HEARST
Electronic
PRODUCTS

Choosing chip technologies

BY STEVE BRIGHTMAN
Aeroflex Microelectronic Solutions
Colorado Springs, CO
http://www.aeroflex.com

The choice of chip technology for your design must be carefully researched and decided on early in the design cycle. To aid in this process, the following technology classifications must be carefully studied, and while no one solution is perfect, the best solution for your project must be selected based on the criteria specific to each technology.

SoCs

Systems-on-a-chip (SoCs) promise much to system designers and their companies. The ability to reduce complex systems to one highly integrated piece of silicon promises reduced component count, a smaller supplier base to manage, higher performance, reduced power consumption, smaller form factor, and reduced unit cost.

To accomplish this task, it is necessary to use an advanced process technology that allows combining some or all of the following capabilities: digital logic, analog functionality, and/or memory. Herein lies the major downside to this approach.

Development costs, in particular mask sets and IP licenses, can make it seem as if SoCs are in the realm of only an elite few. On the other side of the coin, in the case of a custom ASIC approach, long cycle times and the risk of a redesign can even threaten the market-opportunity window of the design.

Does this mean that, as some have predicted, the era of SoCs is over? Are we to resign ourselves to returning to a "kit-of-parts" multichip approach? No, not necessarily.

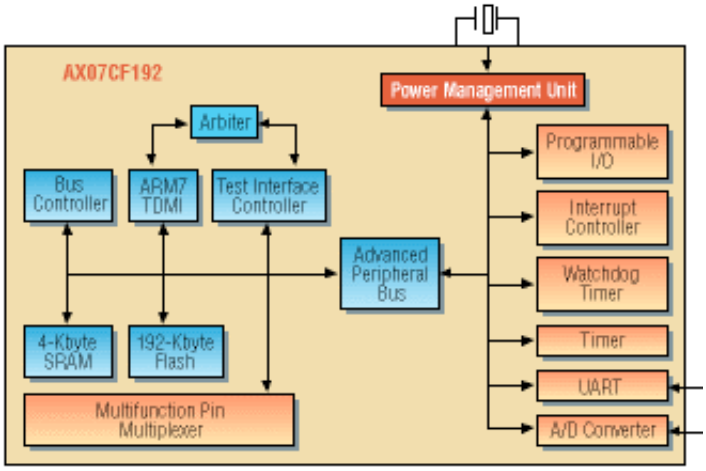
Although there are some who have declared that SoCs, and even ASICs, have had their day; reports of their demise have been somewhat exaggerated. There are several approaches to the implementation of an SoC, and we will review the key features and concerns associated with each. The technical demands of the

application, along with business considerations (for example, is the volume sufficient to amortize the development costs), dictate which approach best suits a given application.

ASIC

This approach offers the most flexibility in terms of optimization for the application need, and potentially offers the lowest unit cost, by virtue of being able to tailor the ASIC to exactly match the system needs--no more and no less. However, this approach can also be the most costly in the long run. A single mask set for today's advanced processes can easily cost in the region of $400,000, and this cost repeats if any sort of redesign becomes necessary.

Alternatively, if the system can be partitioned to use a less "exotic" process, these charges can be somewhat reduced. Add to this the necessary licenses for any IP that is used, and the total NRE charge can soon approach $1 million! Of course, for very high-volume applications, this cost can be amortized across the production run resulting in a less-significant average-unit-price increase. For lower volumes, this cost may be prohibitive.



ASICs such as the Aeroflex AX07CF192 combine an ARM7 core, embedded flash memory, a 5-channel A/D and a variety of other useful peripherals.

IP based vs. full custom

Many functions are already available as existing IP. In this case, the use of these functions offers a proven subsystem with little to no development effort or risk involved. However, if the desired function does not exist, or existing blocks do not offer the desired level of performance, some full-custom-design effort will be required.

Not only will this add to the development time (and cost), but there is also an additional risk the block will not function as desired and necessitate a redesign. Design tools exist to minimize this risk, but it is a fact that the tools are not keeping pace with the silicon capabilities, particularly on large complex designs.

When a redesign becomes necessary, not only does the development cost increase, but also the extra design cycle will impact time to market, typically by three months or more. If the application has a critical market window, this may result in failure

of the program.

Memory

Any microprocessor-based, off-the-shelf SoC solution must contain user-programmable memory. Although an OTP memory would fit the bill, flash has become the technology of choice. Flash memory can be easily rewritten during the development phase, allowing rapid design cycles and devices to be easily upgraded in the field with the latest software.

Devices are now available with flash memory densities of up to 2 Mbytes or more, but this also comes with a price. Not only does the larger memory consume more die area, it is slow for the manufacturer to test and/or verify, and both factors contribute to higher unit costs.

Flash memory must be erased and/or programmed in blocks (sectors), which may be too slow to accommodate all the system's needs for nonvolatile memory, in which case some quantity of EEPROM may also be required. Of course, SRAM is required for temporary storage of data.

FPGAs

The preceding comments have assumed implementation of "hard" IP functions. FPGAs offer an alternative method to implement a custom design by synthesizing the various circuit elements and programming them into standard silicon.

This has the virtue of avoiding the high mask set charges mentioned, but there are still charges associated with IP licensing and the synthesis itself. Other factors to consider would be the utilization factor of the available gates and the suitability for any analog capabilities required.

Off-the-shelf

Another alternative is to look for off-the-shelf devices that meet the application need. This approach has been taken by a number of manufacturers, primarily targeting high-volume applications such as cellular phones or PDAs. These devices, sometimes called ASSPs (application specific standard products) offer the designer a considerably reduced development cycle. They use proven silicon (so no redesign will be required), and they offer low unit cost.

One downside, however, is that no hardware differentiation is possible between multiple companies vying for the same applications. This can be offset somewhat by differentiation in the software implementation.

But what if your application is not of the multimillion unit variety? In the past, the only alternative would be to buy a kit of parts--microprocessor, memory, data acquisition, and other functional blocks as needed--hardly an SoC solution. Luckily it turns out that the same combination of functional blocks is often applicable to a variety of applications, and an increasing number of manufacturers are making such devices available.

As an example, look at Aeroflex's AX07CF192 (see *figure*). Based on an ARM7 core, the peripherals include a five-channel A/D

converter, a power management unit, multiple timers/PWMs, and up to 75 pins of configurable I/O. This feature set makes the device suitable for a variety of applications ranging from DVD players, to point-of-sale terminals, to handheld instrumentation.

Many devices are available from vendors using other popular cores, such as the ubiquitous 8051, and a variety of commonly used peripheral functions. If a device suitable for the application can be found, you will have a solution requiring no hardware development time or cost. The design cycle is limited solely by the time taken to write and debug the software. If time –to market is a key factor, this is certainly the best approach. The one downside is that, by definition, the chip always contains more functionality than is actually needed (if there was less than needed, then the device wouldn't work in the application). The incremental costs of this redundancy will in most cases be more than offset by the lack of (hardware) development costs.

Which factors are significant varies according to the application and will be the basis for the inevitable tradeoffs inherent in all designs. Often the distinctions between approaches can get blurred. For example, a "full-custom" approach may be nothing more than a simple design add-on to an existing "off-the-shelf" device. In such a case, the risks are small, but the increased development charges are not.

Decision time

The first step in the decision process is to determine which approach meets the technical requirements of the application. The second step is purely business: which approach falls within the budget in terms of development cost and unit cost. Finally, a risk assessment should be performed regarding the overall design cycle time: if a redesign is necessary, can you still achieve your market window? Regardless of the application, chances are there will be an SoC solution available to you, whether you need 10 pieces or 10 million.