Nine Cost Considerations to Keep IP Relevant

WHITE PAPER

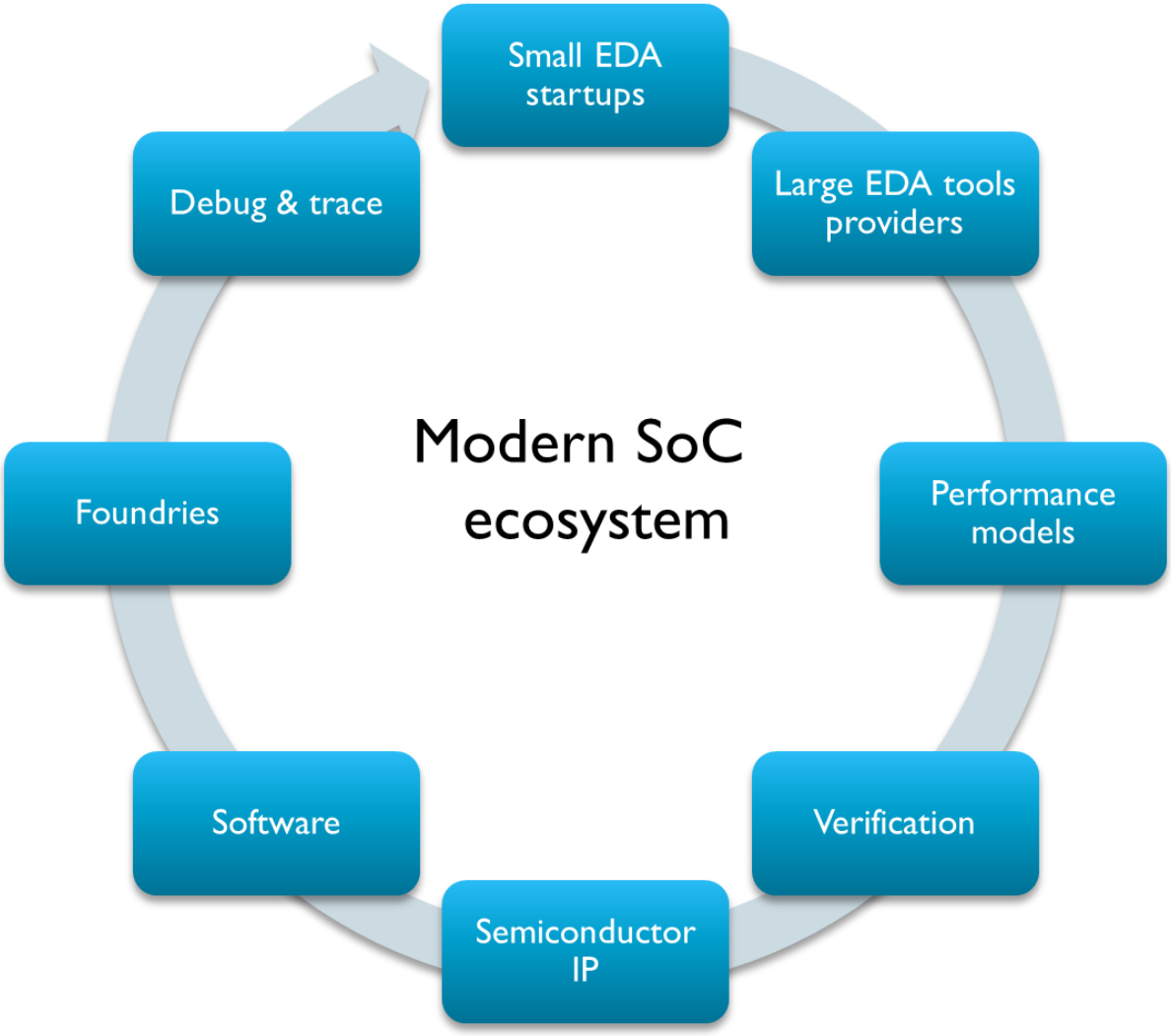
(October, 2015)

Pawan Kumar Fangaria
Founder & President

Fangaria Associates
1202, ATS Greens – II
A 58, Sector – 50
NOIDA – 201 307
www.fangarias.com

Introduction

It's about 15 years the concept of IP development and its usage took place. In the recent past the semiconductor industry witnessed start of a large number of IP companies across the globe. However, according to Gary Smith's presentation before the start of 52nd **DAC**, IP business is expected to remain stagnant for next 5 years. There are reasons to believe into Gary's thesis. A bird's eye view shows an IP sitting at the heart of an SoC or subsystem. This is significant reason for a system company to assess an IP fully before utilizing it; also assess the IP provider's quality and other business practices. At the tip of the iceberg it appears very simple to buy an IP and use it in your SoC design as required. However there are significant implications of using an IP from business as well as technical perspective; not all system companies have bought into the idea of using 3rd party IP, barring some standard and common IP blocks from reputed suppliers. The standardization of IP blocks that go into most of the SoCs reduces cost for the overall value-chain of developing SoCs; however it can commoditize the stuff to an extent that it can start impacting the differentiated value of SoCs. Moreover, there are serious technical implications that need to be considered before using IP. There has been a significant change in the modern SoC ecosystem where the system companies are experiencing increasing need of customizing IP before their use in SoCs.



[Courtesy ARM Community]

Considering it from a macroeconomic angle in a consolidating semiconductor industry, the IP-based business model of SoC design does provide a good proposition provided differentiated value is added into the SoC. However, it's essential that the hidden costs in accomplishing some specific tasks to make this model successful are better understood. Often certain tasks are not performed adequately because of lack of understanding, and also because the associate costs not considered. This can leave an IP in a poor state, inside or outside of an SoC. The success or failure of an IP in a system depends upon how best these tasks are understood, invested-in, and performed by the IP provider as well as the system integrator. There are specific costs involved in doing these tasks which stand apart from the usual developmental cost involved in the normal course of IP development. These exclusive types of costs other than normal development are mentioned below along with what incurs those costs and their proper rationale.

Cost of Differentiation

The differentiation in an IP has to be construed from the design level. The system companies are expecting differentiation in IP that fits into their designs so that they don't have to design the same IP themselves as much as possible. A common form of differentiation can emanate from IP vendors for providing extended solutions such as interconnect along with the cores. It's true that such differentiation can again be seen as common in an IP for different SoC vendors; however it can move the IP to a level up. IP providers such as **ARM**, **Synopsys**, **Cadence** and some others are providing subsystem level IP solutions. On the other side of the coin an IP provider can work in joint collaboration with an SoC vendor to design a completely differentiated IP. In this case the cost can be very high; in-sourcing of the complete IP team may be preferable for the SoC vendor. In other words, the IP team in-house with the SoC vendor can work at the sub-system level which allows the team to add enough differentiation, do trial layouts and optimize, and thus reduce risk and time-to-market.

Power is becoming a prime criterion for differentiation, especially in the mobile and IoT market. An IP characterized for certain power parameters with a particular technology needs to be re-designed in most cases of newer technology; moreover the dynamic power profile may change significantly with the use-cases.

Considering PPA (Power, Performance, and Area), an IP can be designed to have flexibility to scale between different factors such as power and performance according to the technology used.

A new concept of IP abstraction is coming up where an IP is delivered at a higher level of abstraction which goes through High Level Synthesis at the SoC end. This provides scope of differentiating the IP and SoC in power consumption. **Qualcomm** and **Google** have used this approach with **Calypto** (now **Mentor**) HLS solution where they deliver 'C' code which can be further optimized while integrating into SoC at system-level. A start-up **Adapt IP** provides option for delivery of IP at a high level of abstraction. In this scenario, the cost for IP vendor can decrease, but that gets added up in the SoC for the SoC vendor to account for differentiation and implementation. Moreover, this brings in a newer methodology for SoC architecture exploration and integration at the system-level and asks for fresh investment and learning. I will talk more about it in later sections.

Cost of Customization

One may think customization as a part of differentiation, but actually they are distinct. Customization is a process which takes place during integration of an IP in an SoC. There is a separate section on integration in this article. In this section I am talking about the provisions which need to be made in the IP itself to make it customizable according to different environments; interconnect IP is a good example in this case. For example, **Arteris FlexNoC** can be configured and customized for interconnections on the chip that can provide best latency, least congestion, and optimize other aspects. Similarly power management can be another area where configuration can be added for power harvesting. So the question is how configurable your IP is so that it can be customized according to different environments? Configurability in your IP adds provisions in your IP to be customized in different environments. This increases the value of your IP to operate in a wider range of possibilities. More often than not an SoC vendor may need to ask an IP vendor to add specific customization in the IP so that it can exactly fit into the scheme of the SoC. This situation may get extrapolated to an extent that the IP gets transformed into a subsystem; proper evaluation of cost for such customization must be done.

Another kind of customization can be for different market segments such as automotive segment which needs wide range of operating temperature and other environmental parameters.

In certain market segments like IoT, where the standards can vary by a large extent, SoC vendors prefer adding custom IP in-house rather than buying from outside.

Cost of Characterization

This is a big area where IP needs investment, specifically at advanced technology nodes where process can vary significantly between different foundries at the same node. It asks for the characterization of IP at every process variant. It depends how much pre-characterization can be done at the IP level; the SoC vendor might ask for special characterization at a specific node of choice for the SoC. A level of prudence can help here. An IP for GPU or mobile processing may need advanced nodes like 14nm FinFET and hence the characterization for process variants at those nodes will be needed. However, an IP for other applications which can stay at higher nodes may not need too many characterizations. But there may be other complications for specific applications. For example, an IP for automotive applications can stay at higher nodes (although moving down from 150nm and 90nm) such as 55nm, 40nm, or even 28nm for specific cases; however that will need characterization for a wide range of temperature and other PVT conditions.

Within an IP the characterization can be at the fundamental unit level such as bit cell and at macro level. The fundamental unit level characterization may not change frequently, but macro level characterization may change according to the design. So, that kind of characterization needs to be planned appropriately.

Cost of Acquisition

The cost of acquisition of IP is a very important aspect for SoC vendors. Large system companies have specific processes laid out for IP selection and procurement. They include

items such as quality of the IP, ease of its integration, the IP vendor's past record and ranking, vendor support throughout the SoC lifecycle, cost and RoI analysis as per single or multiple use of the IP, etc. It's prudent to explicitly mention, specifically in single use, what kind of modifications and support elements such as error code revisions, defect fixes, configuration modifications, and so on are permitted. Also, the fees applicable for reusing and making variants of IP must be explicitly mentioned.

The evaluation of an IP and its integration into SoC is co-ordinated with the associated EDA vendors, development partners, and design service providers along with the IP provider. It's a costly affair and hence it's required that the list of selected and qualified vendors is kept short. The emergence of **eSilicon** as an IP service provider is a step in the right direction for IP evaluation before its acquisition.

An important aspect comes into picture when the IP needs some customization. In this event it's important for both the IP provider and the SoC vendor to determine how the customized code will be maintained in future, whether the changes are generic enough to be merged into the main code branch. If not, then special support for that custom IP branch will be needed, asking for extra support resources borne by either the IP provider or the SoC vendor. So, here the question comes, how much is the support? Is it scalable and profitable for the IP provider to take it in her/his main stream? If not, then is it justified for the SoC vendor to acquire the commercial IP, customize it, and maintain it, or otherwise develop her/his own IP? If it is customization by the SoC vendor on top of the commercial IP, then the ownership rights must be clarified at the time of acquisition.

The cost of acquisition can also be factored in on a long-run production basis where the IP provider is paid on the basis of royalty fees. This can be finalized on the basis of specific terms such as actual sales or shipments. For an IP provider as well as the SoC vendor a typical challenge appears when the fab does not see enough RoI in creating a slot for a particular IP; this needs right level of negotiation before embarking on the journey.

Cost of Qualification

The quality of IP, specifically design IP is a big question mark today. As you obtain the third-party IP blocks from different regions (which may have different quality culture) of the world, it's imperative that they must be qualified in the premises of the SoC integrator before they can enter into the works. A preliminary assessment and evaluation of an IP must have been done as part of acquisition, but its actual qualification from quality and security perspective according to its intended use in the actual SoC environment must be done in-house by the SoC vendor. For example, an IP for automotive applications must be tested in the SoC environment under the intended temperature ranges and possible electromigration effects. This is essential to ensure the IP is bug-free to avoid resolving the IP bugs at the system-level which can be very unworthy and extremely costly. It's not a complete verification of the IP or system; I will talk about verification in detail in a subsequent section. It's a quick qualification of the IP in focus. **Fractal Technologies** has a tool called **Crossfire** which has an option to quickly qualify your IP with all formats and the SoC environment in which the IP is intended to be used. Similarly there is **IP Kit** from **Atrenta** (now **Synopsys**) available at **TSMC** for partners to qualify their soft IP against Atrenta's **SpyGlass** provided checks. There may be other commercial or internal tools as well to assist in this acute need of qualifying the IP before its use.

Cost of Integration

Although the cost of integration of an IP into an SoC falls into the purview of the SoC vendor, the onus of integration may come on the IP provider who needs to make sure the IP or subsystem along with the software bring-up works properly in the SoC environment. The matter of the fact is that the SoC integrator does not understand all of the IP that are going to be integrated into the SoC; so the SoC vendor has to hire domain experts in different areas. The IP provider in most cases has to work at the subsystem level when integrating an IP into an SoC.

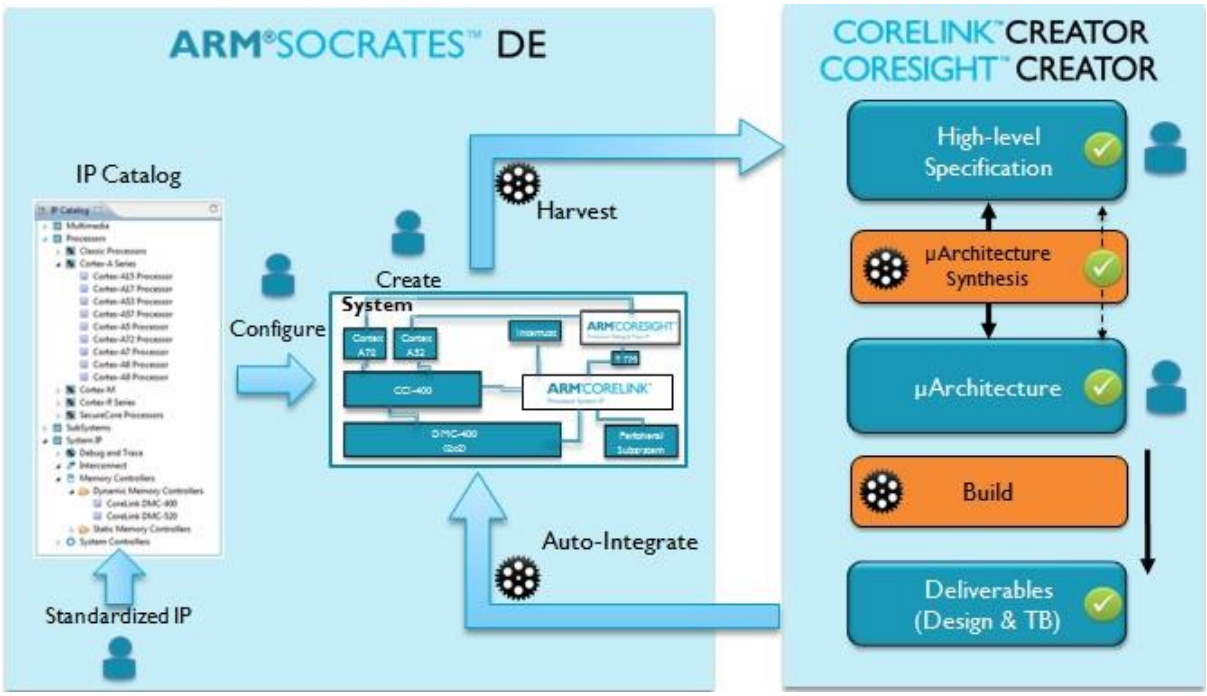
For an IP provider it's like envisioning the system requirements and making provisions for those while working at the IP level. For example, in case of an interface IP the whole channel has to be modeled as per the system requirements; for a physical IP, the PPA has to be modeled with right level of trade-off according to the system requirements. The PHY has to be programmed to make a trade-off between PPA. In modern age, the range of PPA can be very wide in which case the IP has to be segmented into high-range for performance critical applications, mid-range to save power, and low-range to save cost. Again, for an IP with FinFET and smaller geometries, you may need to increase the area to spread the heat. Also the FinFET process varies between foundries, so if you are sourcing an IP from two different foundries, then you will also have to spend in unifying those characteristics to keep the final SoCs uniform. As an example **Apple** sourced A9 SoCs for its iPhones 6s and 6s Plus from **TSMC** 16nm as well as **Samsung** 14nm foundries. So, at the system level their characteristics have to be matched to keep the power and performance uniform for all phones.

Another level of complexity comes when you integrate analog with digital on the chip. The analog portion needs confirmation from the foundry, so it's advisable to keep the analog content as less as possible. However moving things from analog to digital can be another complexity and will surely incur cost.

Considering an IP from system perspective, it's important that the package aspects are taken into account. This includes effects such as noise, signal integrity, ESD, and so on.

More and more of system companies prefer team integration with the IP vendors where the IP team can work with the system team and contribute in SoC roadmap development for future innovative technologies. This is a smart move for giving a lead time to future technologies, provided confidentiality is maintained. These days we are also seeing full merger of IP companies into SoC companies. A flip side to these mergers is that a sustained continuation of such mergers may defeat the purpose of IP-based business model in the longer run.

While integrating an IP into an SoC and optimizing it within the cost parameters for a particular target segment, the integration has to go through several trials to obtain the best optimized architecture. To save the cost of these trials, the IP and SoC industry is gearing towards automating this effort.



[Courtesy ARM]

ARM has already developed tools for such automation. In the picture above there is **ARM® Socrates™** design environment along with **CoreLink™ Creator** for interconnect optimization and **CoreSight™ Creator** for debugging. Read the article, "New Tool Suite to Accelerate SoC Integration" for more details on the "**ARM IP Tooling Suite**".

Cost of Verification

This is the most significant cost in an SoC, almost 2/3rd of the total cost of the SoC. There are multiple verification engines for simulation, emulation, formal verification, virtual prototyping, FPGA prototyping, and post-silicon verification at different stages of an SoC design. The key idea is verification closure through complete coverage of the overall SoC. It's in general very difficult and hard problem to get complete coverage of an SoC. These days the SoC vendors bring up the whole system and run applications on full chip through emulation, or validate the SoC through FPGA prototyping which has its own limitations. There is no way to guarantee complete verification other than the verification coverage metrics, so coverage driven verification gained importance where coverage obtained through different verification methods gets added up.

With the expansion of SoC's size and complexity including hardware, software, and firmware, the verification space of SoC also has expanded enormously. In such as scenario, imagine a configurable IP getting added up. How do you ensure all of its configurations are validated? It expands the verification space further, multiplied by the number of configurations. This keeps the cost of verification increasing.

In the IP-based business model, along with the design IP, the idea of verification IP (VIP) also came with the sole intention of verification automation and reuse of test plan, test bench, and test suite across multiple designs to boost verification productivity. Similarly, standard verification methodologies came into practice for verification automation and testbench reuse, UVM (Unified Verification Methodology) being the most popular. However,

UVM is good for IP and at most subsystem level. The system level test is the bottleneck and that is where the verification space blows up. At the system level, as we have seen, there are multiple verification engines in work and there are multiple IP and subsystems. There is no automated methodology and reuse of tests and testbench across these multiple design levels and engines. This increases the cost of verification exponentially.

To automate verification and reuse at the system level, key initiatives are going on to establish software driven methodology based on use-cases and test scenarios. **Accellera** has initiated a Portable Stimulus Working Group (**PSWG**) to establish a common standard of test and stimulus which can be used by a variety of users across different levels of design hierarchy (IP, subsystem, and system) under different execution platforms (simulation, emulation, FPGA prototyping, post-silicon, or any other) using different verification tools. **Cadence**, **Mentor**, and **Breker** have developed tools for system level verification; and they along with other PSWG contributors are working on establishing the common test standard and making their tools compliant with this standard. This can definitely start a new chapter in verification from system-level and reduce the burden of cost of verification by a large extent through test automation and reuse across multiple design hierarchies as well as verification engines. However, this methodology is yet to be established and needs semiconductor industry level investment, effort, and time. Read the article "Moving up Verification to Scenario Driven Methodology" for more details on this initiative.

Cost of Learning

AS the complexity of SoCs keeps growing the verification space is always open for new learning. In the last section, we talked about the use-case based verification methodology which is being explored for verification at the system-level. Even after this methodology is established, there will be a cost involved in training the verification engineers on this new methodology.

In verification space newer formats, protocols, and standards keep emerging, especially in IoT, mobile, and automotive segments. This needs budget, time and effort set aside for design and verification engineers to learn. Moreover, the verification engineers need to have complete knowledge about the system and the environment or market segment in which the system is going to operate.

Apart from the system and design, the learning has to happen at the process and foundry level as well. The advanced process nodes such as FinFET nodes are distinct with foundries and hence the learning effort gets multiplied. The design, verification, and process engineers need to work closely to understand the new process and its associated rules. The process engineers may need to work in the actual foundry environment to learn about the advanced technology aspects and impart that knowledge to the design and verification engineers for them to incorporate the same in the design and its verification.

Cost of Redundancy

A peculiar scenario arises when you configure an IP to serve multiple requirements for different market segments. Not all the segments are served at the same time, so there are unused portions in the design and those can vary in different situations. This brings redundancy along with configurability. The redundant circuit may also consume power

unnecessarily if not architected well enough to remain shut when not in use. Such wastage of power may not be completely eliminated through multi-mode operation of the SoC. Another case of redundancy comes when an IP characterized for a particular technology node is no longer reusable for another technology node through the use of automated technology migration tools; the IP needs a fresh architecture and rework. In such cases, a proper ROI analysis must be done for the IP to remain profitable in single use.

Summary

The costs described above are known by their simple terminology; however they need to be better understood in the modern context of IP so that the right level of investment in the IP can be justified from both sides – the IP provider and the SoC integrator. It is important that the value of IP continues to increase to keep the SoCs differentiated and the IP-based SoC development model healthy. To achieve this there must be a win-win situation between the IP provider and the SoC vendor with proper understanding of the costs involved.

There are methodologies being explored to reduce different types of costs through automation. However, it depends on the specific methodology and the type of cost that can be reduced. Moreover, the automation methodologies such as system-level synthesis and verification need to be established before they can be explored for wider use. Intermediaries such as **eSilicon** have come up that provide IP services for SoC vendors to do a pre-evaluation of IP integration into their SoCs before buying it. The eSilicon business model is very flexible where they can be paid for an IP either after pre-silicon evaluation or after production. The semiconductor ecosystem is trying to establish a series of drivers that can keep the IP costs in control and the IP-based SoC development model afloat.