



# The Total Cost of Ownership Xilinx FPGAs vs. Traditional ASIC Solutions

September 1994

White Paper

## Overview

Programmable logic devices, particularly FPGAs, continue to gain momentum over traditional ASICs as the logic solution of choice for today's system designs. A primary concern in the selection of PLDs, however, is their ability to function as a high-volume production solution. Historically, FPGA devices have been viewed as appropriate for prototyping or low-volume production, while mask-programmed gate arrays were viewed as more cost-effective for high-volume applications. The difference in unit price typically dictated that decision.

But today's design decisions are far more complex than simply weighing unit price and NRE to determine which technology to use. By analyzing all of the variables that affect an I.C. development program using breakeven formulas and time-to-market models, FPGA solutions prove extremely cost effective at surprisingly high production volumes.

## The "Cost of Ownership" Equation

Custom gate arrays and similar ASIC products have many hidden costs which inflate the "cost of ownership". Unit price and NRE become a small part of the equation when the expense involved in giving up flexibility and time-to-market is considered. By applying simple formulas to the specifications of a development program, breakeven points and total costs can be easily derived.

Factors used to determine this "cost-of-ownership" equation can include:

Unit Price/Volume	Package Type
NRE	Design Time
Production Lead-times	Re-spin Potential
Gate Count	Expedite fees
Tools	Engineering Time
Time to Volume	Fault Coverage
Future Cost Reductions	Quality
Technology	Vendor Service
Test Development	Inventory
Flexibility	Risk

Alone, each of these items have a measurable impact on cost and schedules. Some are interdependent. For example, lengthening production lead-time can affect inventory, time-to-volume, expediting fees and risk. Many of these costs can go unrecognized in a traditional cost analysis, because they occur after the design cycle, or cut across many functional groups such as purchasing, engineering, and quality assurance. Sometimes it is simply because there is not an easy way to express the value of a "potential problem".

This analysis will use a typical 10,000 gate device to demonstrate the concepts and formulas for finding the breakeven points and cost-of-ownership for an FPGA versus an ASIC device. The same formula can be used for any application or situation.

## FPGA Versus Gate Array Technology

FPGAs carry all the benefits of standard products: no NRE, quick delivery and no inventory risk, with fully-tested functionality. In addition, users can verify a design in-circuit and make changes at any time. By comparison, a gate array or custom ASIC solution has a long, rigorous development cycle, with the risk and inflexibility of non-programmability, but offers the lowest unit cost.

Programmable FPGA	Gate Array
Standard Product	Custom Product
Off-The-Shelf Delivery	Months to Manufacture
Fast Time-to-Market	Slow Time-to-Market
User Programmable	Non-user Programmable
No NRE	NRE, Expedite Charges
No Inventory Risk	Customer-Specific
Fully Factory-Tested	User Test Development
Simulation Helpful	Simulation Critical
In-Circuit Design	No In-Circuit Design
Verification	Verification

### Demonstrating the Real Cost of Ownership

#### Adding up the Factors

Most factors that dramatically affect the total cost-of-ownership equation are considered “intangibles”; that is, they don’t show up on a purchase order and are not accounted for in a financial sense. These intangibles are however, measurable and should be included in any ASIC design decision.

The following page lists attributes and cost information needed for the cost-of-ownership equation. By filling in the blanks with costs for a specific project, this analysis can be tailored to any situation. The numbers used in this example are shown in parentheses. These represent typical numbers based on worldwide research by Dataquest and Integrated Circuits Engineering (I.C.E.) a market research firm. Your information and program attributes may be different from the example used.

---

#### Design Attributes

- |   |                 |
|---|-----------------|
| 1) Number of gates in the design  | _____ (10,000)  |
| 2) Package type   | _____ (PLCC)    |
| 3) Selling price of end system  | _____ (\$2,000) |
| 4) Cost of end system   | _____ (\$1,100) |
| 5) Product life in months   | _____ (36)      |
| 6) Engineering \$\$/per person per week<br>(Industry standard is \$3,000) | _____ (\$3,000) |

#### Product Forecast

(total volume of this device used in the product)

- |            |                      |
|------------|----------------------|
| 7) Year 1  | _____ (5,000 units)  |
| 8) Year 2  | _____ (12,000 units) |
| 9) Year 3  | _____ (5,000 units)  |
| 10) Year 4 | _____                |
| 11) Year 5 | _____                |

#### FPGA Attributes

- |                                |                  |
|--------------------------------|------------------|
| 12) NRE                        | _____ (\$0)      |
| 13) Tools                      | _____ (\$10,000) |
| 14) Average price              | _____ (\$39)     |
| 19) Training in weeks          | _____ (1 Week)   |
| 20) Engineering labor in weeks | _____ (4 weeks)  |
| 21) Qualification in weeks     | _____ (3 weeks)  |
| 22) Production lead-time       | _____ (2 weeks)  |

#### ASIC Attributes

- |  |                  |
|--|------------------|
| 23) NRE  | _____ (\$25,000) |
| 24) Tools  | _____ (\$10,000) |
| 25) Average price  | _____ (\$13)     |
| 26) Training in weeks  | _____ (2 weeks)  |
| 27) Design capture in weeks  | _____ (3 weeks)  |
| 28) Simulation in weeks  | _____ (2 weeks)  |
| 29) Test vector development in weeks   | _____ (6 weeks)  |
| 30) Place and route in weeks   | _____ (1 week)   |
| 31) Back annotation/DRCs in weeks  | _____ (1 week)   |
| 32) Final simulation in weeks  | _____ (1 week)   |
| 33) Proto cycle time in weeks  | _____ (2 weeks)  |
| 34) Qualification in weeks   | _____ (5 weeks)  |
| 35) Production lead-time<br>(Industry average for gate arrays is 8 to 14 weeks)  | _____ (9 weeks)  |
| 36) Potential for a re-spin expressed<br>as a percentage of probability<br>(Industry average is 12% due to manufacturer error and 20% due to customer design changes.) | _____ (30%)      |

## Basic Breakeven Is Deceiving

The FPGA versus ASIC decision must include more than the basic price x volume breakeven to realistically model the true cost-of-ownership. By far, the biggest impact on the cost-of-ownership is time-to-market, not unit price and NRE. This discussion will use the basic breakeven model described below, adjusted for time-to-market and other crucial factors.

The basic formula looks like this:

$$\text{Basic Breakeven} = \text{NRE} + \text{Engineering} + (\text{Units} \times \text{Price})$$

Any gate array or ASIC solution will require a much higher development cost than an FPGA solution. While FPGA solutions have traditionally, due to device price, been used in lower volume applications, that trend has reversed itself. An FPGA has no NRE, design cycles are short and the money spent on engineering is low. However, if only the fixed costs of a stable design are considered in a silicon solution, eventually the low entry-costs of an FPGA converge with the unit price of an ASIC.

Using the data from the previous page, typical scenario might be as follows:

$$\begin{aligned} \text{ASIC} &= \\ &\$25,000 \text{ NRE} + \$79,000 \text{ Engineering and tools} + (X \text{ units} \times \$13.00) \\ \text{FPGA} &= \\ &0 \text{ NRE} + \$25,000 \text{ Engineering and tools} + (X \text{ units} \times \$39.00) \end{aligned}$$

Solving for the Breakeven:

$$\begin{aligned} 1) \text{ ASIC} &= 25 + 79 + (13x) \\ \text{FPGA} &= 25 + (39x) \\ 2) \text{ ASIC} &= 104 + (13x) \\ \text{FPGA} &= 25 + (39x) \\ 3) \text{ Total} &= 79 + 26x \\ 4) \text{ Total} &= \frac{79}{26} + \frac{26x}{26} = 3,038 \text{ units} \end{aligned}$$

Using the equation above, the curves show that when only the fixed costs of NRE, engineering dollars and unit price are taken into consideration, the breakeven point between an ASIC solution and an FPGA solution is at 3,038 units. If the total requirement for the above example is not more than 3,038 units over the life of the program, the decision seems simple: the FPGA is a more cost-effective solution to 3,038 units. One critical factor that basic breakeven does not consider is that typically, the prototype-to-production cycle can last for months. (Dataquest estimates that in North America, the average prototype-to-production time is 7 months). During that time, hundreds of pre-production units can be used.

**To stop here with a cost of ownership analysis can easily lead to a false conclusion. The unit price and NRE associated with each option are a small portion of the true costs. Time-to-market considerations make the most difference in total cost.**

## Time-to-Market Cost

Missing a market window, or being late to market with a product because of a long development/debug cycle can have a profoundly negative effect on the profitability of the product over its life. According to the respected consulting firm McKinsey and Co., late market entry has a larger effect on profits than development or product cost overruns. This is especially true in very competitive markets. Figure 1 shows the comparison.

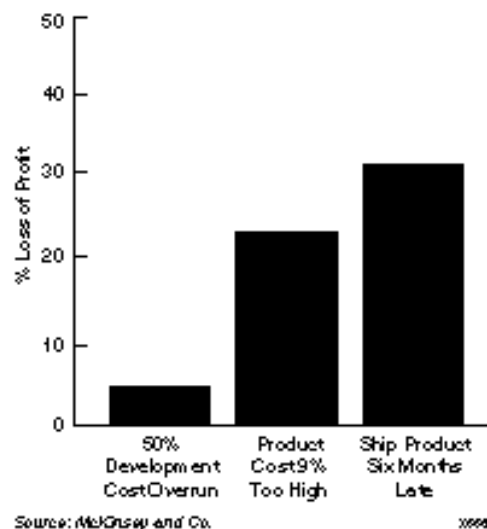


Figure 1.

Adding the effect of time-to-market loss to the total cost-of-ownership equation can be accomplished by using data points from the data filled in on page 2. The triangular model in figure 2, developed by Logic Automation (now owned by Synopsys), reflects the effects of being late to market. It assumes that a market peak is in the middle of the total lifecycle and that the market peak in a delayed market is at the same point as an on-time peak. 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).

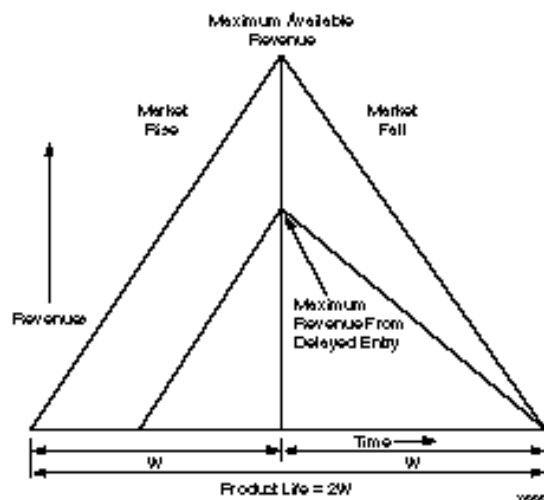


Figure 2.

## The Total Cost of Ownership

The percentage of lost revenue is calculated as:

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

Using the data from page 3, the delay consists of the sum of items #26 through #36. The variables look like this:

26) Training in weeks	= 2
27) Capture	= 3
28) Simulation	= 2
29) Test Vector Development	= 6
30) Place and Route	= 1
31) Back Annotation	= 1
32) Final Simulation	= 1
33) Proto Cycle	= 2
34) Qualification	= 5
35) Production Lead-time	= 9

Total Development Weeks 32 (8 months to production)

The same variables for an FPGA would be as follows:

30) Training in weeks	= 1
31) Capture	= 2
32) Simulation	= 2
33) Test Vector Development	= 0
34) Place and Route	= 1
35) Back Annotation	= 0
36) Final Simulation	= 0
37) Proto Cycle	= 0
38) Qualification	= 3
39) Production Lead-time	= 2

Total Development Weeks 11 (2.75 months to production)

Assuming that the FPGA solution provides the quickest time-to-market of any alternative, the net time lost in pursuing the ASIC would be the difference between the FPGA time-to-market and the ASIC time-to-market. In this example, it would be 8 months minus 2.75 months = 5.25 months.

If the market for the product is three years or 36 months, the formula for a gate array would be:

$$\text{Lost Revenue} = (5.25(3 \times 18 - 5.25) / 36^2) \times 100 = 19.75\%$$

Conclusion: if development starts at the beginning of the market window, time will be lost in 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 loss of profit must be added to the equation. Using #3, #4, #5 and #7 through #11 of the design attributes and product forecast, the total cost equation now contains the time-to-market factors:

Selling Price of End System minus cost of system \* volume per year = net profit

$$\begin{aligned} \text{Net Profit} &= \$2000 - \$1100 \times (5,000 + 12,000 + 5,000) \\ &= \$19,800,000 \text{ life of program} \end{aligned}$$

If the percent of lost profit is as much as 19.75%, the gate array costs an additional **\$3,910,500**.

Inserting the time-to-market cost into the equation from page 3, the total cost equations become:

$$\begin{aligned} \text{ASIC total cost} &= 25K + 79K + 3,910K + (X \text{ units} \times 13) \\ \text{FPGA total cost} &= 25K + (X \text{ units} \times 39) \end{aligned}$$

**The breakeven point moves out to: 153,423 units**

### Additional Re-spin Cost

According to I.C.E.'s ASIC Outlook 1994 book, average failure rates for "first time designs" was only 12%. The real pitfall with design iterations is not that silicon doesn't work, it is that most designs change several times prior to production. Designers change logic or fix bugs and spin the device again. These iterations take time, and require going back to design capture if being developed in a gate array. *This substantially raises the breakeven point of a gate array versus an FPGA.*

A current industry-standard for average re-spin potential on gate arrays is 30%. Using this number, we can calculate the cost of a re-spin and add it to the equation:

Item	Time
NRE	
Design Recapture	1 week
Design Resimulate	1 week
Test Vector Development	2 week
Place and Route	0.5 week
Back Annotation	0.5 week
Final Simulation	1 week
Proto Cycle in weeks	2 weeks
Production Lead-time	9 weeks
Total Re-spin Time	17 weeks

The calculation is:

$$\text{Re-spin cost} = (17 \text{ weeks} \times \text{Engineering dollars per week} (\$3,000) + \text{NRE}) \times \text{re-spin potential}$$

$$\text{Re-spin cost} = ((17 \times 3,000) + 25,000) \times .3 = \$22,800$$

There is no manufacturing re-spin time for an FPGA, so the FPGA total cost equation does not change. The Gate Array Equation now looks like:

$$\text{ASIC total cost} = 25K + 79K + 3,910K + 22,800 + (X \text{ units} \times 13)$$

**The Breakeven point moves to: 154,300 units**

Accounting for the 17 week additional time slip while the re-spin is recaptured, simulated, prototyped and approved can be estimated using the lost revenue formula. **Lost revenue becomes 32.6% or \$6,454,800 of the total revenue instead of the \$3,910,500 before the re-spin. Breakeven now becomes 252,176 units.**

## Inventory Considerations

Stocking inventory is necessary with a custom part because of long lead times, single sources, upswings in demand and end-of-life buys. A generally-accepted standard for measuring the cost of inventory is to use the current interest rate. For this example, the rate will be 8%. Adding that into the equation, it becomes:

$$\text{Inventory calculation} = .08 * \text{units} * \text{price} = (.08 * 22,000 * 13) = \$22,880$$

$$\text{ASIC total cost} = 25K + 79K + 6,454.8K + 22,800 + 22,880 + (x \text{ units} * 13)$$

The FPGA does not have inventory carrying costs associated with it because it is a standard product with a very short production lead-time and is available from distribution. The FPGA equation does not change. **The breakeven becomes 253,056 units.**

Many users of gate arrays require a second source because custom product from a single source can be risky. FPGAs are stocked and available from distributors, alleviating much of this problem. Bringing on an ASIC second source is almost as expensive as developing the first source. The equation in this case could become:

$$\text{ASIC total cost} = 25K + 79K + 6,454.8K + 22,800 + 22,880 + 64,500 + (x \text{ units} * 13)$$

Assuming that a second source costs \$25K NRE and half the original engineering at \$39.5K, the new **breakeven point would be 255,537 units.**

## Effects of Price Reductions on Breakeven

Most FPGA and gate array pricing varies over time and volume. It is not possible for the purposes of this example to include all variables into the equations, but a drop in pricing can have a substantial impact. FPGA pricing has historically decreased 25% per year while gate array pricing has decreased about 5% per year. Including these variables into the equation changes the breakeven point yet again.

$$\text{FPGA Price Reduction} = \$39.00 * .80 = 31.00$$

$$\text{ASIC Price Reduction} = \$13.00 * .95 = 12.35$$

**This price decrease moves the breakeven point to 356,245 units, because the price decrease on the FPGA is greater than the decrease of the gate array.**

## Combining all the benefits of FPGAs with the low unit price of an "ASIC" solution

The ultimate system solution for most designers would be one that provides the quickest time-to-market and most flexibility during the development and early production stages, while offering a cost-reduction path for high volume, all with a minimum of effort. The Xilinx HardWire™ LCA product family of masked gate arrays can achieve those goals. Using the .LCA files created during FPGA development, Xilinx can convert the production-worthy FPGA into a mask-programmed version of the exact FPGA design. In

most cases, the HardWire device is priced as much as 60% less than the corresponding FPGA. Xilinx fully guarantees that the HardWire device will function exactly as the FPGA in the designer's system. All HardWire devices include 100% fault coverage at no charge and there are no additional engineering resources that need to be invested in making the HardWire conversion.

The HardWire LCA conversion methodology is superior to any other cost-reduction option:

- no additional engineering resource
- no recapture, simulation or redesign is required
- preserves all nets, CLBs, relative placement and routing
- 100% fault coverage is standard
- production facilities and packaging used for FPGAs and HardWire devices are identical.

The effects of "cutting-in" a HardWire LCA during the volume production phase of a program has a dramatic effect on the cost-of-ownership equation. In almost all cases, the combined time-to-market power of the FPGA and the redesign-free cost advantages of HardWire LCA are far more cost effective than any other traditional "ASIC" or "FPGA to ASIC" program.

An example (taken from the data on page 3):

$$\begin{aligned} \text{FPGA price} &= \$39.00 \\ \text{HardWire price} &= \$18.00 \\ \text{HardWire NRE} &= \$18K \\ &(\text{no other expenses are incurred}) \end{aligned}$$

Assume a conversion to HardWire beginning in year two, with a forecast of 5,000 units year one, 12,000 units in year two and 5,000 units in year three. The average price of the units works out to be \$22.77.

$$\text{Average FPGA/HardWire price} = ((5K \text{ at } \$39.00 + 17K \text{ at } \$18.00) / 22K)$$

Total cost-of-ownership equation for the FPGA combined with HardWire is:

$$\begin{aligned} \text{FPGA} + \text{HardWire total cost} &= 25K + 18K + (\text{units} * 22.77) \\ \text{ASIC total cost} &= 25K + 79K + 6,454.8K + 22,800 + 22,880 + 64,500 + (x \text{ units} * 13.00) \end{aligned}$$

**The breakeven point becomes 678,196 units**

## Conclusions on Breakeven Analysis

Working through a realistic example like the one above leads to some startling conclusions. By far, the greatest impact on a development program is the time-to-market costs, rather than the unit price or NRE. Time-to-market accounts for 90% of the total cost picture. Combining the FPGA with the HardWire LCA for volume moves the breakeven point into high volume areas.

Other considerations, such as inventory and cost of second sourcing, add to the total cost of an ASIC program, but to a lesser extent than time-to-market or design flexibility.

Choosing FPGAs as a design solution reduces the risks (and thereby the costs) associated with a non-programmable methodology. Redesign potential, second sourcing, expedite fees etc. are virtually eliminated. Combining the FPGA with HardWire LCA cost-reduced devices moves the breakeven point as compared to traditional ASICs or third-party redesign out so far that it virtually eliminates the need for an ASIC program in many cases.

The following figures summarize and contrast the entire scenario. Deriving a total cost through looking only at unit price times volume (figure 3a) can be very deceiving. In the example used in this paper, the ASIC solution would look less expensive than the FPGA over the life of the program. It does not encompass the real and inevitable factors that make the ASIC program so much more expensive. Figure 3b provides a look at the real cost-of-ownership. When time-to-market, re-spins, inventory costs, second-sourcing and price reductions are included in the equation, the FPGA plus HardWire solution is less expensive over the life of the program by 92%.

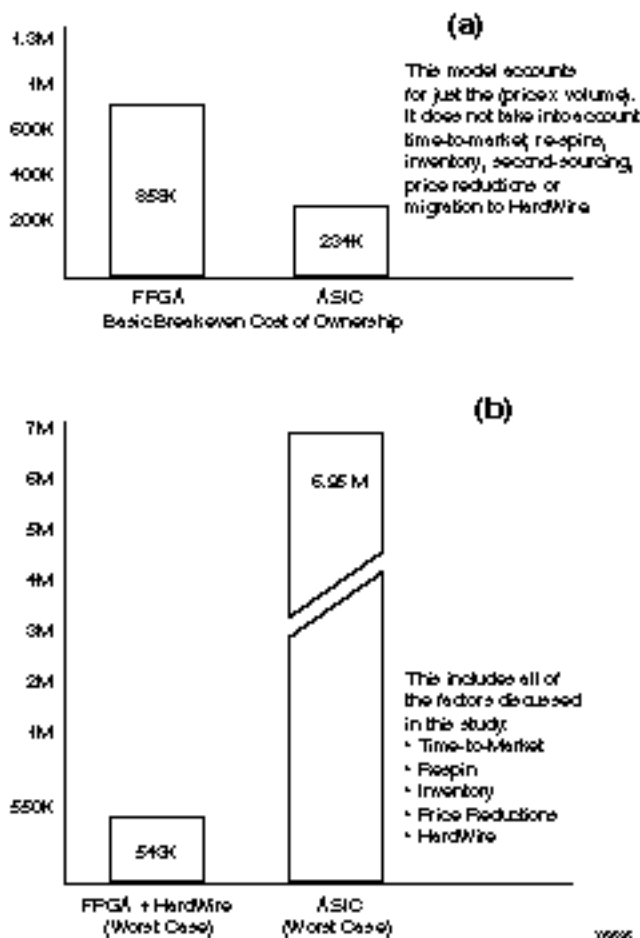


Figure 3. (a) Basic Breakeven Model  
(b) Cumulative Breakeven Points

Looking at the cumulative breakeven points in figure 4, it is evident that a combination of the FPGA's flexibility plus the low cost of a HardWire LCA makes the Xilinx solution more cost effective for programs of substantial volume.

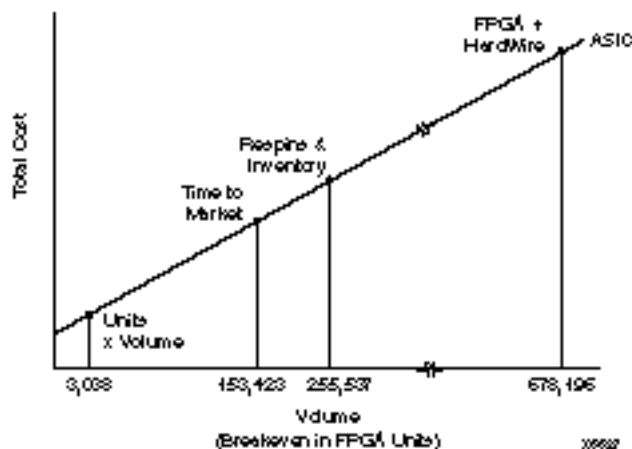


Figure 4. Cumulative Breakeven Points

Using this analysis on any development program can help decision makers avoid making incomplete, non-quantitative design decisions. Long development cycles and loss of flexibility can add far more cost to an I.C. development program than the unit price or NRE.