to take place.

A second generation of the tools was also necessary in order to make equivalence checking a replacement technology rather than an adjunct technology. That was the success of RTL to gate equivalence checking. When this technology first appeared it was

also questioned, but for different reasons than the earlier gate to gate tools. I can personally remember one objection: "Why is a tool like this necessary, when all it does is check to see if the synthesis results are correct?" When Synopsys came out with a tool to do this, it was common to hear comments such as "is this the fox guarding the hen house"? While there were some real concerns in these comments, they completely missed the value of such tools. Not only did they ensure that the two designs at different levels of abstraction functionally matched, but they also enabled modified gate designs to be compared back to the golden RTL sources rather than a derivative gate description, which was more difficult to understand and debug when issues were found. It was only with the commercial acceptance of this product that companies became comfortable with relegating gate level simulation to a comfort check rather than a necessary step.

Today, only a few percent of companies perform any gate level simulation. Typical figures for equivalence checking claim to perform complete verification of a million gate design in 2 to 3 hours using 300-600MBytes of memory per million gates. Another case reported doing an RTL to gate equivalence check for 50,000 lines of RTL, that synthesized into 2 million gates which took 23 minutes of CPU times and consumed 420MBytes of memory. This is perhaps close to the time it would take to perform one gate level simulation run which would cover just a few percent of the design.

Industry Impact

It should come as no surprise than none of the major EDA players were a significant factor in the emergence of equivalence checking. Universities and internal CAD departments of the major electronics companies were the first to experiment with the techniques and small startup companies were the first to attempt to commercialize the technology.

The effect and rate of adoption within the major EDA companies was also inversely proportional to the success that they were having in gate level simulation. The better solution they had, the more customers and money they had coming from those tools. They could not see the value in the new technology and instead attempted to prolong the life of the old technology as long as they could. Mentor Graphics was a clear number one in the gate level simulation area with their Quicksim II simulator. The weakest of the major players was Synopsys who had entered the simulation market late and had backed the VHDL - RTL verification market. Not only was VHDL weak in the gate level area, but it was also slow. Synopsys quickly began to lose ground in this area, and so they were the first of the majors to embrace equivalence checking and later supplemented that lead with technology acquired from Avanti, which itself was acquired technology from Chrysalis, the original leader in the startup phase. While the early Synopsys solutions were somewhat weak, the head start enabled Synopsys to get established in this market and with the acquisition of supplementary technology quickly became the technology leader, and by 2001 had a 47% market share.

Cadence followed later with an internal development and the acquisition of technology from a number of sources, including acquiring Verplex, a company that had become the strongest player in the startup field with a 45% market share. Mentor was the last to fully accept the new market, and attempted to write a tool from scratch, without bringing in any external technology. This action lost them even more time in entering the market and

meant that it was not really possible for their tool to differentiate itself from the rest. As a result it had no displacement power, managed to only get up to about 5% market share, and in 2004, Mentor effectively exited the market.

While it could be argued that the introduction of equivalence checking diminished the total market size for simulators, this would be a very short sighted view. It is possible that there was a slight dip for a while, but almost all of the gate level simulator licenses were translated into RTL simulator licenses. Freed from the necessity of performing additional back end verification, the engineers could spend more time simulating at the RTL level. The extra productivity this gave them allowed more complex designs to be created which increased the need for RTL simulation, and the demands placed on these simulators has been steadily increasing.

A look at the revenue numbers for these tools also tells the story.

	2000	2001	2002	2003
Equivalence Checking	33.1	49.9	50.3	49.8
Static Timing Verification	61.3	68.3	64.7	67.9
Gate Level Simulation	16.5	13.2	9.8	6.4

All figures are in \$M

This shows that Equivalence checking completed the transition in the 2000 to 2001 period and quickly settled at this level. Static timing analysis which had also been successfully adopted independently before this, also took a bump up in sales during that transition period. At the same time gate level simulation went on a steep and continuous decline. However, there is a net addition to the value of the tools with the increase in sales of the new tools more than compensating for the decline in gate simulation sales.

Future Disruptions

This disruption provides an interesting possibility for considering future disruptions: it is often by the separation of issues that significant success can be achieved. We are today seeing another crisis building in verification, this time at the RTL level. Many companies now report that verification is taking 70% or more of the total design time. A large portion of this time is spent performing system level verification functions using RTL models. While a number of innovations have improved the efficiency in the generation of the vectors necessary for verification, they still have to be fed through the RTL simulator. Companies have been building large farms of simulators to help speed up this process, but this approach does not scale, and the utilization of automatic vector generators is actually making it possible to run even more test cases, and thus compounding the problem.

This leads to two possible outcomes: firstly, the possibility that equivalence checking may move up to the next level reducing the need for as many RTL simulation runs, or secondly, that the abstraction used for system level verification will move up to a higher level of abstraction making it possible for those simulations to run much faster. In a

recent experiment performed by Cadence and reported at DVCon 2005, they showed a 450 X speed up in simulation by using a higher level of abstraction for their models. They went on to note, that known inefficiencies in their approach could account for another 2X to 5X performance increase beyond the result reported.

Given these levels of possible increases in verification performance and the corresponding increases in design efficiency that would result from this, it is not surprising to see many startup companies working on solving these verification problems. With design only accounting for 30% of the total time, companies who expect the next disruption to happen in the synthesis space, which includes all of the major EDA companies, may be looking in the wrong direction. If the latest round of startups succeed in producing the tools to solve these problems, one or more of them may become the next new big EDA company, or at the very least be the next round of companies that the existing major companies have to acquire in order for them to stay in the running.

It is also interesting to note that if the analogy of equivalence checking is carried forward, that we should not expect either the system simulation adoption to be completely successful or to see widespread adoption of sequential equivalence checking. This is because there is a third technology development that is missing - the ability to ensure that the design conforms to the timing requirements of the specification. The equivalence checker will only provide the assurance that functional aspects have been maintained in the design process. This then leads to a new possible place for a startup company to enter with a product that is the notion of timing analysis moved up to the system level.

Recent introduction of assertion based verification and the associated languages of PSL and SystemVerilog have provided a language base on which this kind of product could be built, but so far no company has produced a tool that allows timing constraints based on these languages to be formally applied to the results of system level synthesis. So far, assertions have been applied to a mixture of both functionality and timing and perhaps this is a mistake. As stated previously, it is often by a separation of issues that problems can be effectively solved.

Conclusions

Looking at the past and extrapolating into the future can be very dangerous as it usually results in companies missing the significant changes that are happening in the field. Disruptive changes are better handled by smaller, more nimble companies. This is true whatever industry you in. But in the EDA industry, the operating model of the major companies as sales and distribution channels has almost built this into the way in which the industry operates. It thus becomes necessary that the major EDA companies keep a close watch on the startups and the areas in which they are working. The trick then is to decide which will be successful, and how long to wait for the proof that the application of their new technology is gaining momentum before acquiring them. This is likely to provide a better cost to risk ratio than doing the development internally.

While extrapolation from the past can be problematic, a careful analysis of it can also provide clues to where the next block of innovations may come from. However, care

must be taken not to latch onto only one model or line into the future as is being done with synthesis. It is too easy to be swayed by the last major success story and to expect the next one to play out in the same way. History shows us that this strategy is flawed

References

Brian Bailey. "Verification moves to a higher level". EETimes October 3rd 2005

Sebastian Jäger. Diploma Thesis "Technical and Economical Barriers and Drivers for the Introduction of Formal Methods in the Verification of Digital Systems".

Clayton Christensen. "The Innovators Dilemma". Harvard Business Press 1997

Clayton Christensen. "The Innovators Solution". Harvard Business Press 2003

Clayton Christensen. "Seeing What's Next". Harvard Business Press 2004

Geoffrey A. Moore. "Crossing the Chasm". Harper Business 1991

Geoffrey A. Moore. "Living on the Fault Line". Harper Business 2000