**Application Platforms and Synthesizable Analog IP**

**Peter Santos**
**Vice President, Marketing & Business Development, Barcelona**

## Abstract

Perhaps the most compelling new trend in system-level semiconductor design is the concept of application "platforms". Such platforms are intended to make the development of complex, application-specific systems-on-chip (SOCs) an order of magnitude faster and more reusable. They may provide either functional and infrastructure IP, or both, as well as definitions and interfaces for layers of abstraction in the design process. Functional IP may include microprocessors, DSPs, specific logic functions, mixed signal, memory and interfaces (USB, Serdes, etc). Infrastructure IP may include clock and power network architecture, cell libraries and test setups.

The intent of such platforms is to enable rapid customization of complex chip designs. This is typically accomplished through modifications in software or small blocks of logic, while the functional IP is basically static.

However, the key to making platforms truly flexible, and a vehicle for highly differentiated semiconductor products with fast development time, is to make the functional IP rapidly customizable to the greatest extent possible. There are already companies offering configurable memories and processors, as well as firms offering libraries of such functions, and these capabilities are seeing significant adoption for complex chip designs.

The most difficult functional IP to render customizable is analog and mixed signal blocks. These functions are tied very closely to process technology and vendor, and have for years resisted not only configurability, but also even the accumulation of a library of functions (due to the difficulty and time to implement an analog function in a target process).

This is particularly problematic in light of the now well-documented increase in analog content in SOCs, a trend that will only accelerate. The fastest growing part of SOCs, and that which has in many cases the greatest potential for differentiation, is the most resistant to configurability.

Limited innovation in analog design methods has been a primary cause for the lack of customizable analog solutions. Developments in synthesizable analog IP are changing this state of affairs, however. It is now possible to very rapidly (minutes to hours) synthesize a complex mixed signal function such as a PLL, op amp or data converter, from specification direct to placed & routed GDSII. This type of IP has been proven in silicon in 0.18 and 0.13um CMOS technologies, and can make a huge contribution to the utility of platform-based design.

With synthesizable analog IP, a platform-based designer can now not only customize the analog and mixed signal sections of the chip, but they may adapt these circuits to changing chip level requirements up until a few week before tapeout. This is in contrast to current practice where analog IP is totally fixed and the design must accommodate the IP.

## Introduction

The focus of this discussion will be on platforms for the implementation of application-specific SOCs. Generic platforms for system design, silicon infrastructure, and programming, while important, are not considered here.

## Purpose of Platforms

The principal objective of application platforms over time has been to allow systems designers to simultaneously achieve the often-conflicting goals of managing complexity, fast time-to-market, and achieving differentiation for their products. They have been in existence for decades in fields such as automobile design and computers, but recently complexity pressures have brought the platform concept to receive strong attention in semiconductor design. In the electronics industry, "reference designs" have often served the same goals as are now being envisioned for semiconductor IC platforms.

### Drivers

The driving force in consideration of platform-based design of chips is of course complexity, plus the perceived opportunity on the part of platform builders to leverage a single design for many different individual chip instances. Importantly, the complexity driver comes not just from increasing system functionality, but also from the desire to reduce system cost, and therefore to integrate more and more functionality onto a single device. Further, the economics of platform reuse are based largely on the emergence of a small number of specific applications, which are the target of significant systems & device development investment. These well-known applications in consumer multimedia, wireless systems, datacom/telecom and storage, share in common a heavy emphasis on real-world interfaces, and rapidly increasing operating frequencies.

### Objectives

The ultimate objective of platforms is to make enough of a design fixed, be it the processor, interconnect, functional IP or memory, so as to accomplish fast, reliable time to market. But it must also allow the user to apply significant differentiating value, without losing the design time advantages conferred by leaving the other part of the design fixed. Also, any manufacturing cost penalty of using a platform must be small enough as to be outweighed by reduced time-to-market and design risk. A corollary objective is to allow the platform user to focus on their core competency and value, which is generally in system-level design and architecture. A key requirement for platforms is that they enable system-level tradeoffs and analyses to be made quickly.

The platform provider is rewarded for enabling this by either being able to sell the manufactured platform instance units to the user, or by receiving license fees from the user. Thus, for the platform provider, the objective is to achieve significant financial returns on an investment in a single, flexible or customizable system design. The return is achieved through individual sales of platform designs, but also (preferably) through locking users in to a proprietary platform programming or customization methodology.

## The Role of Analog

Interestingly, in all of the considerable public discourse regarding application platforms there is little discussion of the analog/mixed signal aspect of platform development and use. While there are certainly numerous important and difficult problems to solve around processor programming, bus architectures, RTL coding and the like, the analog content of platforms merits much greater attention than given to date, for a couple of important reasons.

First, as noted above, integration is a key driver of the complexity management imperative fueling platform-based chip design, and with integration inevitably comes analog functions as it reaches the edge of the system. Second, the key applications in consumer, communications and computing/storage that can economically justify platform development are analog-intensive. As integration progresses, analog is always at the edge in these types of systems.

Multimedia interfaces, high-speed interconnect & clock distribution and communications transmit/receive channels are all mixed signal subsystems, and are very often key components of an application system platform.

**Types of Platform Approaches & Analog Considerations**

There are as many types of platforms and definitions as there are types of vendors serving the electronics design and component industry, meeting the needs of many types of end users, depending on their functional and application competency. These different types have advantages and disadvantages with respect to analog/mixed signal function. Before detailing these however, it is worthwhile to consider a somewhat more independent and comprehensive definition of the platform-based design construct.

Alberto Sangiovanni-Vincentelli of the University of California, Berkeley, and the Gigascale Semiconductor Research Center (GSRC) defines a platform as "a layer of abstraction with two views." The upper view allows an application to be developed without referring to the lower levels of abstraction. The lower view is a set of rules that classify a set of components belonging to the platform.

In this view, a key requirement is to construct the platform such that higher-level design work such as system architecture exploration and tradeoff can be done reliably without consideration of the lower level platform layers, yet with confidence that the final implementation will reflect the assumptions made in system-level design.

This is a challenge that is particularly difficult with regard to analog/mixed signal systems.

**Types of Platform Targets & Analog Considerations**

There are a variety of types of application platforms available in the market today. We can think of these in terms of both the form in which they are provided, and whether they primarily exploit systems design or implementation technology.

In terms of form factor, we see three basic types of platform solutions:

> Standard product chip-based
>
> These are programmable hardware or software devices, manufactured in large quantities, provided together with a proprietary programming interface and sold at a premium to semi-custom or custom chips of comparable size. This category includes FPGAs, hybrid FPGAs, general-purpose processors and application-specific processors such as Texas Instruments OMAP. The design cycle for these devices consists solely of software programming.
>
> There is generally little or no provision for analog components in these devices.
>
> Semi custom chip-based
>
> These are offerings from chip companies with ASIC or ASSP capabilities, designed to leverage both their semi custom methodology expertise and their IP portfolios to provide customers with a platform which can be more flexible and less costly than a standard programmable device. These devices, such as basic SOC offerings or emerging platform offerings such as LSI Logic's Rapidchip or Philips Nexperia, may have significant amounts of fixed hardware and architecture.

The design cycle for users of this type of platform includes at least some hardware design, but affords the opportunity to put some key functions in hardware for performance or cost reasons, and may provide flexibility in the analog functions that may be integrated.

Semiconductor IP-based

Provided by semiconductor IP vendors, these solutions require full hardware implementation design cycles and may or may not include analog specifications, depending on the nature or the application, and the competency of the provider. This category includes focused system platforms such as ParthusCeva's Bluestream and processor-based platforms such as ARM's Primexsys.

In general, these systems are not committed to hardware and offer higher analog flexibility.

Overall, we can see in figure 1 a tradeoff between the time-to-market advantage and the analog/mixed signal flexibility offered by these different approaches.

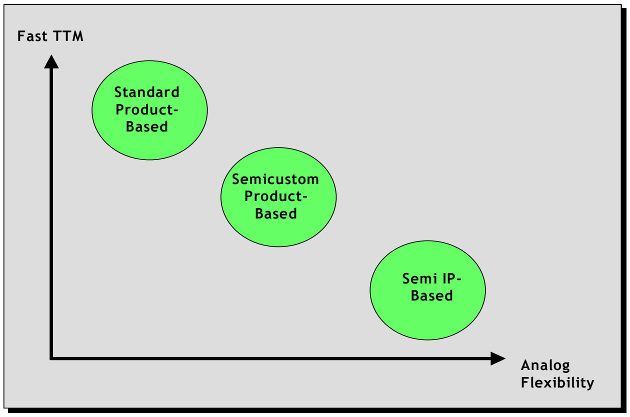

Figure 1. Time-to-market vs. analog flexibility

A further distinction among these platform types, as seen in figure 2, is whether they are based on applications or implementation technology. Generally, technology-based application platforms offer less focused IP, including analog, than application-based, which rely on systems knowledge of which functions are needed.

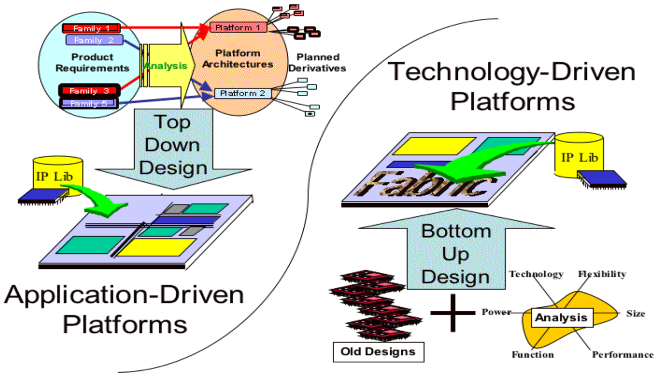

Figure 2. Different platform types. Source: Bob Altizer, Larry Cooke, and Grant Martin, EE Design, Nov 13, 2002

Last, we can classify individual company approaches by their form factor as well as whether they are technology or application based. This provides platform users with options to complement their core competency, as well as a choice of where to make the flexibility/time-to-market and cost tradeoff. What is still lacking however is the means to effectively deal with analog/mixed signal content in a true platform design paradigm.

| | Standard product-based | Semicustom product-based | IP-based |
|---|---|---|---|
| Application-based | TI OMAP | LSI RapidChip<br>Philips Nexperia<br>Toshiba SOCMosaic | Parthus |
| Technology-based | Xilinx Virtex | IBM Bluelogic<br>LSI SOC | ARM<br>Sonics |

Figure 3. Platform types vs. form factor.

**Platform Definition**

The role of analog in the definition of 1st generation of application platforms has been either non-existent, or inflexible, depending on the type of platform. Analog is in some left to be implemented separately by the system designer, as in standard product-based approaches, or with some technology-based IP approaches. In others, such as semi custom product-based approaches with fixed IP blocks, a fixed analog IP block is provided.

System Design Time-to-Market and Flexibility

The problem this presents is that it leaves the analog portion of the system out of the platform paradigm. It cannot be flexed during system-level tradeoff analysis as can digital circuits, because it is either a) not present in the system architecture provided, b) present by an inflexible block or c) present in the form of a high level model which cannot reliably represent what is achievable at the next layer of abstraction down.

In these cases, time to market is lost due to the requirement to handle analog separately, outside the platform-based design methodology, or flexibility and the opportunity to differentiate in this part of the system is lost. Further, the return to the developer is diminished; the system platform is either incomplete (where analog is not included), or the inflexibility of the analog subsystem limits the range of use of the platform.

As integration continues in the analog-intensive applications that are the target of platform development, this exclusion of analog/mixed signal from the platform paradigm becomes more costly, compromising the key benefits of the platform approach.

**Basic Requirements & Considerations for Platform Effectiveness: Analog Aspects**

To address this problematic aspect of application platforms, several criteria must be met:

## Flexibility

The analog blocks must be flexible so that a wide range of system specification can be explored. The types of systems targeted by application platforms have analog specifications in terms of accuracy (resolution) and frequency that can span a wide range, as shown in figure 4.

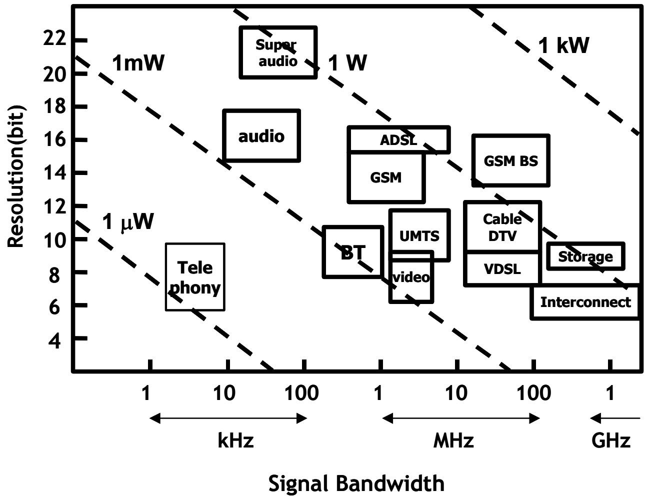

Figure 4. Analog requirements of selected applications. Source: ITRS 2001.

## Synthesis

It must be possible to represent the analog blocks or subsystem at a high level of abstraction, which can be reliably synthesized down to a lower, implementation level.

| Abstraction Layer | Design Flow Deliverables | Benefits |
|---|---|---|
| System | High Level Specs (Ex: Jitter, BW, Power, Area) | System-level trade-off analysis |
| RTL | Verilog -D/AMS | Functional Testing |
| Gate | Spice Netlist | Functional and Timing Testing |
| Silicon | GDS II, LVS Netlist, .lef | Easy integration, Physical Testing |

Figure 5. Analog abstraction levels.

## System-Tradeoff Capability

It must be possible to evaluate the implication of different system requirements within the analog subsystem itself, evaluating the impact of analog or digital system specs on performance, cost and reliability.

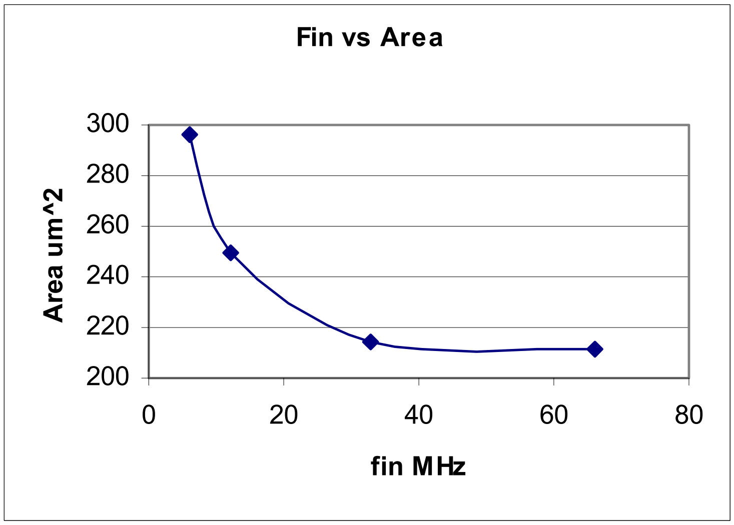

Figure 6. Analog system tradeoff analysis. Source: Barcelona.

Implementation Speed

The implementation of analog blocks from flexible, high-level specifications today is very slow. The synthesis process must be very fast, within a small portion of the overall design cycle.

Reliability

The resultant implementation must be highly reliable, without any intervening silicon test chips for verification purposes.

**Configurable Analog IP**

We can define configurable analog components as blocks which can in some fashion be expressed at a high level of abstraction and reliably be converted in an automated way to a lower level (physical) implementation. In an application platform-based design approach, these would take the form of one of the semi-flexible system-level elements available to the user. It is semi-flexible in that it can perform a range of performance specifications, but has some limitation in range of function or architecture.

There are other existing types of semi-flexible, or synthesizable components that are seeing use in platform based design. These include configurable memory from companies like Virage Logic and configurable processors from companies like Tensilica. In all cases, these types of components offer flexibility and time-to-market in exchange for the some limitations on architecture, and the use of proprietary configuration systems.

Approaches

A variety of approaches have been taken to bring configurability to analog blocks. These include analog arrays, metal configurability/tiling, simulation toolkit-based analog and geometric programming- based analog.

**Analog arrays** from companies like Rohm provide an array of transistors that can be configured with metal. This approach suffers from cost and performance limitations imposed by the array structure, is not a good candidate for platform-based design. It also offers limited ability to program at a high level of abstraction.

**Tiling** of lower level blocks achieves better performance and cost than arrays, and can be programmed based on performance-level abstraction reliably.  However, they cover a very narrow range due to the fixed nature of the analog sub blocks, and offer little flexibility.  . Tiled versions of phase locked loops are available from companies like ParthusCeva and Virtual Silicon.  They are best suited to platforms where very little value comes from the ability to change the analog block.  This approach is best suited to metal-only programmable solutions such as FPGAs.

**Hard analog IP** blocks are another alternative.  Hard IP does enable abstracted levels of design due to availability of higher-level models, and can sometimes be made configurable such as through programmable bits in a PLL. Programmable PLLs can be used within an analog block with elements such as converters to enable a level of flexibility. The trade-off is that there is usually an area penalty to using such a block and this directs results in cost inefficiencies that make this a non-viable solution for platforms since platforms are best suited for high volume cost-sensitive end-user applications.  This approach can be applied to a variety of platform types.

**Geometric programming-based** (GP-based) analog components from companies such as Barcelona, and inside academic groups, rely on the coding of circuit and process characteristics into a geometric program.  Then a solver is used to synthesize the physical circuit that will achieve the users system performance specifications.  This approach is functionally limited to whatever circuit/process combination has been coded, but offers a reliable system programming level of abstraction, and very fast (hours) implementation times.  It also readily enables system performance tradeoffs and is broadly applicable to a variety of platform types.

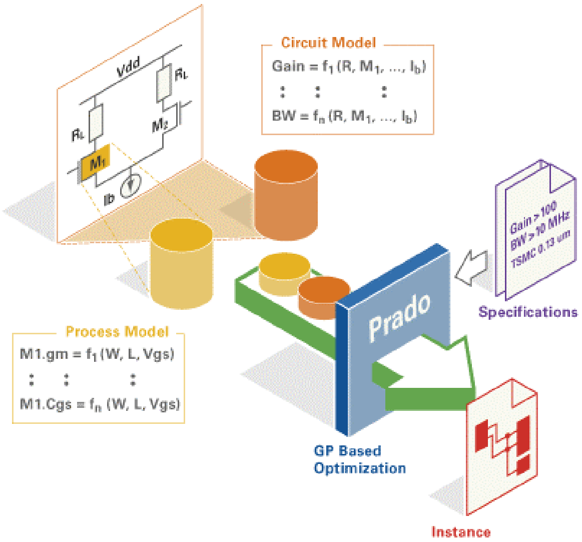

Figure 7.  Geometric programming-based analog IP.  Source: Barcelona

Thus, while there is a range of options for configurable analog IP, it is clear that GP-based synthesizable analog IP offers significant advantages for frequently used functions. As a focus of application platforms is on very specific types of systems, where the same types of functions are likely to be used repeatedly, there is a very good fit.

| | Analog Array | Tiler | Hard | GP-based |
|---|---|---|---|---|
| Flexibility | Med | Low | Low | Med |
| Synthesis | Low | Low | Med | High |
| System Tradeoff | Low | Med | Low | High |
| Implem Speed | High | High | High (Low if changed) | High |
| Reliability | Med | High | Med | High |
| Cost | High | Med | Med | Low |
| Applicable Platform | NA | FPGA | All | All |

Figure 8. Configurable analog IP approaches vs. platform requirements

## Building Synthesizable Platforms

In the development and provision of application platforms, the use of synthesizable analog IP has important implications.

### Definition & Design

The range of applications that can be targeted in an application platform business plan can be significantly wider using synthesizable analog IP than with hard analog IP. Instead of using a single fixed block, or a selection of blocks (most of which would be unused), a single synthesizable block can be instantiated in the platform, covering a wide range of possible system performance specs.

Further the solution can be more complete than one that does not include the analog subsystem at all. Together, these translate into higher design wins and/or higher value and pricing.

Also, the use of synthesizable analog IP enables the designer to build in powerful analog system tradeoff capabilities in an integrated flow.

### Support

A frequent support requirement for provider of semi custom product-based application platforms is to tweak or change certain fixed elements of the platform to meet key customer requirements. When these changes are needed for the analog section of the platform, long design and verification cycles are involved, possible even a silicon spin.

With the use of geometric-programming based analog IP, the modification of the analog section is push-button. The circuits generated by this method are correct-by-construction and are thus pre-verified and reliable. This results in dramatically lower support costs.

Analog Considerations

The use of true synthesizable analog IP, such GP-based IP, makes the analog subsystem truly programmable, bringing the analog domain of an application platform into the same system design paradigm as the digital domain. It is possible to go from a system performance specification reliably and directly down to the implementation level of abstraction.

### Using Synthesizable Platforms

For users of application platforms, the inclusion of synthesizable analog IP offers significant benefits. They are able perform system design analysis and tradeoffs of the entire system, including both the digital and analog domains. They can be confident that the system abstraction level will accurately represent what can be implemented. And the customization time required for the synthesizable blocks will be extremely short.

Effectively the knowledge and expertise of an analog circuit designer is placed, in an automated fashion, in the hands of the system designer.

### Summary

The primary objective of application platforms is to achieve the best possible combination of design differentiation and time-to-market in an environment of enormous implementation complexity. The use of synthesizable analog IP in the design and utilization of platforms brings this combination of high flexibility and fast time-to-market to the analog subsystem, enabling platforms to provide much greater utility to users, and better returns on platform investment on the part of developers.

While truly synthesizable analog IP has only recently emerged in industry and academia, it holds tremendous promise in allowing platform users to bring analog up to a higher level of reliable abstraction. With this model, analog systems analysis and programming can become a reality, accommodating the escalating analog content in the applications where platforms are most needed.

---

Thanks to Amit Nanda and Bart DeCanne of Barcelona for their assistance with this paper.