

The Changing Landscape of System-on-a-Chip Design

Ann Marie Rincon, William R. Lee, Michael Slattery

IBM Microelectronics
Essex Junction, Vermont 05452

Abstract

The 1940s are remembered in electronic history for the invention of the transistor, and the 1950s for the invention of the integrated circuit. As the millennium comes to a close, another major development in the world of electronic design has come to the forefront: the ability to integrate an entire "system" onto a single computer chip. System-on-a-chip (SOC) designs have been in existence for only a few years; however, in that span of time, the definition of the "system" designed and manufactured on a chip has significantly changed and expanded as did the technology, skills, tools and methodologies required to produce it.

This paper examines the changing characteristics of SOC design, from the mid 1990s when there was a limited number of processor-based in-house designs, to the present-day definition which includes multiple processors, peripherals, communication subsystems, bus hierarchies, analog components, and radio frequency (RF). The increasing levels of SOC integration will be illustrated by a family of set-top box designs.

I. Phase 1

Processor-Centric Turnkey Solutions

Dataquest defined an SOC design as "an integrated circuit that contains a compute engine, memory and logic on a single chip(1)." This adequately describes the early SOC designs and SOC companies in the market during the mid 1990s. In 1994 Motorola announced a system known as FlexCore™ for creating custom microprocessors based on the 68000™ and PowerPC™. In 1995 LSI Logic made headlines with an SOC designed for Sony that contained a MIPS microprocessor, memory, and proprietary Sony logic which has been used in millions of Sony Playstation™ video games. In 1996, IBM manufactured its first SOC ASIC containing a PowerPC 401™ microprocessor, SRAM memory, a high-speed analog memory interface and proprietary customer logic.

The Playstation chip, like many of the early SOC chips, was designed not by the end customer, but by the ASIC vendor, to meet the customer's specifications. The number of SOC designs in a given year was limited, in many cases, by the design resource available at the ASIC vendor so there were only a few applications but they had extremely high volumes. As the applications for SOC grew, the marketplace demanded

more broad access to SOC design. SOC providers had to expand their focus beyond in-house design capability to develop and deliver SOC design kits and methodologies directly to the end customer.

II. Phase 2-Move to the Field

The Early Issues

Moving the design of SOC designs out of ASIC vendors' hands and into the hands of multiple end customers required a robust methodology and focus on several key areas related to delivering complete SOC design kits. These included the availability of intellectual property (IP), IP protection, designing IP for reuse, and the use of standard interfaces.

The inherent complexity of these designs required a proven methodology to handle high-performance, high-density chips. IBM chose to integrate its SOC offering into its existing ASIC design methodology, including static timing sign-off and full scan design-for-test(2). As a result, each IP block, or "core," offered in an SOC design kit had to be fully modeled for each tool in the methodology(3). Required simulation, synthesis, static timing, full-scan test, and layout models were developed for each hard core. Synthesizable source, synthesis scripts, technology-specific netlists and static timing assertions were developed for each soft block.

Most early core IP was owned by, or licensed through, silicon vendors. Exporting the design data outside vendor walls drove rigorous IP protection requirements. IP owners wanted to deliver key design data in abstracted or non-human-readable formats that would be still usable by the end customer's design tools. Methods for compiling simulation data into executable binary models (3) have evolved, as well as standards for describing and compiling detailed timing data (4), which primarily protect critical hard IP such as microcontrollers or digital signal processors (DSPs). Soft IP was generally delivered as source protected by license agreements.

Beyond Microcontrollers

The applications demanding SOC solutions became more diverse and rapidly expanded beyond the original SOC definitions of compute engine, memory and customer logic. Wired communications applications added Ethernet, Sonet, Serdes, and the universal serial bus (USB). Consumer applications added MPEG,

6.1.1

83

Firewire™, NTSC/PAL and Video DACs. Data processing applications added AGP4X, ALDC, VGA, PCI and Rambus™. Wireless applications required DSPs and RF components.

The list of required IP components became so large and diverse that it was impossible for a single SOC company to develop, maintain and migrate the entire catalog of IP functions required by their customers. The marketplace responded with a growth explosion of independent third-party IP developers such as Advanced Risc Machines (ARM™), Rambus Corporation, Virtual Chips and Inventra. These companies developed and licensed IP for fabrication at multiple foundries. Companies such as ARM and Rambus focused on a single highly optimized function that was translated into hard IP at the ASIC vendor. Virtual Chips and Inventra offered more general libraries of soft IP that end customers purchased and then synthesized into their target ASIC vendor's library.

All IP Is Not Created Equal

A fundamental realization that hit the industry in the early stages of SOC design was that all IP was not equal for effective integration and use by multiple customers in multiple applications. Whether designed by the silicon vendor or third-party provider, the IP, hard or soft, had to be initially designed as a reusable component. This requires two to seven times more resource than the engineering investment for one-time use (5). However, without the parameterization, documentation, testability and unit verification that re-use demands, the IP is difficult, if not impossible to be used by anyone other than the original designer. Therefore, in 1996, organizations such as RAPID and VSIA were formed to address standard deliverables for re-usable IP.

Standardizing Interconnect

Standardizing the IP deliverables and widely available IP were important steps toward expanding the reality of SOC. But a third ingredient, standard interfaces or on-chip buses, was needed to truly enable IP component integration into a single chip. In the infancy of SOC design there were as many design interfaces and communication protocols as there were IP blocks. Sub-optimal glue logic was often required to integrate one or more cores with each other and with unique customer logic. The benefit of using pre-designed and pre-verified IP blocks was at times lost due to the difficulty of integration. Standards for on-chip bus structures began to evolve. IP was developed to communicate with one or more of the dominant processor bus architectures such as PowerPC and ARM.

The availability of diverse and truly re-usable IP, designed to integrate with common on-chip bus structures began to substantially change the number, size and content of SOC designs. The SOC designs of 1998, con-

taining 30 or more cores with multilevel bus structures, are far more complex than the processor plus memory-based SOC designs of the mid-1990s (Figure 1). The number of SOC designs processed at IBM more than tripled between 1996 and 1998. Average chip size and the number of cores per chip also grew by almost 300%. The maximum number of cores on a single chip had grown from three in 1996 to well over 30 in 1998, and the percentage of core content had increased from less than 50% to almost 95% in some cases.

Table 1: SOC Design Characteristics

Year	SOC Designs	Average SOC Design Size (Gates)	Average ASIC Design Size (Gates)	Cores/Design
1996	3	280K	500K	2
1997	7	385K	700K	4
1998	12	790K	1,100K	9

Processor-based designs, specifically PowerPC, account for 57% of the designs manufactured from 1996 to 1998. Non-processor applications include chips using PCI, graphics acceleration, data compression and analog functions such as video DACs and NTSC/PAL. Only 30% of chips currently in layout or in logic design are processor-centric (PowerPC, ARM or DSP), indicating a continued expansion of new applications requiring reusable IP and SOC design techniques.

Design Gap vs. Time-to-Market Gap

The data in Table 1 actually runs contrary to the prediction that SOC designs, through their use of pre-designed blocks, would lead the way in gate utilization, thereby closing the growing gap between the number of usable gates on a die and the number used. The experience at IBM is that designs for high-end, high-performance data-processing applications, and not SOC designs, are the most aggressive in terms of gate and I/O count.

Table #1 shows that the average size of SOC designs, although growing each year, lags the average ASIC design size at IBM by 30-40%. The largest ASIC currently in process contains over five-million gates and 1,000 I/Os, more than four times the largest SOC. SOC designs, however, have the most aggressive time-to-market requirements, and therein lies the primary value of SOC design -- filling the time-to-market gap. The average spec-to-silicon design schedule has dropped from nine to 12 months in 1997 to six to 9 months for 1998 designs.

The growing IP availability and the methods to protect it, coupled with standard deliverables and on-chip buses, has enabled SOC designs to achieve new levels of integration and complexity. Continued progress presents a new set of challenges related to building

6.1.2

and verifying these designs while continuing to decrease the design cycle. The following case histories detail challenges faced in designing today's SOC's.

III. Case Histories

Evolution of Set-Top Box Integration

While factors such as standards, technologies, applications and transmission methods affect the product definition of a set-top box (STB), the primary driver is cost. It is not unusual for an STB to be sold at a loss because the cost of the STB hardware is typically recovered in a one-year subscription contract with the satellite service provider.

The case histories describe three generations of STB design developed over a period of three years in successive technology generations at IBM (0.45um, 0.36um, 0.27um). The amount of function in each generation remained the same, but the level of integration increased, reducing the overall chip count. The part-number reductions combined with the migration to denser technology provided significant cost reduction and improved performance.

Pre-SOC: Discrete Standard Products

A discrete STB solution composed of standard products manufactured in 1995 is depicted in Figure 1. In this implementation several 0.45um standard product chips were integrated into a single 0.36um chip using IBM Core+ASIC methodology(3); i.e., three pulse width modulators, two UARTs, an SCP interface, IEEE-1284, Smart Card Interface (SCI) and general-purpose I/O (GPIO). The resulting product, known as the set-top-box peripheral chip (STBP), provided all the necessary I/O and communication needs for STB. All other STB components, including the 33MHz PowerPC 403GC processor, MPEG transport and decode were implemented as individual components.

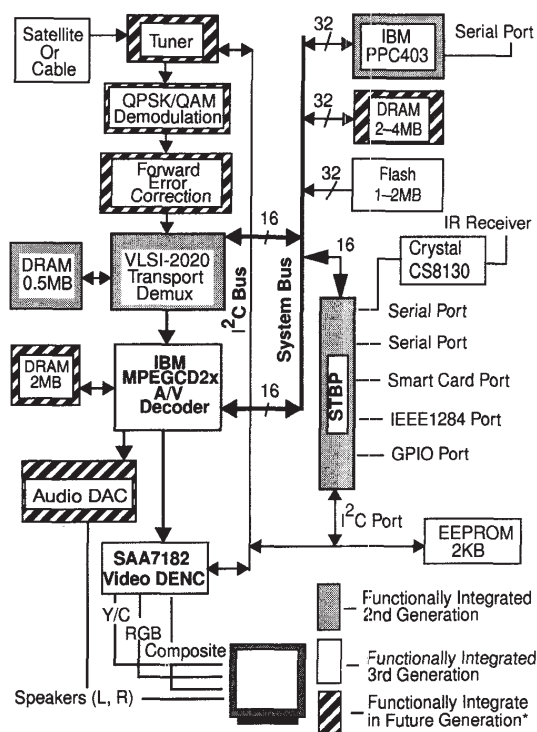
Although the STBP chip did not contain a processor, the design team experienced many issues associated with early SOC's. The chip was constructed using pre-design logic functions targeted for reuse in the Core+ASIC library. The IP was not all complete or in a re-usable form when required by the integration team and multiple IP integration iterations were required as bugs were discovered and functions were completed. The peripheral functions were integrated with the processor memory bus macro by using custom glue logic rather than a common bus interface macro.

Verification with event-based simulation was possible within the targeted time frame because of the relative simplicity of the peripheral functions and an overall chip size of approximately 100K gates.

Small SOC + Standard Products

The second-generation STB solution, manufactured by

IBM in 1996-1997, integrated the PowerPC compute engine, the transport demultiplexor, and the peripherals onto one chip, creating a true SOC. The MPEG-2 A/V remained on a separate chip, leveraging standard products available from companies such as IBM, SGS-Thompson and LSI Logic.



* - Indicates future potential integration, NOT a product roadmap

Figure 1- Set Top Box System: Integration Roadmap

The primary motivation for the CPU integration, transport and peripheral functions was to lower cost by replacing three chips with one integrated SOC. Although the two-chip solution remained in 0.36um technology (therefore, total silicon area remained approximately the same), significant cost reductions were achieved through the elimination of chip packages.

Cost was further lowered by integrating the transport DRAM with the CPU system DRAM, which further reduced the chip count. Data movement from the transport unit to the CPU was accomplished by passing a memory location pointer rather than copying blocks of memory from one chip to another, which reduced the system bus bandwidth over the discrete solution.

IBM's on-chip bus architecture was used to integrate the PowerPC, transport, and peripheral functions,

which allowed faster integration, eliminated inefficient glue logic, and provided a performance advantage over the previous discrete chip solution. The on-chip bus architecture provided separate read and write data buses with a shared arbitration. This split-transfer architecture offered overlapped read-and-write data transfers which allowed the CPU data cache to write to a peripheral while the CPU instruction cache fetched instructions from memory.

Integration of the processor and transport function with the peripherals brought the overall chip size up to approximately 600K gates. Functional verification was still achieved using event-based simulation.

Set-Top Box: Fully Integrated Solution

The third-generation solution in 1998 integrated the previous two-chip solution into one SOC: a complete set-top-box controller. The product chip, called the STB01000, contains over 30 cores representing more than 90% of the chip's greater than one-million gates. In addition to integrating the PowerPC CPU, transport function, video decoder, audio decoder, and peripherals, several analog functions including the DENC were brought on-chip.

The integration of three chips into a one-million-gate SOC provided significant cost advantage over the previous solution. The migration to 0.27um CMOS technology provided improvements in overall silicon density and a lower power dissipation.

IV. Methodology - Today's Issues

Large SOC's like the STB01000 provide function equivalent to a PC motherboard on a single chip. At this level of integration the chip design is no longer gated by the functions available as IP blocks, but by what can be verified at the system level in a given time-to-market window.

Using a gate-level simulator for detailed simulation of the entire system is too slow to allow for sufficient verification, so alternative methods were required. Simulation with an architectural C model of the entire chip provides the highest simulation performance, but this level of abstraction does not verify much of the chip implementation. Selecting the correct level of design abstraction (gates, RTL, behavioral, bus functional, and architectural models) for the job is crucial to maximizing productivity.

Choose the Right Level of Model for the Job

During STB01000 development, the chip was partitioned by subsystem and then by core. For each simulation test case, the core model type used was determined by the specific function(s) being tested. In this manner, the minimum required amount of chip logic was simulated to accomplish the verification goal.

If the details of a specific core's function were not needed, a dummy wrapper was used. If detailed core function was required, a cycle-accurate, RTL, or gate-level model was used. In all test cases, the top-level interconnect remained unchanged. The choice of appropriate model type was determined by the sub-system verification leader.

Simulation Technology Choices

Figure 2 illustrates the growing gap between the increasing demand for verification and the performance of today's simulation technologies. Event-driven simulation is adequate for small and some medium ASICs. Complex ASICs and SOC's require greater simulation performance to complete verification on a reasonable schedule. Options include cycle simulation, emulation and simulation hardware acceleration.

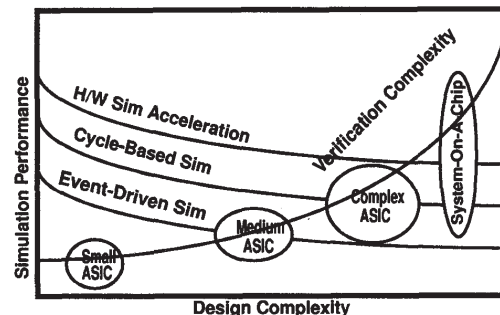


Figure 2 - Simulation performance vs complexity

The chip integration team overcame many difficult obstacles when attempting to use the higher performing simulation technologies for the STB01000 development. Several cores used an asynchronous design style which does not allow both cycle simulation and emulation on the original design source. Unlike event-driven simulators, the coding-style requirements of cycle simulators is fairly restrictive and models of these cores had to be rewritten. Clock synchronous synthesizable RTL is often the best choice for external interface models supporting an SOC verification. This ensures the highest degree of compatibility between simulation technologies.

Verifying the video graphics display and audio functions required simulation of worst-case transport-stream scenarios. Bus performance must be sufficient to prevent FIFO underruns that translate into disruptions in the audio and/or video data streams. Because the MPEG-2 A/V decoding application requires a huge number of simulation cycles to complete a small sequence of frames, emulation was used to verify the various transport-stream scenarios.

Table 2 identifies performance differences for the various levels of the STB01000 simulation model. Hardware emulation is 600 times faster than the RTL cycle

6.1.4

simulation, but it is expensive and only one unit was

Table 2: 1M Gate SOC Simulation Performance

Model and Simulation Technology	Cycles/second
Gates w/Delays: Event Driven	1.5
Gates w/o Delays: Event Driven	2.4
RTL: Event Driven	5
RTL: Cycle Simulator	27
Gates: H/W Sim Acceleration	16,000

available. The real ratio of simulation throughput is the cycles per second (cps) times the number of available licenses. Cycle simulation on a network of 100 workstations with 100 cycle-simulation licenses, yields 2700 cps, one sixth the performance of H/W simulation acceleration.

Hierarchical Simulation: Core, Subsystem and Chip

Efficient SOC integration requires a hierarchical approach to verification. Verification of cores being developed or undergoing modification should be separate from the chip-level development and verification. Breaking up the total verification requirement into hierarchical components maximizes efficiency and productivity. Even cycle-based simulation and emulation are not adequate if required to verify 100% of the functionality for every core and the chip integration of those cores. IP provided as reusable cores must be pre-verified with a high degree of quality to allow SOC verification to complete within any reasonable time frame.

Bus Functional Models (BFM) Communicate Protocol

Well-defined bus structures and BFMs were key to ensuring that a core-level simulation was accurate and representative of a chip environment. A clear and accurate bus specification is not enough to properly communicate bus protocol to design teams developing IP all over the world. Misinterpretation of the spec and mistakes in implementation are inevitable.

IBM provides a toolkit that supports a master model, slave model and protocol monitor for IBM's on-chip buses. The master and slave models are incorporated into both the core-simulation environment and the chip-simulation environments. The successful use of these models was a significant factor in the interoperability of the IBM Core+ASIC library.

Three Layers of Verification Coverage

The use of preverified cores allows chip integration verification to ensure that the cores properly intercommunicate, are properly interconnected, and provide needed system concurrency.

Interconnection & Intercommunication Testing

For intercommunication and interconnection testing, a set of tests that exercise 100% of the functional core I/O (excluding manufacturing test I/O, such as scan inputs and outputs) must be developed.

If a particular chip I/O is difficult to test because of required chip setup or simulation time length, interconnection testing is appropriate to obtain the desired coverage. The difference between intercommunication and interconnection testing is that interconnection testing uses a model of the core to capture/check inputs and to drive outputs of the core. The testbench has direct control of the model and the amount of simulation time and/or setup for the testing of this I/O is greatly reduced.

The required interface testing for a UART used in a SOC environment includes verification of the serial, DMA, OPB, interrupt and clocking interfaces.

The serial interface is an external chip interface that can be easily tested by wrapping the transmit data to the receive data. The modem-control I/O pairs can be tested in a similar fashion.

The interrupt output can be tested by ensuring that the received characters cause interrupts. The use of wrap-mode testing on the DMA interface for both transmitting characters and receiving characters can eliminate the requirement to write a behavioral model of the serial interface. The baud clock output and input can be tested by connection of the baud clock output to the general purpose timer input. The OPB interface testing can be checked by writing and reading registers within the UART core. Additionally, several signals such as the "error acknowledge" must be forced by reading a hole in the address map for the UART. All tests should be self-checking to allow regression tests to run automatically without requiring an engineer's inspection time.

Concurrency and Contention Testing

After the intercommunication and interconnection tests run successfully on the integrated cores, the system must be tested for its concurrency and contention. Two functions that operate independently may not operate concurrently due to problems with the sharing of common resources or limited bus bandwidth. Chips that share DMA channel resources or have significant pin multiplexing require concurrency testing. Applications, such as MPEG-2 decoding, with high memory bus bandwidth requirements, needed bus contention testing to ensure that all the defined system operational concurrency is supported. For the MPEG-2 decoding concurrency tests, worst-case (bandwidth) streams were decoded while DMA memory-to-memory testing and CPU code execution were running.

6.1.5

Interface Compliance

Because many STB01000 cores (e.g., IEEE-1284, Smart Card, I²C, NTSC/PAL) communicate through industry standard (official or defacto) interfaces, a considerable amount of compatibility testing was applied using standard compliance test suites. Functions such as AC-3 or Macrovision required certification at the chip level by certification labs. For interfaces where compliance test suites were not available, the test card supported multiple devices from various manufacturers, allowing additional test coverage for these interfaces. When licensing a core, it is important to understand whether the core has been certified, tested for compliance and which industry devices were used in the testing.

FPGA emulation can be used for the compliance testing of cores with lower speed interfaces prior to obtaining silicon. The Smart Card Interface core used in the STB01000 was put into a single FPGA and emulated, at speed, communicating to a PowerPC 403GC evaluation board expansion memory interface connector. Interrupts, DMA and memory-mapped I/O were all supported, allowing both interface compliance and full driver development to be performed prior to chip manufacture.

V. Planning for the future

Although core IP libraries have enabled reductions in development schedules, time-to-market will continue to remain the most critical design constraint. This forces minimization of specification, design, implementation, integration, verification, and simultaneous hardware and software development. To be successful, the tasks must be accomplished more efficiently in less time. This will require new advances in IP reuse, verification, and hardware-software co-design.

Chip Spec Generation from Core Specs

The job of generating a chip spec can be a very difficult and often underestimated task. Architects and designers often have very little time to support the chip spec. For chips that are primarily the sum of "n" cores, it is mistakenly thought that all the needed information is available and only a reformatting is necessary.

Core specs are written from the core perspective and do not always contain all the information that chip designers, core designers (developing a derivative) and software developers (writing drivers) need. Information on internal interface signals and timing is not necessary and should be removed from a chip spec. Information on register address maps should be combined to allow the programmer to look in one place to correlate the register address with the register.

With significant work, approximately 80% to 85% of the chip spec can be derived from the core specs. The

remaining required chip information, e.g., overview, system-level descriptions and the chip I/O definitions, constitutes only 15 to 20% of the spec.

Synthesizable Parameterized Soft Cores

Synthesizable parameterized soft cores can provide the exact number of supporting features for each unique chip without wasting silicon area; they are preferred over fixed function, non-parameterized soft cores. System features, such as the number of interrupts, DMA channels, timers, GPIO, address decode, and amount of on-chip memory, change with almost every chip and application. If the designer must edit each core to configure the correct quantity of these functions for each chip and reverify the changes by editing each suite of test cases, time-to-market will be impacted.

Parameterized soft cores should be preverified for all configurations or have a supported self-checking parameterized testbench. Care must be taken to ensure that the synthesized version matches the instantiated configuration of the RTL. Parameterized soft cores require considerable additional development and use of new coding styles for HDL, but the time saved is well worth the effort for chip developers. Without parameterization, design changes are required to eliminate unused core function which undermines the goal of a quick chip assembly.

Standardize On-Chip Buses

Third-party IP vendors will continue to proliferate. Systems will continue to evolve and integrate more complex functions. As design shops search farther abroad for IP to integrate, it will become increasingly more difficult to combine designs from different development methodologies. Designs are not only characterized by the HDL description language that they are written in, but also by the bus interfaces they are designed to work with. Today's SOC's are equivalent to multiple early single-bus SOC designs with bridges in place to communicate between different bus architectures (Figure 3).

Early SOC designs required standard on-chip bus architectures to effectively integrate embedded processors and peripherals. However, as design houses established protocols around different embedded processors, multiple standards emerged. This was acceptable if the IP chosen for a particular design existed with the correct on-chip interface. Now, as multiple bus structures and IP for those buses are being integrated into a single-chip solution, the need to find a single common on-chip bus architecture is greater than ever.

VSIA has begun the difficult task of resolving the differences between various on-chip bus architectures (6). Because industry members could not agree on a common bus, VSIA has proposed a hierarchical solution

6.1.6

that enables multiple buses to communicate through bus “wrappers” and bridges. The decision to bridge rather than merge disparate architectures can adversely affect bandwidth and latency, which are critical issues in system-level design performance.

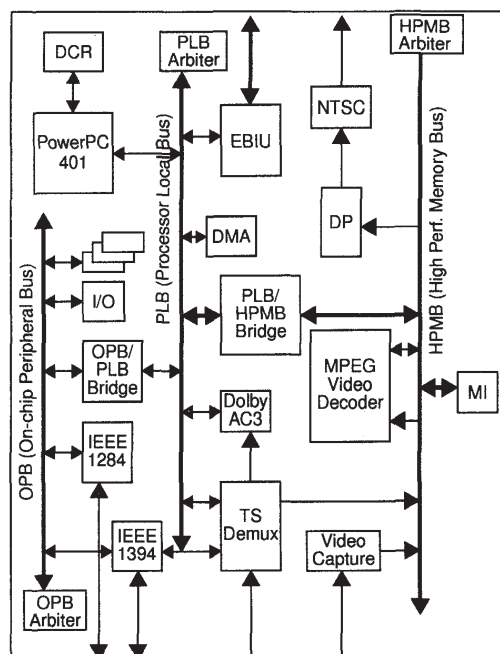


Figure 3 - Multiple Bus Architectures

Just as early SOC designs drove definitions of on-chip bus architectures, multi-SOC designs will force convergence of these protocols into a common, compatible standard. That standard should address register access, low-performance, and high-performance bus structures. Surprisingly, this convergence of on-chip bus architectures has already begun. Consider ARM's AMBA Advanced Peripheral Bus (APB) and IBM's On-chip Peripheral Bus (OPB). With a small amount of glue logic, an APB peripheral designed to interface to AMBA can interface with IBM's OPB without bridging to a second bus protocol.

When IP can be developed independent of the interface protocols required to use it in a particular system, the design becomes reusable. This frees the system architect to trade off different embedded microprocessors without being inhibited by the IP's bus compatibility.

Facilitate Verification

While resolving on-chip bus differences represents a large amount of work, verification remains the greatest obstacle to reducing time-to-market. Whereas it was

once thought that having a library of IP cores would enable larger, more complex designs and reduce time-to-market, SOC developers only realized the time-to-market gains. The SOC designs manufactured at IBM to date are actually 30%-40% smaller than the non-SOC designs (Table 1). It appears that the amount of function that can be verified within a given schedule is limiting the final size of the chip. Although the IP has already been designed, not all core configurations have been tested together. In addition, higher levels of integration bring former box and card issues, what used to be known as the system, into the chip environment.

As the amount of required verification continues to increase, it is necessary to streamline creation of the verification environment and improve simulation performance to achieve SOC time-to-market objectives. Accelerating the verification environment construction requires off-the-shelf test suites and pre-existing modular testbench components. Simulation performance will improve as event simulation will give way to cycle simulation and cycle simulation is extended by hardware-simulation acceleration.

New Definition of IP

Consider a new definition of intellectual property where the core and the components of the core verification environment are designed for reuse. Specifically, the simulation drivers that stimulate off-chip interfaces should be reusable and portable. An interface standard must be defined between simulation driver and testbench.

Architecting Superstructures

The first embodiment of this new intellectual property definition will be a verification method called superstructures. A superstructure includes a configuration of cores, a testbench to verify the configuration (including simulation drivers for off-chip interfaces), low-level device drivers to initialize and test individual cores, and a test operating system to apply tests to the configured system. This new approach to verification makes the simulation environment highly reusable.

The simulation drivers for a core's off-chip interfaces must be based on a standard driver-testbench interface to be portable between simulation environments. The simulation driver stimulates the interface according to its design protocol. With this work done in advance, the time required to build the verification environment is significantly reduced.

A low-level device driver (a sequence of instructions that initializes the core for operation and demonstrates functionality) must be provided for each core. It serves as a useful programming example and core-connection test and provides a good foundation for the more complicated system-level tests.

Finally, a test operating system (TOS) for the embedded processor is required to apply the low-level device drivers to the individual cores and allows randomization of events in the system, which represents more typical operating conditions. The TOS should be written in the C language to provide easy porting to other embedded processors.

With the superstructure established, integrating the system simulation environment is a minimal task. The system-level verification effort can emphasize the system-level tests. Time-to-market is significantly reduced when the effort required to build the system verification environment, test the interconnects, and demonstrate moderate functionality is minimized. Whereas today's schedules take six to nine months prior to release to physical design, superstructures will reduce the schedule duration to three to five months.

Higher Levels of Abstraction

A significant part of the time-to-market schedule that most silicon vendors do not control is the software-development phase of design. Typically, low-level software development starts with the simulation models delivered with the cores. However, because the simulation performance, at best, reaches hundreds of cycles per second, real software development is inhibited until hardware is available. To properly model a system earlier in the design cycle, higher levels of abstraction (Figure 4) are necessary to describe the core IP. The schedule can be reduced if the software can be co-developed along with the hardware.

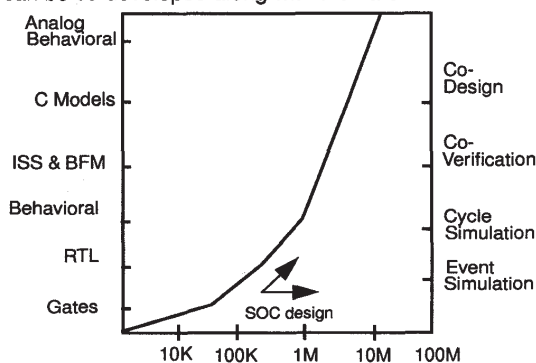


Figure 4 - Modeling Levels of Abstraction

With hardware-software co-design, the design point can be validated early in the schedule. Although most verification efforts are significant, verification only proves that the design operates as specified; it does not demonstrate that the specification is correct. High-level models can permit the system architect to appropriately partition the hardware and software tasks. With the correct tools in place, systems can be modeled, taking into account different embedded processors and on-chip bus architectures, and their performance trade-

offs. This can eliminate a second pass of hardware, often required when the initial design point chosen was incorrect.

Integrating Technologies

The definition of the "system" in "system on a chip" has expanded well beyond a processor, memory, and logic and will continue to do so beyond the year 2000. As the system expands to incorporate DRAM, flash memory and RF components (7), SOC's will not only drive IP integration, but also technology integration. Many ASIC vendors have already added mask steps to their logic process to build DRAM cores on-chip. High-speed analog and RF macros can be integrated with CMOS logic using silicon-germanium technology.

This new level of integration raises many questions. How many designs can bear the cost of integrated technology? Added features translate into added cost and, in many cases, an integrated solution may be more expensive than a multichip discrete approach. The power requirements in the digital world continue to decrease and will likely dip below the 1V level in the next five years. Can the higher voltage requirements of the RF analog circuits continue to function with decreasing digital power levels? Yet to be clearly defined are the markets and applications for whom the advantages of SOC integration outweigh the alternative multichip solutions (5). When this occurs, the new multi-technology SOC will again reinvent the tools and methodology.

References

1. B. Lewis, "ASIC Suppliers Target System-Level Integration," Dataquest, ASIC/SLI Worldwide, Market Trends, April 16, 1998.
2. A. Rincon, M. Trick, T. Guzowski, "A Proven Methodology for Designing One-Million-Gate ASICs," Custom Integrated Circuits Conference, pp. 45-52, 1996.
3. A. Rincon, C. Cherichetti, J. Monzel, D. Stauffer, M. Trick, "Core Design and System-on-a-Chip Integration," IEEE Design and Test of Computers, pp. 26-35, October-December 1997.
4. "Integrated Circuit Delay and Power Calculation System," IEEE Standard 1481 (Draft), Catalog #D55704, <http://standards.ieee.org/catalog/drafts.html>.
5. "The Revolution that is System-on-a-Chip," Semiconductor International, Cahners Publishing Co., p. 29, July 1998.
6. "VSI On-Chip Bus Attributes (OCB 1.1.0)," August 8, 1998, <http://www.vsi.org>.
7. D. Bursky, "The system on a chip: it's not just a dream anymore," Electronic Design, Vol. 45, No. 22, pp. 105-112, Oct 13, 1997.
8. J. Lipman, "Innovative processes bring systems on chips to life," Electronic Design News, Cahners Publishing Co, Vol. 43, No. 18, September 1, 1998, pp. 46-54.