

# PLD Capacity & 'Gate Counting'

By BRADLY FAWCETT ♦ Editor

Every user of programmable logic at some point faces the question, "How large a device will I require to fit my design?"



One of the most difficult problems facing those of us who make, market and sell programmable logic devices is establishing meaningful capacity metrics for our devices. We would like to supply potential users with metrics that provide an accurate indication of the

amount of logic that can be implemented within a given CPLD or FPGA device, and, at the same time, reflect the relative capacity of our devices compared to other manufacturers' competing products. Unfortunately, these two goals are often at odds with one another. In fact, a look at the density claims that I see in some PLD advertisements these days reminds me of the old saying that there are three types of falsehoods - lies, damned lies and statistics.

Of course, what is a problem for us is also a problem for potential programmable logic users, who must sift through the competing claims of the various vendors and try to select the most cost-effective solution for their application.

At the root of the problem is the multiplicity of available programmable logic architectures. Since each family of devices tends to have a unique architecture for its logic resources, direct comparisons, such as merely counting the number of available logic blocks or macrocells, are not always sufficient. (However, in some instances, direct comparisons of available resources probably could be used a bit more often than I think they are).

Most vendors, including Xilinx, describe device capacities in terms of "gate counts" — the number of 2-input NAND gates that would be required to implement the same function. This metric has the advantage of being familiar to ASIC designers and, in theory, allows the comparison of programmable logic device capacities to those of traditional, mask-programmed gate arrays.

But FPGA and CPLD devices do not consist of arrays of 2-input NAND gates; they have structures such as look-up tables, multiplexers and flip-flops for implementing logic functions. Thus, "counting gates" is far from an exact science; different vendors apply varying methodologies to determine gate counts.

All too often, gate counting becomes a game of "one-upmanship" among competing vendors. In fact, it seems to me that some marketeers decide what gate capacity they want to claim, and then design their metrics to reach that number. For example, in ancient times (that is, a couple of years ago), one of our competitors was bringing a large FPGA to market with a claimed capacity of 22,000 gates; the device had even been named to reflect that claim. Then Xilinx announced and started sampling the "25,000 gate" XC4025. Suddenly, the competing device somehow "grew" an additional 4,000 gates and was renamed and brought to market as a "26,000 gate" FPGA.

With the advent of FPGAs like the XC4000 Series that offer on-chip memory as well as logic resources, the issue is further complicated by the use of "memory gates" in addition to "logic gates". Each bit of memory counts as four gates. This factor can quickly inflate gate counts. For example, a single 4-input look-up table (LUT)

*Continued on page 5*

## XCell

Please direct all inquiries, comments and submissions to:

**Editor:** Bradly Fawcett

### Xilinx, Inc.

2100 Logic Drive

San Jose, CA 95124

**Phone:** 408-879-5097

**FAX:** 408-879-4676

**E-Mail:** brad.fawcett@xilinx.com

©1996 Xilinx Inc.  
All rights reserved.

*XCell is published quarterly for customers of Xilinx, Inc. Xilinx, the Xilinx logo and XACT are registered trademarks; all XC-designated products, XACTstep, LogiCore, Foundation Series, Alliance Series, NeoCAD, FastFLASH, and EZTag are trademarks; and "The Programmable Logic Company" is a service mark of Xilinx, Inc. All other trademarks are the property of their respective owners.*

can account for anywhere from one to nine “gates” when used to implement a logic function, depending on the type of function that is implemented. However, when that same look-up table is used as 16 bits of ROM or RAM memory, it now is implementing  $16 \times 4 = 64$  “equivalent gates.” Thus, FPGA gate counts rapidly inflate when the capacity metric assumes a significant amount of on-chip memory usage; it’s important to know what assumptions about memory usage apply to a claimed gate density in these devices.

For example, the device names in the XC4000E and XC4000EX families are based on our “maximum logic gates” metric — a measure of logic capacity only that assumes no memory usage. However, a major competitor has named its competing family based on a metric they call “typical gates” that includes both types of gates, and further assumes up to 35% on-chip memory utilization. Superficially, based on the device names alone, it appears that their devices are much larger than ours. However, even a cursory examination of device resources reveals a quite different story. For example, their “20,000-gate” device includes 1,152 4-input look-up tables, 1,344 registers and a maximum memory capacity of 12K bits. In contrast, the Xilinx “13,000-gate” XC4013E includes 1,152 4-input lookup tables, an additional 576 3-input lookup tables, 1,536 registers, and a maximum memory capacity of 18K bits.

This is not to say that Xilinx has not, at times, been guilty of “gate-inflation.” However, we have been more consistent over the years than most of our competitors. We recently reviewed and slightly revised our methodology for assigning gate counts for XC4000 Series and XC5200 family FPGAs. The results of that effort can be seen in the product specifications included in our latest data book. Xilinx application note #059,

“Gate Count Capacity Metrics for FPGAs,” explains our capacity metrics and the methodology used to obtain them; the application note can be viewed at our WebLINX web site ([www.xilinx.com](http://www.xilinx.com)).

Besides being subject to statistical manipulation, gate counts used as capacity metrics suffer from another severe drawback — they usually take only logic block and on-chip memory resources into account. Modern FPGAs include a host of other important features. For example, architectural features in the XC4000 Series that are not reflected in our capacity metrics include wide-edge decoders, dedicated arithmetic carry logic, registers and logic in the I/O blocks, global buffers and clock distribution networks, and internal three-state buffers. These important features can considerably boost the capacity and system integration capabilities of these devices.

Experienced CPLD and FPGA users realize that there can be considerable variation in the logic capacity of a given device dependent on factors such as how well the application’s logic functions match the target device’s architecture, the efficiency of the development tools, and the knowledge and skill of the designer. So, my advice is to apply a healthy dose of skepticism to CPLD and FPGA manufacturers’ gate count metrics (yes, even ours). Examine the assumptions behind the “gate counting methodology.” Better yet, take the time to examine and compare all the internal resources of the various devices being considered for a design. Fortunately, with a little experience, most designers can get a good intuitive feel for the logic capacity of the devices that they use. ♦

*Continued from page 2*

*“Take the time to examine and compare all the internal resources of the various devices being considered for a design.”*