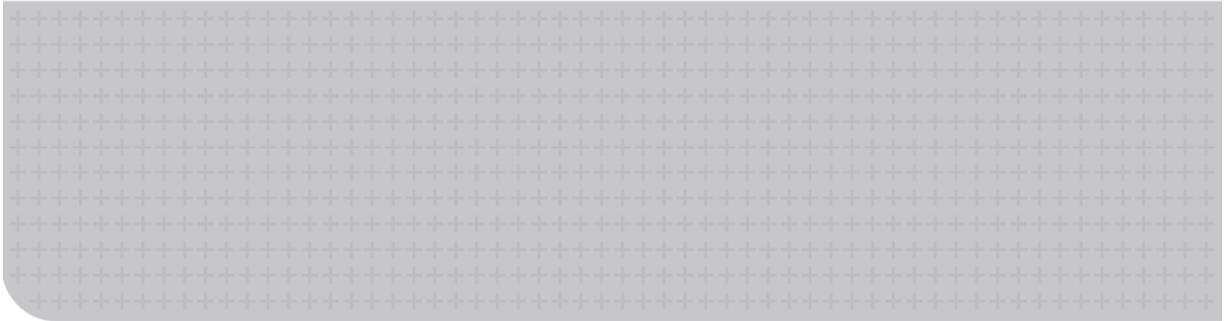




BUILD VERSUS BUY

Understanding the Total Cost of Embedded Design

Table of Contents

- I. Introduction
 - II. The “Build” Approach: Custom Design
 - a. Hardware Design
 - b. Software Design
 - c. Manufacturing
 - d. System Integration
 - III. The “Buy” Approach: Off-the-Shelf Embedded Systems
 - a. Hardware Design
 - b. Software Design
 - c. Manufacturing
 - d. System Integration
 - IV. A Hybrid Design Approach: NI Graphical System Design Tools
 - V. Making the Decision
 - a. When to Build
 - b. When to Buy
- 

Introduction

When developing an embedded system, deciding which portion of the system to design and which to buy off the shelf can be daunting. By designing and building a custom solution, you can customize the end result and optimize costs, but any design specification changes or oversights can cause lengthy and expensive delays. Alternatively, an off-the-shelf platform increases the cost of goods sold (COGS) and you may pay for features that are not necessary for your design. Nevertheless, off-the-shelf systems typically provide a faster design and validation cycle and, therefore, shorter time to market.

National Instruments has gathered customer information and data regarding some of the cost differences between building a custom solution versus using NI off-the-shelf tools. Using this data, we built the [Graphical System Design 'Build vs. Buy' Calculator](#). The calculator can help show the financial differences between building a custom solution versus buying an off-the-shelf system. This paper discusses the benefits and drawbacks of both a traditional custom design approach and off-the-shelf embedded tools.

The “Build” Approach: Custom Design

A custom embedded design requires a wide range of experts such as hardware digital designers, hardware analog designers, software developers, and mechanical designers. Additionally, there could be other engineers and scientists on the team with application domain expertise focused on the application or industry your company is trying to serve. The end result is a large team for a custom design.

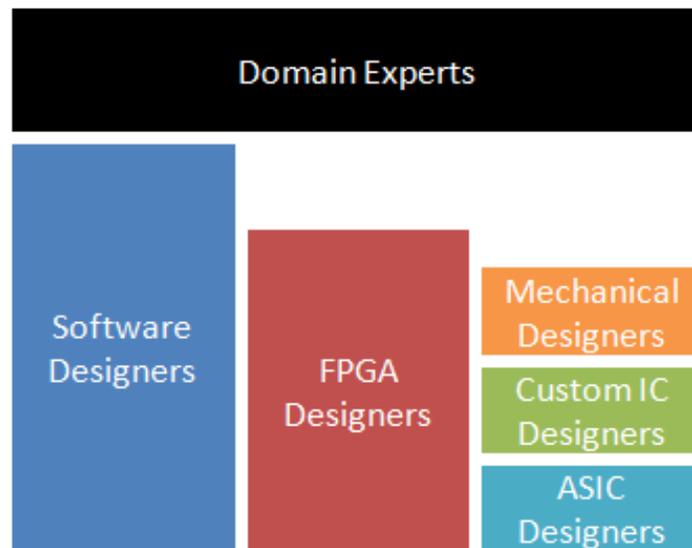


Figure 1. Custom design traditionally takes a large design team with different levels of hardware, software, and application expertise.

Hardware Design

A custom design team faces the big decision of choosing the processor technology – such as the five listed below – that they use as the design’s central processing or control unit.

1. **Microcontroller** – Microcontrollers are extremely cost-effective and generally offer an integrated solution on a single chip, including I/O peripherals. They tend to contain very small amounts of on-chip memory, leaving little room for complexity and expansion. In addition, clock rates are typically in the tens of megahertz, and, therefore, you generally cannot achieve high-performance control or signal processing loops.
2. **Microprocessor** – With a microprocessor, clock rates are higher and there is usually an external memory interface, so performance and expandability are often not a concern. But your applications may require complex driver development because there are usually no on-chip analog peripherals. In addition, high-density packaging techniques such as ball-grid array may lead to a need for more sophisticated manufacturing processes. This adds up to more difficult hardware debugging and manufacturing costs.
3. **Digital signal processor (DSP)** – A DSP is a specialized microprocessor with additional instructions to optimize certain mathematical functions such as multiply and accumulate. DSPs are extremely useful for computationally heavy applications, but you usually need specialized knowledge to take advantage of this software capability.
4. **Application-specific integrated circuit (ASIC)** – An ASIC chip is designed for a specific application rather than general-purpose programmability. ASICs are widely considered a superior technology for technical considerations such as power consumption, size, and cost of goods. However, the development and fabrication costs of ASICs are extremely expensive, and these high costs typically deter all except those with the highest-volume products.
5. **Field-programmable gate array (FPGA)** – FPGAs provide an interesting middle ground between custom ASIC design and off-the-shelf technology. They offer a high level of specialization, but are reconfigurable so you do not have the high fabrication cost that hampers ASIC development. Although you can use FPGAs for a variety of processing applications, complex FPGA design is uncommon because the VHDL programming paradigm is foreign to most embedded software developers who are more comfortable with sequential programming in ANSI C.

In many cases, a single processor technology cannot serve all the needs of an application. Therefore, hybrid architectures have recently become increasingly popular. One such architecture is shown in Figure 2. The real-time processor manages network communication and potentially the user interface while the digital logic manages interfacing to I/O components and high-speed control, timing, and signal processing tasks. This hybrid architecture is becoming common in embedded system design for building control and monitoring solutions.

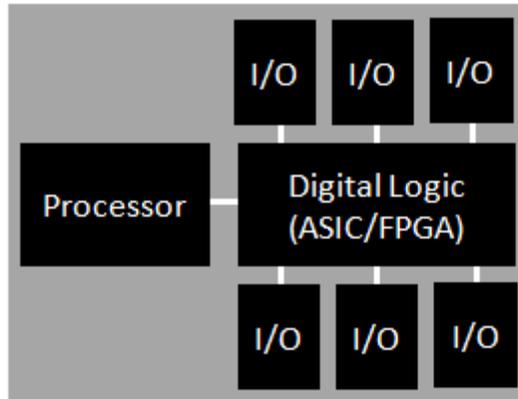


Figure 2. *This ideal hybrid hardware architecture is used within advanced embedded monitoring and control applications.*

After deciding on the processor technologies and the digital design, you must develop the I/O circuitry. If any of the signals needed for the embedded monitoring or control system are analog, the design requires analog-to-digital converters (ADCs) and/or digital-to-analog converters (DACs). A wide range of microcontrollers and processors have ADCs and DACs built into them, but, for most applications, additional analog components are required to build systems with higher analog quality, performance, and channel count. You also need to create the power supply circuitry to power all of the components on your design.

Additionally, a mechanical design is needed to ensure that the embedded design can operate in the environments in which it is to be deployed. Component placement and passive or active cooling should be done to help cool the hotter components of the design.

Software Design

In addition to the hardware digital and analog hardware design, a custom embedded solution requires software design. This often makes up the largest development expense. Within the software design, there are many steps and techniques to developing a real-time software application to execute on the processing hardware you selected. You can use several different tools and architectures to implement the control or monitoring aspects of your design. Software development within a custom design requires expertise for low-level tasks such as real-time OS board-support package development, device driver development, driver API development, application development, and so on.

Manufacturing

Through the prototyping and final manufacturing phases, printed circuit board (PCB) manufacturing is required to evaluate and validate the design. PCBs are inexpensive but require extensive layout effort and large initial costs. Many companies choose to outsource PCB manufacturing because of the capital equipment required. While PCBs are cost-effective for high-volume designs, the length of time involved in iterating on designs can add up. The PCB often needs multiple revisions during the prototyping phase, which can cause delays in getting the final PCB assembled and ready for the final product.

Apart from the PCB manufacturing for the final solution, you should consider sustaining costs associated with maintaining the product. These can include managing part obsolescence and upgrading and maintaining the product over its lifetime. Your team owns the entire design, so the sustaining and upgrade costs of the product will be fairly high.

System Integration

With the PCB(s), other components, and software designed, the last step is putting it all together to form a final product. The system integration step can include finalizing the mechanical design with an enclosure, developing any additional application software, and integrating the different electrical and mechanical components. Mounting and cabling the components within the final design can be time consuming. These components can include the power supply, main controller board, user interface, network interfaces, and so on. The more time spent on mounting and cabling up front can produce a product that is easier to manufacture.

The “Buy” Approach: Off-the-Shelf Embedded Systems

An alternative option to a custom design is to purchase an off-the-shelf system or platform. Although you typically pay significantly more than the cost of the board components, you can expect to reach the market much more quickly because the off-the-shelf vendor already does a lot of the low-level design and implementation work for you. In addition, these systems have smoother expansion paths, so addressing the inevitable feature creep that occurs during concept and prototyping phases is far less painful.

Hardware Design

As with processor technologies, you can consider several deployment technologies for your embedded system. The off-the-shelf platforms are grouped into two categories.

1. **Unpackaged Embedded System** – Available in several form factors (Mini-ITX, PC/104, and so on), unpackaged embedded systems tend to be the most cost-effective solution for off-the-shelf deployment. These systems also have a variety of processor architectures to choose from and a small set of OS and I/O support. However, the software development tools for such a system are almost never integrated, and these systems typically require you to verify regulatory certifications such as EMI and CE compliance along with additional mechanical packaging and cooling work.
2. **Packaged embedded systems** – In addition to featuring the same components as unpackaged embedded systems, packaged embedded systems deliver specifications for shock, vibration, operating temperature, and environmental certifications. These systems are generally more expensive, but they often include integrated software development tools and a more extensive set of integrated and modular I/O options. Examples of modular packaged embedded systems are programmable logic controllers, programmable automation controllers, and industrial PCs.

Some mechanical design is needed when buying a prebuilt embedded system. Unpackaged solutions require more mechanical work than packaged systems, but less work than a custom solution. Packaged solutions normally have built-in cooling and thermal designs so they are the easiest to deploy.

Software Design

Much of the software design work in a custom design is already prebuilt with off-the-shelf platforms. Some vendors are better than others at providing a complete software application solution for their hardware. Some off-the-shelf systems include prebuilt board support packages, device drivers, and even application-level software for development. Vendors that provide both hardware and software tools offer the most integrated solution and can save design teams development costs.

Software development is normally the top development cost of a custom solution. However, an off-the-shelf solution reduces the number of software developers needed, which helps lower development costs and creates a more efficient design process. Smaller teams can iterate on designs more quickly and get final prototypes and products developed much faster.

National Instruments provides a graphical system design platform that includes off-the-shelf hardware with fully integrated graphical software. Customers using graphical system design tools have proven that they can use off-the-shelf tools to build custom embedded systems with much smaller teams. With system level software tools for programming hardware with built-in processors, FPGAs, and I/O, smaller teams can accomplish a task that traditionally would have required twice as many people. The result is a decrease in hardware and software development expenses that typically are the top expenses for an embedded design.

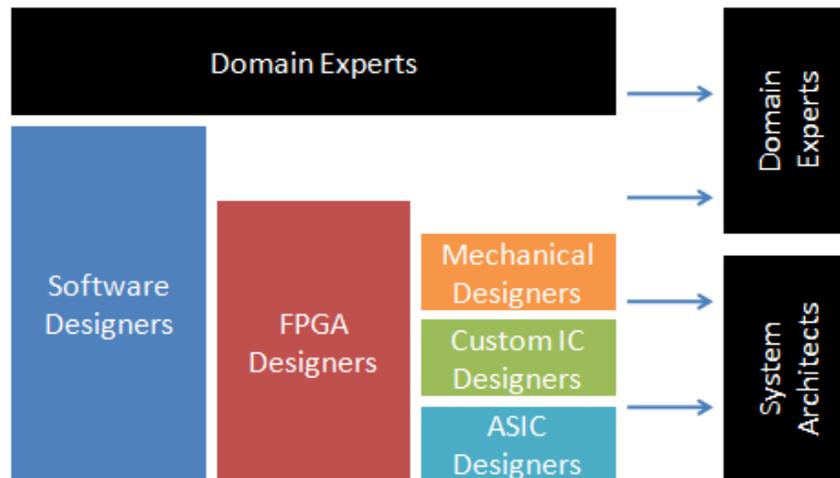


Figure 3. Smaller teams can use an off-the-shelf platform with integrated software and hardware to accomplish embedded tasks that traditionally would have required a large specialized team.

Manufacturing

Off-the-shelf technologies eliminate the need for PCB manufacturing. In some cases, design teams might need to create a custom PCB for a specific I/O or timing need, but those designs are normally less complex. More off-the-shelf vendors are integrating FPGAs within products, which can give a design team the ability to customize a prebuilt system without the need to build a custom PCB.

Sustaining costs for off-the-shelf solutions have proven to be much more cost-effective compared to building a custom board or device. For off-the-shelf solutions, vendors assume the burden and costs of factors such as part obsolescence management, environment ratings, and certifications. The result is a much lower sustaining cost for off-the-shelf solutions versus building a custom board or device.

System Integration

There is little to no difference in the system integration work required for off-the-shelf solutions versus custom designs. Integrating the prebuilt embedded system with other electrical and mechanical components is needed. This involves similar tasks such as creating additional application software to interface the off-the-shelf system to components such as displays, sensors/actuators, and network interfaces along with creating an effective mounting and cabling solution.

A Hybrid Design Approach: NI Graphical System Design Tools

National Instruments graphical system design tools, including NI LabVIEW graphical software and reconfigurable packaged and unpackaged embedded systems, combine the benefits of an off-the-shelf platform with the customization and flexibility of custom hardware.

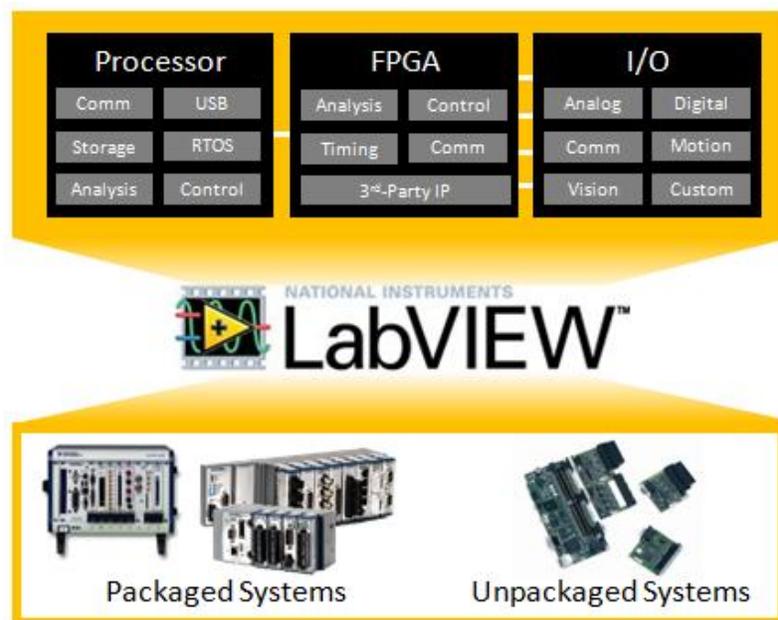


Figure 4. National Instruments graphical system design tools including LabVIEW and reconfigurable embedded hardware combine the benefits of off-the-shelf tools with the performance of custom design.

[NI embedded hardware](#) includes rugged packaged systems such as NI CompactRIO, high-performance packaged systems such as PXI, and unpackaged systems such as NI Single-Board RIO. All NI hardware systems share the same reconfigurable I/O (RIO) hardware architecture of a processor, FPGA, and modular I/O. Each component of the architecture is programmable with [LabVIEW](#) tools, which gives engineering teams the ability to prototype and deploy embedded systems faster using fewer hardware and software designers. LabVIEW is an open environment that allows engineers to integrate existing ANSI C/C++ code, text-based math models, and VHDL IP.

On average, NI customers have reported that they can get to market 50 percent faster using 20 percent fewer engineering resources with off-the-shelf graphical system design tools. The applications that fit NI tools the best are custom monitoring and control applications that involve specialized analog I/O and advanced control that require custom signal processing or control algorithms. Typically, these applications are in areas and industries of innovation that are developing new control and signal processing products. Industries and applications such as energy, medical, machine control, and big physics are a few of the areas where NI products are used. Below are two examples of companies that have been successful developing embedded systems using NI tools.

KC BioMedix is a small company that developed a product called the NTrainer System using CompactRIO and LabVIEW. The device helps premature babies learn to oral feed and greatly increase their chances for survival. Initially, KC BioMedix looked at outsourcing the design to a third-party company of embedded specialists to build a custom solution. When it became clear that the cost was too high, they decided to bring the development in house. In only three weeks, they created a proof of concept with CompactRIO and LabVIEW demonstrating the ability of CompactRIO to replace the need of a custom embedded solution.

“With National Instruments LabVIEW and NI CompactRIO, we were able to reduce our development cost by \$250,000. In addition, we were able to reduce our development time from four months to four weeks, and avoid the necessity of developing custom control software and drivers.”

[Read the full KC BioMedix case study](#)

Saara Embedded Systems is another example of a company using NI graphical system design tools to reduce time to market and development costs. Saara engineers used LabVIEW and NI Single-Board RIO to develop a Remote Facility Management System (RFMS) to precisely monitor and control a facility's total power consumption. With the flexibility of the embedded remote terminal unit, their customers can monitor and control different points on their infrastructure, making the RFMS an ideal system for effective energy consumption and optimization.

“A short time to market for our design was paramount. The NI platform allowed us to rapidly prototype with NI Single-Board RIO and LabVIEW in the record time of two months and saved us six months of development time.”

[Read the full Saara Embedded Systems case study](#)

Making the Decision

Mostly, technical capabilities are not the determining factor when deciding between build and buy. Rather, it usually comes down to a simple financial analysis. If the return on investment of the engineering cost incurred in product development is justified by eventual profits, then you have made a good decision.

	Custom Design (Build)	Off-the-Shelf (Buy)
Hardware Design Cost		
Hardware Component Cost		
Software Design Cost		
Manufacturing Cost		
Sustaining Cost		
System Integration Cost		

  
 Good Better Best

Table 1. Compare the financials of build versus buy.

To make an educated decision, you must accurately estimate the cost of building your own custom solution. This is never as simple as it seems; if you just add the cost of the board components plus the hardware and software development time, you grossly underestimate the total investment. You need to consider other “hidden” costs before accurately assessing the true cost of the job. For example, manufacturing and inventory costs typically account for an additional 25 to 35 percent of the COGS of the system. In addition, on average about 30 percent of total software development time is spent on OS, driver, and middleware development – though by choosing a packaged platform with integrated hardware and software, you can eliminate the need for this “board bring-up.” Also, you need to account for other hidden costs including environmental regulations, validation, end-of-life components, and last-minute specification changes forcing design alterations and complete redesigns.

When to Build

So is the lesson to never develop a custom board or product? Definitely not. Rather, when you are assessing technologies that are crucial to your product’s success, be sure to determine early on what projects better match a custom approach. With a custom solution, you can completely customize the

end result and optimize costs, but any design specification changes or oversights can cause lengthy and expensive delays. The following are typical characteristics of projects that are more suited for a custom-built solution:

- High volumes (10,000+ per year)
- Larger design teams
- An iteration on an existing custom design
- Custom size or shape required
- Very stringent technical requirements (such as ultralow power consumption)

Also keep in mind the confidence level you have regarding how many units you expect to sell per year. Engineers and management are often overly optimistic in the volumes they project, especially the volumes within the first two or three years. If you are unsure of the volumes you expect to achieve, it might be financially more advantageous to start the design with an off-the-shelf solution to save time and money and lower your risk. If the volumes increase over time, you can always choose to optimize costs and create a custom board or product later.

When to Buy

Off-the-shelf vendors, such as National Instruments, make tools more productive and give companies and smaller engineering teams the ability to get their products to market faster. In areas where you can use off-the-shelf technology, let your suppliers take on the logistics and costs of the design, including development and sustaining costs. Then you can focus on differentiating technology to make your product better and get to revenue sooner.

When working with a new technology or innovation, getting a prototype working quickly is often imperative to determining whether the product has technical and business value. With newer products and technologies, it is difficult to certainly determine the market demand and unit volumes the product will achieve. Therefore, within areas of innovation such as clean technology, medical and life sciences, and robotics, off-the-shelf tools provide teams the ability to quickly iterate on designs and prototype systems faster without having to invest too much capital up front. This allows design teams to focus on gathering market requirements and feedback much earlier in the design process. The following are typical characteristics of projects that fit with using off-the-shelf technologies:

- Low- to mid-volumes (100–10,000 per year)
- Smaller design teams
- New technology or product (for example, clean technology, medical, robotics, and custom machinery)
- Uncertainty of market demand or potential
- Critical time-to-market pressure

Additional Resources

Try the Graphical System Design Calculator (www.ni.com/buildvsbuy)

Learn more about NI graphical system design tools (www.ni.com/embedded)