WP112 (v1.0) February 23, 2000

Total Cost of Ownership: Xilinx FPGAs vs. Traditional ASIC Solutions

Author: Shelly Davis

Summary

This white paper provides an in-depth discussion of the technical, cost, and benefit trade-offs between a Xilinx programmable solution and an ASIC gate array, embedded array, or standard cell solution.

Overview

Until 1998, only ASIC vendors offered million-plus gate solutions, and standard cell technology offered the fastest, most complete silicon and libraries for emerging applications. Exponentially rising costs, as well as the difficulty of completing these complex designs, often make the full custom approach unrealistic for designs with time-to-market or engineering budget constraints.

Xilinx Virtex™ and Spartan™ FPGA families offer density and performance ranges that make them a cost-effective ASIC alternative. FPGAs are pre-engineered to offset many of the deep-submicron issues that complicate ASIC design. The FPGA design flow provides distinct alternatives that save designers time in the final verification loop and in the traditional ASIC foundry hand-off cycle. In addition, up-front development tools and non-recurring engineering (NRE) costs remain lowest in the industry.

Programmable logic is adaptable, flexible, and never has minimum volume requirements. Re-programmability is crucial in many of today's state-of-the-art datacom and telecom applications, and it can be a significant advantage in cases where systems are traditionally ASIC-based.

Now that this alternative is available, designers need to target the right technology choice for their own development programs, based on technological fit as well as the time-to-market and cost of "owning" that solution in production volumes.

The Cost of Ownership

Determining which solution best fits a specific requirement can be difficult, without the right tools and a model that allows objective comparisons. This white paper, and the accompanying ASIC Estimator program, allows users to model their own specific program and use tools to examine fixed, variable, and ownership costs. The true cost of ownership includes crucial items such as development NRE charges, but it also includes time-to-market and upgradability factors, which if not considered, can impact the profitability, life span, and overall success of the end product.

Figure 1 illustrates the essential differences in time-to-volume of an ASIC flow vs. an FPGA flow.

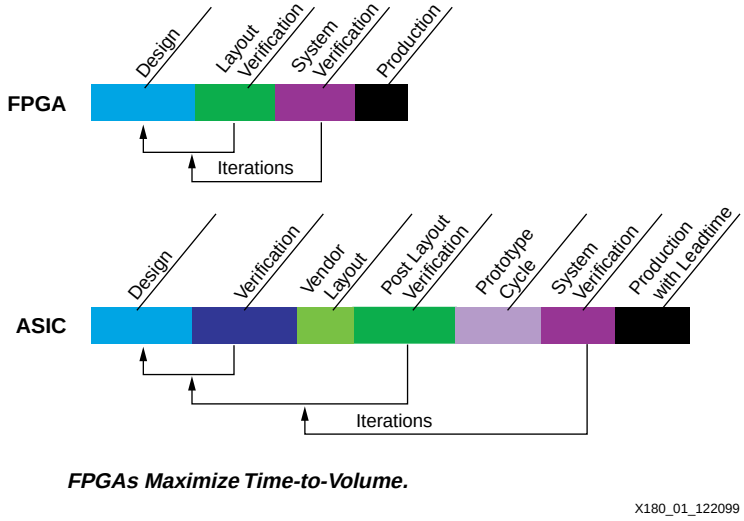

Figure 1: **Maximizing Time-to-Volume**

FPGA Performance Increases

Most designers recognize that one of the key benefits of programmable logic is flexibility. Enhancements and debugging take place in real time using actual silicon, rather than purely in a simulation environment as with an ASIC flow. Designs move from concept to working silicon quickly, without the agonizing wait for prototypes to be returned from the fab. But flexibility has a price as well. FPGA architectures carry "gate overhead" to gain programmability. This means that FPGA devices will generally have larger silicon areas than their standard cell counterparts, and that on a gates-only basis, there may be differences in performance. But the gap is narrowing. At 0.25 micron processes and below, available gates virtually become a non-issue, replaced by RAM, system-level design enabling features and I/O performance as the drivers. In these areas, programmable logic has reached parity with standard cell and embedded array solutions.

FPGAs Become More Affordable

The cost-of-ownership model changes too. The price of programmable logic has decreased five-fold in the last few years by taking advantage of deep-submicron process technology. FPGAs comptetitive range is broadening, even to the point of usurping the low-end gate array market. At the high end, Xilinx Virtex products can so enable a quick time-to-market that the solution is viable in production for extremely high-volume applications.

Remote Upgradability (IRL)

In the near future, networked systems will become the standard. Office equipment, satellite and communications networks are standard technology today. Any system that can be remotely accessed can also be remotely upgraded. As shrinking market windows, changing standards and quick product enhancement requirements all converge, it is critical to find a faster method of modifying hardware than fixed logic solutions allow. Each change or upgrade to a fixed logic (ASIC or ASSP) device requires a full re-design of the device, plus a complete product development effort. For today's deep sub-micron devices, this can be a 9 to 12 month lead-time.

Any system that contains Virtex FPGAs and can be remotely accessed, can be remotely upgraded. An entire network can be upgraded within moments, rather than through the lengthy, expensive and resource intensive effort of "Field Upgrading" fixed logic devices. Xilinx calls this "Internet Reconfigurable Logic" or IRL. Through the use of encrypted Java applets, bitstream data can be downloaded to any networked system, instantly upgrading the Virtex FPGAs in the system. The same concept can be applied to any system, internet enabled or simply

upgradable through a floppy disk carried to the system. As being used to upgrading software in this fashion, it is now possible to upgrade system hardware in the same manner.

IRL will enable sweeping changes in the way total costs and development efforts are determined. We will examine a potential IRL vs. fixed logic scenario in this study.

Making the FPGA or ASIC Choice

The Cost-of-Ownership Equation for Initial Development

Each logic development project has different goals and motivations. A standard formula or break-even analysis can ignore crucial factors beyond NRE and unit price. In today's highly competitive market, the first product to market establishes strong market share that is difficult to dislodge. End users demand increasing levels of customization, and solutions that can be quickly upgraded or enhanced will be the most cost-effective in the market. The cost-of-ownership equation weighs all the major factors that contribute to total cost and emphasizes the costs associated with product introduction slips.

Table 1 describes factors used to determine the total cost-of-ownership.

Table 1: **Total Cost of Ownership Considerations**

Development Considerations	Production Considerations
NRE	Unit price at various volume
Design tools	Minimum order quantities
Core/IP availability and cost	Lead-times
I/O standards	Future cost reduction potential
Engineering time from logic design through final verification	Fault coverage and reliability
Test development	Upgradability
Debugging tools	Inventory costs
Vendor support	Vendor support
Cost and time required for silicon re-spins	Expedite fees
Package technology	Risk
Device complexity and RAM capability	
Performance requirements	

Individually, each item can have a measurable impact on cost and schedule, and some of the factors are interdependent. For example, lengthening production lead-times can affect inventory, expediting fees, time-to-volume and the overall risk of the solution.

Often, many of these items can be overlooked in a technology evaluation. Sometimes they occur after the design cycle is complete, or they cut across several functional groups, such as engineering, purchasing and manufacturing, where a complete comprehension of the cost is difficult to assess. Some items are simply difficult to express using a dollar value, such as a potential test issue or the need for a silicon re-spin.

Demonstrating the Cost-of-Ownership Equation

In the following example, a Xilinx XCV1000E device in a ball grid array (BG560) package is used and is compared to an embedded array and a standard cell approach. The device attributes are described in Table 2.

Table 2: **Device Attributes Example**

Design Attributes	Virtex FPGA	Embedded Array	Standard Cell
Logic gates	500,000 system gates(1)	500,000	500,000
Package type	BG560	BG560	BG560
I/O standard	LVTTL	LVTTL	LVTTL
Voltage	2.5V core and I/O	2.5V core and I/O	2.5V core and I/O
RAM	1Mb Block RAM 20Kb distributed RAM	1Mb Block RAM 20Kb distributed RAM	1Mb Block RAM 20Kb distributed RAM
Core requirements	PCI Core	PCI Core	PCI Core
Test	Full JTAG	Full JTAG	Full JTAG
Development Considerations			
NRE	$0	$150,000	$290,000
Tools Maintenance	$20,000 $10,000	$60,000 $10,000	$60,000 $10,000
Engineering weeks in development	10 @ $5000 per week = $50,000	25 @ $5000 per week = $125,000	40 @ $5000 per week = $200,000
Weeks in design and synthesis	5	5	5
Weeks in verification	3	6	6
Weeks in test program development	3	8	12
Weeks in prototyping	0	4	7
Weeks in silicon verification	0	8	8
Weeks in production qualification	6	6	6
Re-spin NRE	0	$80,000	$200,000
Probability of re-spin	0	30% (industry average is 12% for manufacturer error and 20% for customer design changes)	30% (industry average is 12% for manufacturer error and 20% for customer design changes)
Product Considerations			
Pre-production unit price	N/A	$150 x 500 units = $75,000	$120 x 500 units = $60,000
Minimum order	N/A	10,000 units	20,000 units
Price at minimum order	$250	$46	$38
Volume forecast yr 1	20,000	20,000	20,000

Table 2: **Device Attributes Example (cont'd)**

Design Attributes	Virtex FPGA	Embedded Array	Standard Cell
Volume forecast yr 2	50,000	50,000	50,000
Volume forecast yr 3	80,000	80,000	80,000
End selling price of system	$8,000	$8,000	$8,000
Cost of producing system	$5,250	$5,000	$5,000
Life of product in months	48	48	48
Inventory carrying expense	5% per year	5% per year	5% per year

Notes:

1. Systems gates are equivalent to

Some of the numbers listed such as engineering time, NRE and re-spin data are typical industry averages as reported worldwide by market research firms such as Dataquest and Integrated Circuit Engineering (I.C.E). The end user's information and program attributes may be different from the example used.

Adding Up the Factors

The items in the above list that most dramatically affect the break-even decision are often the "intangibles". These are the items that do not show up on a purchase order and may not be accounted for financially, but can lead to great success or ruin a development effort. The intangibles can be measured however, and they should be included in any cost-of-ownership analysis. The basic breakeven formula is deceiving, because the "intangibles" are not encompassed.

The basic breakeven formula looks like this:

NRE + Engineering + (Unit Volume* Price) = breakeven

Any gate array or standard cell solution will require a much higher NRE and development cost than an FPGA solution. But, in most cases, the FPGA carries a higher unit cost than the equivalent ASIC device. Even though an FPGA has no NRE, the design cycles are short and the money spent on engineering is the lowest possible of all logic solutions, eventually the low-entry costs of the FPGA will converge with the lower price of the ASIC and there will be a cross-over point of cost effectiveness. If only the fixed costs are considered in this cross-over equation, and the simple break even formula (above) is used, you may reach a false conclusion, and choose a solution that will cost more money (and use more resources) over the life of the program.

Using the data from the previous example, the design alternatives would look like this:

The Standard Cell Approach

$290,000 + 200,000 + (20,000*$38) = 490,000 + 760,000 = $1,250,000

From this, it is straightforward to derive the cross over breakeven volume for an FPGA

$0 + $50,000 +(x*$250) = $1,250,000

or $250X = 1,200,000

= 4800 equivalent FPGA units.

Using this outcome, a designer would consider FPGAs for system development through 4,800 units. Beyond that, the Standard Cell approach becomes more cost effective.

The Embedded Array Approach

Using the same simple formula, the designer can do the analysis on an embedded array approach.

$$\$150,000 + \$125,000 + (10,000*\$46) = \$275,000 + \$460,000 = \$735,000$$

Solving for the Virtex FPGA equivalence:

$$\$0 + \$50,000 + (x*\$250) = \$735,000$$

$$\$250x = \$685,000$$

$$= 2740 \text{ equivalent FPGA units.}$$

If the analysis is stopped here, an embedded array technology that supports the product requirements would be the most likely choice. To do so without considering the cost-of-ownership denies the most important elements of the logic decision.

Breakeven With Time-to-Market

Missing a market window, or being late to market with a product because of a long development/debugging cycle can have a profoundly negative effect on the profitability of a product over its life. According to the Market Consulting Firm McKinsey and Co., late market entry has a larger effect on profits than development cost overruns or a product price that is too high. This is especially true in highly competitive markets, and those that have short market windows. Figure 2 shows the comparison.

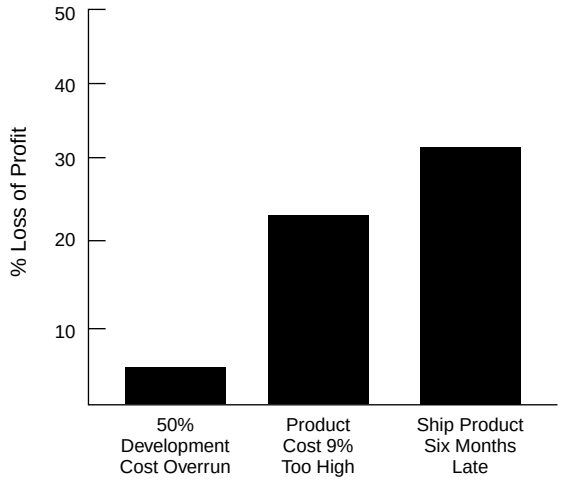

Figure 2: **Time-to-Market Cost**

Adding the lost time-to-market effects to the total cost of a project can be done using the data from the tables on the previous pages. Using a common market life-cycle model developed by Logic Automation (now owned by Synopsys), late market entry effects on profits are easy to demonstrate (Figure 3)

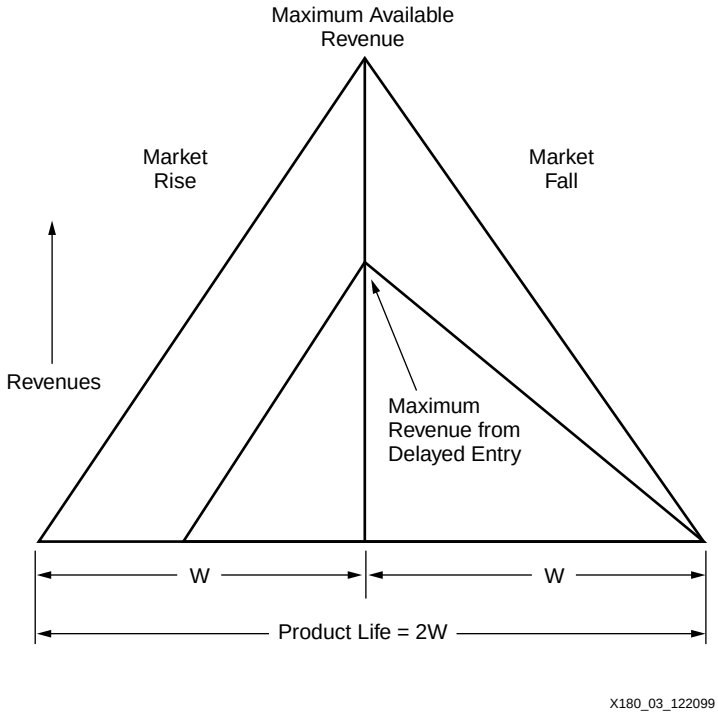

Figure 3: **Market Peak vs. Life-cycle**

This model assumes that the triangle represents total possible revenue from a single product life. It also assumes that the peak of the market is in the middle of the product life, and that the market peak in a delayed entry is still at the same point as an on-time entry. The results of the calculation show the percent of lost revenue from the total possible revenue (area of the large triangle less the area of the small triangle).

The formula is:

$$\text{Lost revenue} = (\text{Delay}(3W–\text{delay})/2W^2)(100\%)$$

Using the data from the previous example, the development time variables for the standard cell implementation add up to 44 weeks. This is common for a standard cell development effort.

The embedded array cycle saves some time by instantiating pre-diffused blocks, but still requires 41 weeks through the entire design cycle.

The FPGA development requires a total of 17 weeks. Time savings are due mostly to a shorter verification cycle and the lack of a prototype cycle prior to beginning in-system testing with silicon.

In this case, the FPGA solution provides the quickest time-to-market of the three alternatives, and the net time lost pursuing the standard cell solution would be the difference in the FPGA time to market and the standard cell time-to-market. In this case, it is difference of 44 weeks minus 17 weeks which is 27 weeks or almost seven months.

If the market for the product is three years or 36 months, the formula for the lost profit from pursuing a standard cell would be:

$$\text{Lost revenue formula} = (\text{Delay}(3W–\text{delay})/2W^2)(100\%)$$

$$\text{Lost revenue applied} = (7(3*18–7)/36^2)*100\% = 25.4\%$$

The conclusion is that if development starts at the beginning of the market window, time will be lost getting to market. Even if profit is generated at the same pace as if the product had been ready at the beginning of the window, time lost cannot be regained, and that lost profit must be added to the equation. Apply the loss percentage to the total profits and determine the potential dollar impact and the result would be:

Profit formula = (System price – system cost) * (Total production quantity)

Profit formula applied = ($8000 – $5000) * (20,000 + 50,000 + 80,000) = $450,000,000

Taking 25.4% from the profit is $114,300,000 in potential loss!

Applying the same formulas to the FPGA development, the equations are:

Lost revenue applied $(4.25(3*18–4.25)/36^2)(100\%) = 16.3\%$

Profit formula applied = ($8000 – $5250) * (20,000 + 50,000 + 80,000) = $412,500,000

Taking 16.3% from the total profit would be $67,237,500 in potential loss

Comparing the two outcomes, the standard cell could yield a potential of $335.7M for the life of the program, and the FPGA could yield $345.26M. Now the FPGA becomes more cost effective for the life of the planned program.

If the breakeven formula is applied and the potential loss is included from time-to-market delays for both alternatives, the result is surprising:

Standard Cell Breakeven Equation:

$290K NRE + $70K tools + $200K Eng + $114,860K Lost profit + (150,000 units * $38) = $120,560,000 in real cost and time to market profit loss.

Applying this to the FPGA equation, the breakeven units are derived.

FPGA Breakeven Equation:

$0 NRE + $30 Tools + $50K Eng + $67,317K Lost profit + (x units *$250) = $120,560,000

or

$250X = 53,242,500 = 212,970 FPGA units

The conclusion is that unless the program was forecasted to require over 200K FPGA units, then the FPGA alternative is most cost effective.

Additional Ownership Costs Add Up!

There are additional factors that may also affect the cost of ownership breakeven equation, such as:

- Inventory carry costs for custom product
- Future cost reduction potentials
- A change to the ASIC device once the prototypes have been committed
- Second sourcing considerations
- Cost of additional tools required to accelerate verification

Not only do these items add weeks to a development schedule, they also contribute additional NRE, engineering time and cost.

In the ASIC Estimator program, there are variables for inventory and for re-spin potentials. These factors can raise the breakeven points by thousands of units.

The Cost of Change is Dramatic

The future of logic design calls for super complex system-on-a-chip implementations to be completed ever faster, as networking, datacom and telecom continue to reinvent our communications infrastructure. These designs also have to be upgradable and changeable in

shorter cycles, as standards and features change. Unfortunately, the trend toward shorter product lives conflicts with the increasing difficulty of completing a fixed logic device. A fixed logic standard cell device becomes inordinately expensive to change as it progresses from simulation files through to silicon. After masks are committed, a change means a new NRE. After in-system testing and field trials, a change can require months of additional engineering resource as well as materials costs. When a device is deployed in a production system, a change often means a whole new product introduction, or in the worst case, it means a recall of a current system.

The ability of programmable logic to absorb changes at any stage of development is one of the reasons that the technology is growing in popularity in production applications. Now, the concept of virtual upgradability through Xilinx Online™, extends to possibility remotely upgrading the chip hardware to any connected system. Imagine adding a new feature to a fax machine, or upgrading a network hub in a remote area simply by downloading an encrypted bitstream to the device. The savings in product cycles, field maintenance and device costs shift the entire cost-of-ownership theory in favor of programmable, flexible solutions.

Conclusion

Looking at the entire scenario, you can plot the breakeven dollars and units of any ASIC vs. FPGA decision. Some decisions are simple; there is an overriding need for technology or features offered in one or the other technology that dictates the technology choice. But those are the exceptions. With programmable logic now capable of system level speeds, densities and enabled with state-of-the-art features, the decision often is made based on the most cost effective solution that meets the development schedule. Determining the fixed, variable and the cost-of-ownership costs for each development project can lead to some surprising results. Getting to market quickly and with a flexible solution increasingly outweighs the production price differential between ASIC and FPGA.

The Figure 4 illustrates how the breakeven point is a moving target.

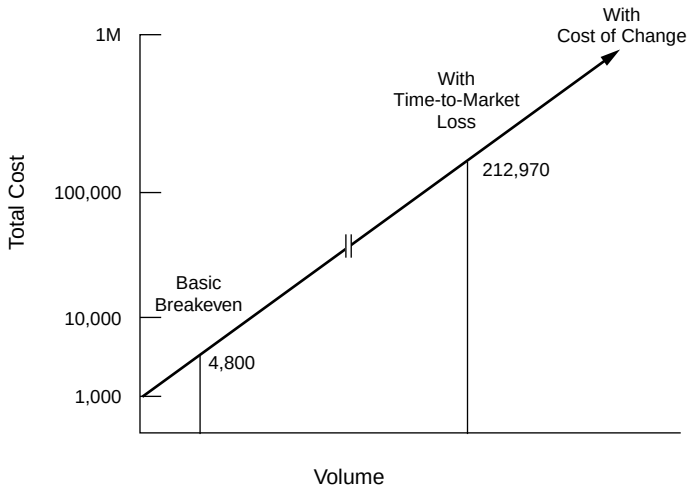

Figure 4: **Basic Breakeven Model**

Use this white paper and the online Xilinx ASIC Estimator to model specific requirements. It can help the designer avoid incomplete, non-quantitative design decisions. Long development cycles and the loss of flexibility add far more to the cost of an IC development

Revision History

Date	Version	Revision
2/23/00	1.0	Initial release.