BASICS of Design

ASIC TRADEOFFS

Dave Bursky, Digital ICs/DSP Editor

Select The Optimum ASIC Approach

When pushing the performance of a custom network processor or widening the bandwidth of a next-generation broadband interface, the performance of the customized chips used in the system banks heavily upon selecting the best implementation approach. CMOS FPGAs, structured/platform ASICs, and full custom ASICs provide a broad array of design options for all digital systems. Within the custom ASIC arena, designers now have the choice of a "standard" CMOS process, or for performance/power-critical designs, silicon-on-insulator (SOI) CMOS processes. New options also are appearing for designs at the leading edge. For example, strained silicon enhances electron mobility in the gate region, improving transistor switching speed.

Requirements of high speed for data networks and high frequency for broadband systems make the selection of the best ASIC approach critical for both performance and cost.

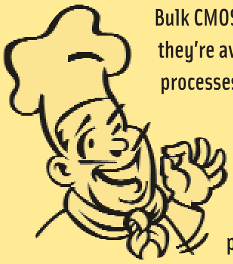
For the mixed-signal portion of the design, the choice of approaches is often limited to either a cell-based or full-custom approach because few off-the-shelf "platforms" include analog or mixed-signal functions, save for phase-locked loops or a few specialized I/O cells. But while the choice of approaches is smaller, the actual implementation scheme choice becomes more complicated. Technologies such as biCMOS, silicon germanium (SiGe), SOI, and shallow trench isolation can help push performance to the limit when standard CMOS won't fit the bill. Table 1 lists several technology options and provides some perspective on which technologies are best applied to different technical requirements.

Table 1: Process Technology Options

	Density	Performance	Noise	Geometries
Bulk CMOS	✔			0.18 µm –> 65 nm
SOI CMOS	✔	✔		0.18 µm –> 65 nm
SiGe biCMOS		✔	✔	At least a generation behind

ED Online 9077

Pick The Best Recipe



Bulk CMOS processes are the most common, and they're available from many ASIC suppliers. The processes are typically available in a wide range of feature sizes, from legacy 0.35-µm low-cost/low-performance processes with a few levels of aluminum interconnect to the latest 0.09-µm high-performance processes with nine or more layers of copper interconnect. Such processes support a wide range of applications, from very cost-sensitive products such as cell phones to high-performance processor-based products for networking and computing systems.

By placing a thin, insulating layer such as silicon dioxide between the active layer of silicon and the silicon (or other material) substrate, circuits can operate faster and/or consume less power because the insulating layer reduces parasitic capacitances that often cause performance losses as operating speeds go up. SOI can be a cost-effective option if power consumption or absolute, top-notch performance is a critical concern.

When analog performance is a key concern, biCMOS employing SiGe or silicon-germanium-carbon provides a mix of bipolar transistors for high-unity-gain frequency cutoffs with lower noise figures than possible with CMOS alone. Transistors also have better linearity than conventional bipolar transistors. Very often, the biCMOS processes are used to fabricate wireless networking and broadband connectivity products due to the multigigahertz operating frequencies of the RF transceiver. The low noise and high fT of the transistors is a key issue in such applications. In digital applications, the biCMOS structures can be used to implement high-speed transimpedance amplifiers or other transceiver circuitry.

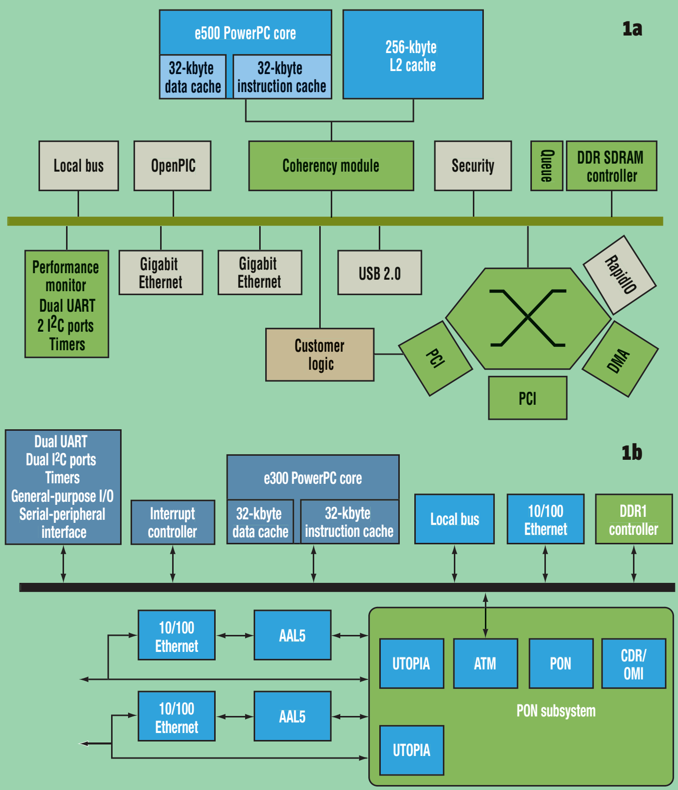
In networking applications, high-speed serial interfaces such as SERDES are often a popular building block. The application will determine the speed needed—1.25, 2.5, 3.125 Gbits/s, or still higher speeds are readily available—to achieve the desired I/O performance. Other specialized I/O for memory interfaces may be required to handle double-data-rate (DDR) memories, or well-defined clock trees on the chip may be critical to keep multiple subprocessors running in lock-step to process data at line rates of up to 10 Gbits/s.

Broadband applications have some different needs and often require analog signal processing. For instance, low-noise adaptive amplification will be needed to recover high-speed signals transmitted over long cables or phone lines. Additionally, circuits must also be able to implement equalization algorithms to pre-emphasize a signal prior to transmission and de-emphasize the signal on the receiving end to recover the data.

There Is No Typical System Architecture

In the realm of ASICs, the wide variety of designs necessary by designers really means that no typical chip exists. For instance, a high-performance network processor might incorporate a high-end PowerPC CPU core and cache subsystem, which is supported by a high-speed memory interface and a collection of high-speed peripheral interfaces. Those interfaces could include dual 1-Gbit/s Ethernet ports, a USB 2.0 serial port, and customer-defined logic. To minimize latency, various peripheral interfaces could be interconnected with an embedded crossbar—PCI or PCI-X host bus interfaces, Rapid I/O serial interfaces, a multichannel DMA controller, etc. *(Fig. 1a)*.

To serve the broadband market, a different mix of functions might better support applications such as passive optical networks (PON). In this case, a complete PON subsystem including the UTOPIA interface, the ATM transaction logic, and the related B-PON layer termination logic would have to be integrated along with a CPU, DDR memory controller, several 10/100 Ethernet ports, and other support functions *(Fig. 1b)*. This might require the combination of digital and analog functions on the same chip to deal with the optical network interface.



1. Two SoC designs, one targeted at high-performance network applications (a) and the other at PONs (b), show the different approaches to integrating resources.

Estimating Your Design Requirements

To determine the optimum technology, start by estimating the complexity of the design and the required operating speeds. The design may involve multiple "standard" cores such as a CPU, cache memories, Ethernet controller, and DRAM interface, as well as blocks of proprietary logic offering unique functions the "catalog" cores can't provide.

Using data for each block of IP provided by the ASIC vendor will enable a rough summary of the overall power consumption and chip gate count. Any custom block sizes and power could be estimated by approximating the gate count, specifying the operating frequency, and then computing the power by multiplying the number of gates by the power per gate/MHz. Most IP descriptions will also include the approximate area occupied by the cell for a particular process. Sum all the cells and approximate the area of the custom logic by dividing the estimated gate count by the number of gates per square millimeter possible in the selected process node.

In addition to estimating the total logic area, the number of I/O pins and power and ground pins are also important considerations. If the chip requires multiple large buses and lots of control signals, the number of I/Os and power and ground pins may ultimately determine the chip area if traditional pad rings are used around the perimeter of the chip. More advanced schemes use staggered pad rings or flip-chip assembly with contacts distributed across the entire surface of the chip, eliminating the pad ring altogether. Table 2 sums up some of the packaging options based on pin counts and power-dissipation requirements, from low-cost quad-sided flat plastic packages (QFP) to flip-chip ball-grid-array (BGA) units, while Table 3 looks at the range of I/Os and thermal capabilities for various BGA and flip-chip packaging options.

Plastic QFPs are among the lowest-cost, moderate pincount packages. They're available with or without an integrated metal heatsink. However, they're limited to only about 2 to 3 W. And for the same number of contacts, they actually occupy a larger board area than a BGA package. The BGA packages offer more power and ground connections than QFP options and offer higher I/O counts when the newer reduced-spacing BGA option is selected. The larger number of power and ground lines can help in noisy signaling environments by allowing better signal isolation and power distribution. Multiple package variations within the BGA category provide different degrees of I/O pin counts, frequency performance, and power-handling ability.

The process technology you select (180, 130, 90 nm, etc.) plays a key role in determining chip area, power, and speed. A chip with about 7 million gates (memory consumes about 4.5 million equivalent gates, so about 2.5 million gates of logic) occupies a chip of approximately 14 mm on a side when fabricated with 180-nm design rules. When the design rules are scaled to 130 nm, the same chip fits in an area just 10 mm on a side—less than half the area of the original design.

Table 2: Common Packaging Choices

	BGA (wire bond)	FCBGA	QFP
Number of I/O	Medium	High	Low
Power dissipation	Medium	High	Low
Signal integrity	Medium	High	Low
Cost	Medium cost	High cost	Low cost

In such a case, more than twice as many chips will fit on a wafer, and thus the cost per chip should drop even though the cost of masks and running the 130-nm process is higher than that of the 180-nm process. Alternatively, rather than make the chip smaller, more logic and memory can be integrated into the same 14- by 14-mm chip, integrating more of the system or enabling designers to add new, previously impossible functions.

Once a target process is selected, you can plug in the numbers and calculate the approximate chip area, speed, and power. Depending on performance requirements or power-consumption limits, you may also want to evaluate process options such as silicon-on-insulator to reduce losses due to parasitic capacitances or up the operation speeds by 10% to 20%.

CMOS works well in many mixed-signal applications. But in a few areas, biCMOS and SiGe technologies can deliver superior performance. Yet that performance often comes at a price. Both processes are typically at least one generation behind CMOS when it comes to process design rules, and that may limit the level of integration that can be achieved.

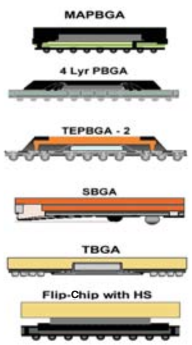


Table 3: BGA Package Variations

Status	Performance (MHz)	Thermal performance at 45°C T_A (W)	Package pitch (mm)	Ball count	Body size (mm)
Production-capable now	100 - 400	1.25 - 1.75	0.5 - 1.0	16 - 480	8 - 19
Production-capable now	100 - 400	2.25 - 2.5	1.0 - 1.27	144 - 900	17 - 40
Production-capable now	100 - 400	2.5 - 2.75	1.0 - 1.27	144 - 900	17 - 40
Production-capable now	100 - 1000	2.75 - 4.0	1.0 - 1.27	256 - 900	23 - 45
Production-capable now	25 - 125	2.75 - 4.5	1.0 - 1.27	276 - 1036	23 - 45
Production-capable now	1000 - 10,000	2.0 - 60.0	0.5 - 1.27	119 - 1500	16 - 35

Dissect Your Design To Integrate

When it actually comes to starting a design, a good look at the tool suites, libraries (both vendor-developed and third-party partners), design services, etc., that the ASIC vendor offers will usually provide a strong indication if the ASIC vendor is well suited to meet your needs. Does the vendor offer the cores or other logic functions necessary for your design? Are the design tools off-the-shelf tool suites provided by the ASIC vendor, or must you purchase the generic tools for installation on your workstations? Does the vendor use any propriety tools that may end up locking you in to its design flow? Is the IP from the vendor included as pa design, or are key blocks licensed in ently? But where do you actually sta the evaluation?

One good starting point is to examine your CPU, memory, and I/O requirements, then match them up against the ASIC vendor's offerings. In the CPU area, do you already have a particular CPU in mind, or just a word width or instruction set that you would like to use? Will the CPU need level 1 instru and data caches, a level 2 unified cac management, DMA control, and so forth? In the memory area, will on-chip SRAM fit your needs? Or will you need embedded DRAM or off-chip memory such as reduced-latency DRAMs or fast-column DRAMs for network applications, or standard DDR1 or DDR2 SDRAMs to achieve the desired memory transfer rates?

Within the chip, you must also determine the interconnect scheme used to tie together all the IP blocks and what type of test scheme to implement to ensure that the final chip can be tested sufficiently to prevent defective chips from getting into systems. Many interconnect options are available, with the simplest consisting of a basic on-chip parallel bus that all the IP blocks will connect to. There are many bus structures to select from—asynchronous or synchronous buses, specialized buses that maintain cache coherency, and crossbar interconnect schemes that provide nonblocking data movement among multiple blocks simultaneously. Bus selection may require a deeper analysis of your application to make sure you get maximum performance from your system.

Testing issues should be considered when you sketch out the initial design. Embedding test early in the design cycle ensures that the system-on-a-chip (SoC) will be testable when you get the silicon back from the ASIC vendor. But just how much test circuitry ude? Should you just include basic test structures and use automated pattern generation to test for stuck-, transition, and path-delay faults, or should you also incorporate built-in self-test features to offload testing, especially for memory blocks, on the large automated test systems? Also, if the SoC is memory intensive, hould you include redundant mem- rows and columns in the memory to patch memory blocks that might tive bit or two? Or should you incorporate some features such as parity checking or error-detection and correction (EDC) to prevent bad memory bits from corrupting the data?

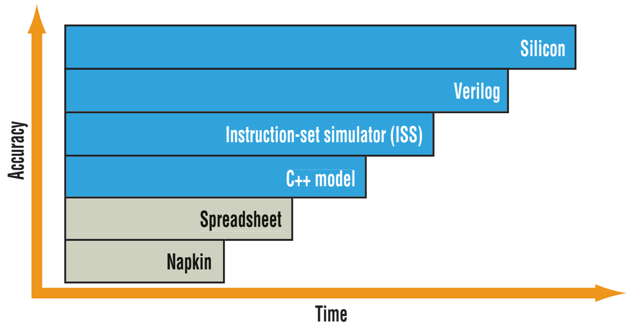
For I/O requirements, will you need standard interfaces such as 32- or 64-bit PCI, USB 1.1 or 2.0, 10/100 or 10/100/1000 Ethernet, DDR1 or DDR memory controller, or something a bit more up to date such as PCI Xpress, SPI 4.2, or multigigabit SERDES? By blocking out your CPU, memory, and I/O requirements, you can quickly scan the vendor IP libraries to ensure they have most of the blocks you need or locate partner companies that can supply blocks not directly offered by the ASIC vendor that will fabricate your design.

Who Does The Work?

When starting the design, sketching the functional diagram on a "napkin" usually is the first step toward visualizing the design. From there, it can quickly progress to a spreadsheet to calculate area and power, to C++ models to verify functionality, to an instruction-set simulator to verify the algorithms, and finally to Verilog or VHDL creation for full implementation and verification before releasing the design file to the ASIC manufacturer *(Fig. 2)*. Then the ASIC vendor might also perform the placement and routing of the design file and run one last set of design rule checks.

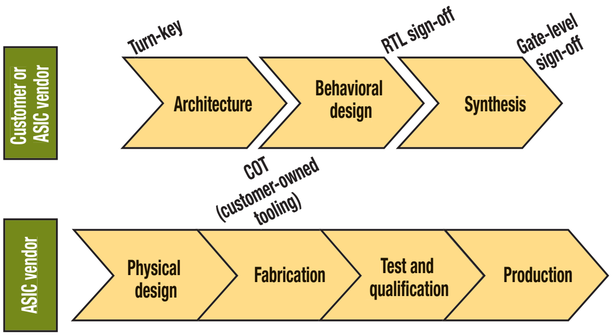
As you start the development cycle for an SoC, you can hand off portions of the design to the ASIC vendor at many levels. The handoff point depends on many factors, your own company's expertise, the availability of the design and analysis tools, the time-to-market demands, and still other factors *(Fig. 3)*. The four major points of handoff start with the easiest—handing the design over to the ASIC supplier after you have defined the architecture and letting the ASIC vendor do the full design. Or, if you have the design expertise, you can develop the architecture and perform the behavioral design to create the RTL description. At this point, it's also convenient to hand off the RTL file to the ASIC vendor. For larger designs, placement-based synthesis can aid in timing closure. Having physical design and synthesis co-located at the ASIC vendor may also be a motivation to hand off the project at the RTL level.

If you have synthesis tools available, you can go the next level and perform the RTL-to-logic synthesis, then hand off the gate-level description of the design. This is the most common handoff point for designers to transfer their design files to the ASIC vendor. It's also possible to take the design one more level and actually perform the physical design and layout of the chip, then hand over the physical design description to the ASIC vendor. This is often called customer-owned tooling (COT) because the customer has done all the work, down to the files that describe the masks used to fabricate the chips.



2. As a design progresses from a concept penciled on a napkin to final silicon, different tools must be used to provide ever more accurate analysis of the design.

3. When working with an ASIC vendor, several points in the design process provide natural transitions to transfer the responsibility for the next step or steps in the design process from your organization to the ASIC vendor's organization. The ASIC vendor can do a turn-key design once the architecture or functional specification is defined. You can hand off when the RTL definition is complete. After synthesis, you can perform a gate-level sign-off. Or if you have the resources, you can do the physical design and just hand off the data files that define the masks.



Once the design is done, the challenge of supporting the design begins. Because many of today's SoC designs employ embedded processors (and in many cases more than one processor core is embedded), there are a plethora of system and application software development and support issues to ensure that the SoC is used to its full potential. If the chip you create will be sold to other customers, it often requires support by run-time environments, drivers, stacks, operating systems, and the end application software. All of this must be created in conjunction with a full development environment that includes compilers, debuggers, emulators, logic analyzers, co-verification tools, and still other support such as evaluation boards.