White Paper

FPGA

An Intel Company

Lowering the Total Cost of Ownership in Industrial Applications

MAX® 10 FPGAs and Cyclone® Series FPGAs provide product success with profitability.

Authors

Jason Chiang
Sr. Technical Marketing Manager
Intel Corp.

Tom Schulte
Sr. Technical Marketing Manager
Intel Corp.

Stefano Zammattio
Sr. Technical Marketing Manager
Intel Corp.

Table of Contents

Introduction

Approximately one-third of embedded designers surveyed on the adoption and use of FPGAs for embedded applications responded that they perceived FPGAs as too expensive to use in their designs. However, a look at the total cost of ownership (TCO) at the system level (as measured by development, enhancement, replacement, and maintenance costs over the lifetime of the product) reveals that Intel® MAX® 10 FPGAs and Cyclone® series FPGAs offer competitive and flexible alternatives to discrete microcontroller unit (MCU)/digital signal processor (DSP)/ASSP products.

Facing neverending global competitive and economic pressures that continue to threaten their business models and bottom line profitability, industrial automation and process-control manufacturers are constantly grappling with cost challenges, including:

- Profitability versus R&D investments
- Time-to-market pressures to adapt to changing economic conditions
- Effective use of limited resources to update existing products, replace existing products, or launch new products
- Managing the product life cycle

This white paper uses a design example to help designers—system, hardware, and software engineers—understand how they can take advantage of MAX 10 and Cyclone series FPGAs and to realize a lower TCO as measured by development, enhancement, replacement, and maintenance costs over the lifetime of the system. As shown in Figure 1, a lower TCO over time directly contributes to increasing the gross profit, thereby relieving a pressure faced by most of today's design teams.

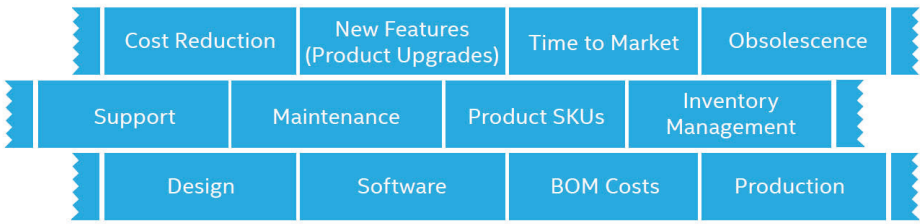

Figure 1. MAX 10 FPGA Power Sequencing of Arria 10 FPGA

FPGAs Lower the TCO

To illustrate how MAX 10 and Cyclone series FPGAs contribute to a lower TCO, this white paper uses a drive control application (Figure 2) as an example that can apply to any industrial design challenge a customer may encounter. A traditional drive architecture consists of a control module to run the algorithms to generate position and/or velocity set points and to close a control feedback loop. The control module sends these set points to a drive controller that converts them into electrical signals (current or voltage) to drive one or more motors and/or actuators, which generate the torque to move the load or mechanical components in an inverter or servo drive system. Feedback sensors such as encoders and Hall-effect devices provide the position or velocity of the motor/actuator to the motion controller to close the signal loop.

MCU and DSP devices are the mainstay of motion-/drive-control architectures today, with FPGA architectures gaining momentum. MCU/DSP architectures enjoy an established user base and established architectures, development tools, and motion- control algorithms used primarily in single-axis drive applications. As the complexity of drive systems and the number of drive control axes and product features increase, MCU/DSP architectures quickly run out of the performance overhead and flexibility needed to keep pace with changing market requirements. As system performance increases, designers can only increase MCU/DSP frequency and optimize the software algorithms to a certain point.

To address this problem, designers use multiple DSP devices, a combination of DSP and MCU devices, or a combination of MCU/DSP devices and/or FPGAs to partition the performance and functions in their design. While MCU/DSP architectures do enable some degree of code reuse, reuse of highly optimized code is a labor-intensive process and is difficult to partition and move to new devices. Therefore, design methodologies based on MCU and DSP devices require heavy hardware and software resources to partition the application function and performance across x number of devices. Depending on the complexity of the software, this approach requires several months to over a year (development time, t) to port the application to the distributed architecture.

Migrating to Industrial Ethernet

In addition, industrial networks are migrating to Ethernet-based networks, often with drive systems connected to these factory networks. While MCUs and some newer digital signal processors can support (standard) Ethernet TCP/IP with software overhead, this combination can be problematic for several reasons:

- Most MCUs lack the bandwidth for and most digital signal processors are not capable of processing Industrial Ethernet and Fieldbus protocols concurrently with drive control
- MCUs tend to be limited in their PWM output to drive precision motion control
- Many DSP devices are not capable of addressing TCP/IP stacks because the architecture lacks the word alignment feature required to support TCP/IP

These difficulties mean that designers are forced to use an additional MCU, ASSP, or FPGA device to bridge the current product to industrial networks. Designers must also contend with the fragmented nature of Fieldbus and Industrial Ethernet protocol standards, such as:

- DeviceNet and EtherNet/IP
- Profibus and PROFINET RT/IRT
- CANopen and EtherCAT
- CANopen and PowerLink
- SERCOS I/II and SERCOS III
- CC-Link and CC-Link IE

To further complicate matters, Industrial Ethernet protocols such as EtherCAT, PROFINET IRT, and SERCOS III also require a protocol-specific MAC to address their determinism and real-time requirements. Most MCUs and digital signal processors do not support protocol-specific MACs. This problem can be addressed by using a MAX 10 or Cyclone series FPGA to reconfigure the MAC IP, thereby eliminating the need for a different MCU, ASSP, or ASIC to support the different protocol standards.

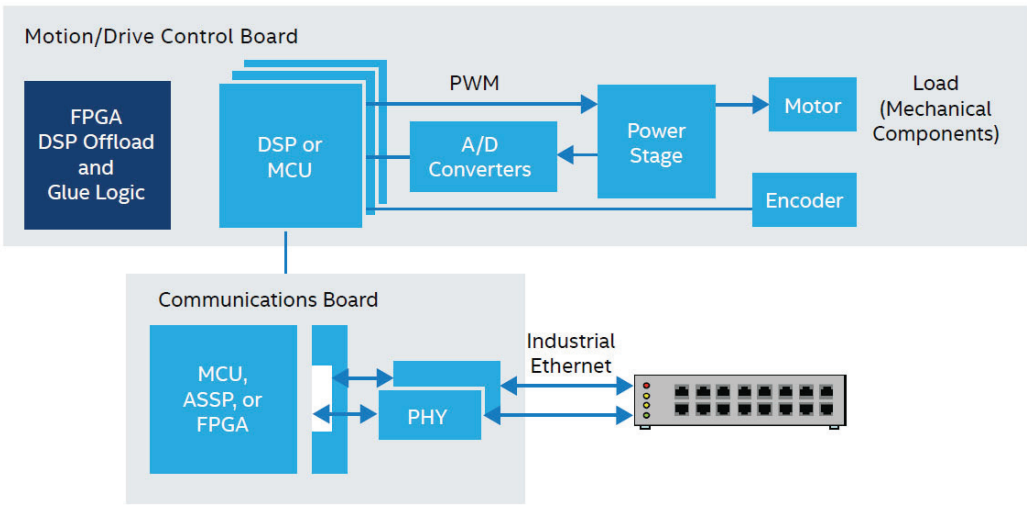

Figure 2. A Traditional Drive System

For designers involved with safety drive applications, functional safety adds yet another degree of complexity to the design. Safety functions must be separated from the non-safe parts of the application, including the communications path. Cyclone series FPGAs can help mitigate the development time and risk by using TÜV-qualified components to help shorten the end-product certification time and to enable safety, drive control, and communications to be integrated into the same FPGA device.

For further information on functional safety, please refer to the white paper, Developing Functional Safety Systems with TÜV-Qualified FPGAs.

Decreasing Time to Market

Using drive controls with one to several, x, MCU or DSP devices, with the addition of networking and potential safety requirements, will increase the development time, t, by another 18 to 24 months. The additional time translates to increased R&D costs and potential lost revenue and profit. The BOM cost for the product also increases when additional components are added to the board.

With devices such as the MAX 10 or Cyclone series FPGAs, designers can address drive control, industrial networking, functional safety (if required), and new product features, along with any design updates on a single device and on a single hardware platform. The initial FPGA development time may not be shorter than the initial MCU/DSP development time, but time-to-market benefits can be realized with the same FPGA platform should changes need to be made to the design.

MAX 10 and Cyclone series devices offer performance at a low cost with the ability to embed ARM* processors, Nios® II soft processors, or DSP blocks that outperform equivalent MCU/DSP devices running on optimized software. Designers are also able to embed additional features, such as sensor interfaces (e.g., some encoders are available only as IP for FPGAs), high-precision PWM control, and other custom logic. This hardware-optimization approach allows a measure of product differentiation (Figure 3), and in some cases, it can provide a measure of protection against software applications migrating from the drive systems into the programmable logic controllers (PLCs) or host PCs. In addition, designers can reconfigure the FPGA on the same hardware platform to support each communications protocol needed without re-spinning a new board. Customization on a common FPGA platform allows designers to differentiate their products and release products quicker than when using MCU/DSP-based solutions.

Using Existing Software

At this point, it is worth mentioning that all is not lost with manufacturers' existing software investment in MCUs and DSP development. Software engineers can apply their MCU/DSP experience toward programming embedded CPUs, such as the dual-core ARM Cortex*-A9 MPCore*, Nios II embedded processor, the ARM Cortex-M1, and Freescale's ColdFire V1 cores. The development tool flow and operating systems (i.e., Linux) are very similar.

Today's electronic products are far more capable, flexible, and complex than they were 10 years ago, and include features enabled by the use of processors, operating systems, and application software. Many products have evolved to the point where there are many more man years invested in software design rather than hardware design. The implication is that when considering a product upgrade, the selection of a processor that does not support the same operating system as the current system can result in a significant amount of software porting, severely restricting the project's choice of devices and flexibility.

Popular open-source operating systems such as Linux or eCOS (supported on soft processor cores and a host of external processors) have the benefit of an active developer community constantly working to improve the operating system and develop new applications and features. These improvements and features potentially can save designers many man-hours of effort in developing and supporting a product throughout its lifetime. Fortunately, not only is there a wide range of soft and hard processor cores (Nios II embedded processor, ARM Cortex-M1, ARM Cortex-A9, Freescale V1 ColdFire, etc.) available for implementation in Intel low-cost FPGAs, but there is also a wide choice of operating systems available. For example, the Nios II embedded processor supports the full open-source and commercial versions (i.e., Wind River, Timesys) of Linux*, eCOS, ThreadX, Nucleus, and other real time operating systems.

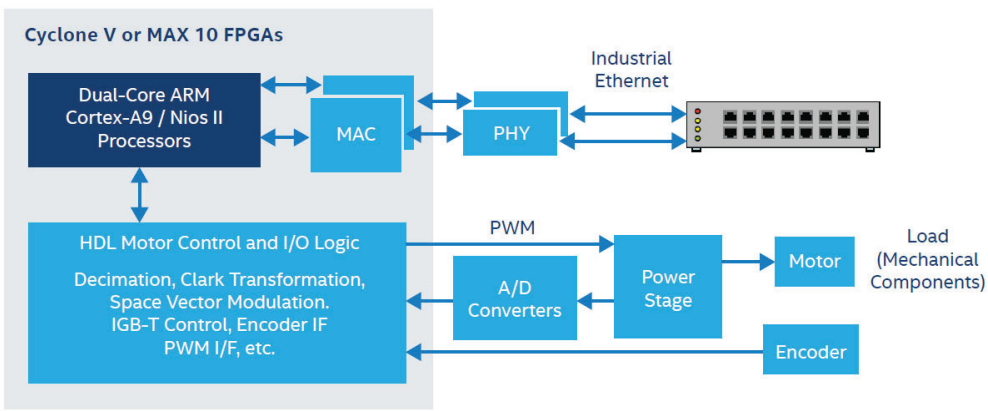

Figure 3. MAX 10 FPGA or Cyclone Series FPGA

To a hardware designer, the benefits of FPGA devices with an embedded processor are compelling. With full support from many operating system companies and the feature enhancements possible, many software engineers are beginning to take advantage of the benefits that FPGA-based systems can bring over the lifetime of a product. It may take a little work for both hardware and software engineers to learn the balance between C-code for software and the VHDL coding of FPGA design.

However, once designers are up to speed on FPGA design methodology, using FPGAs should be more cost effective and flexible than solutions based solely on MCU and DSP devices.

Making Changes to Multiple Designs

So far, this white paper has discussed how FPGAs can be used to address the development challenges and costs on one program. What happens when multiple products result from the original design? What happens if design updates and feature enhancements need to be added to all of those products? New product features such as a change in industrial networking protocols would likely require a modified or new board for each product release (or SKU).

Using the drive example, if y products are developed from this base platform with different features, then the design teams must first implement their hardware and software by x devices per board, take t amount of time, and then instantiate their hardware and software by y times for those products over however long a period, t2, it will take to complete these multiple products. This translates into a cost factor of $((x \times t) \times (y \times t2))$. Design teams already know that this methodology is not a trivial undertaking, may negate any processes for getting products quicker to market, and may negate any desired lifetime cost savings, which include development, enhancement, support, maintenance, and replacement costs.

To facilitate the process, designers can take advantage of the Cyclone series architectures to design a common hardware platform that supports multiple product lines (SKUs) and features. (Refer to Figure 4 for an example from Softing GmbH, an Intel FPGA intellectual property (IP) and solutions partner who specializes in data communications technology for the industrial automation and automotive industries.) After the initial design, an FPGA-based product could save a design teams many months (if not more) of engineering work per subsequent product based on the original design by allowing engineers to reconfigure the Cyclone series FPGA without the need to re-spin the PCB.

Take the example where a drive manufacturer needs to provide products that support several networking protocols, such as EtherCAT, PROFINET (RT/IRT), and Modbus/TCP. Not only does a MCU/DSP solution require an extra device to support the communications channel, but it also requires three boards. Designers face developing or licensing the protocol-specific MAC IP (if required) and protocol- specific stacks along with three boards at a potential cost of up to $300K (or $100K per board), in addition to the corresponding software development. (Development costs for each customer will vary, so the actual cost may be more or less.)

However, designers can use MAX 10 or Cyclone series FPGAs to integrate drive control and Industrial Ethernet on the same FPGA and use the same hardware platform to support multiple products lines (SKUs) and the desired features. Instead of developing three boards (one for each IE protocol standard) in the scenario previously described, a vendor can potentially save up to $150K to $200K in MCU/DSP development costs along with significant BOM cost savings by using fewer components and fewer PCB variants in production. With fewer PCB board designs, manufacturers also can streamline the logistics involved in building and shipping their products. An FPGA design methodology helps engineers overcome the $((x \times t) \times (y \times t2))$ cost factor after all of the development resources and times have been added up.

When designs reach the maximum potential of the FPGA device, designers have the option to migrate to a higher density device or re-compile the design and move quickly to another Intel FPGA. Another way to look at this is as a fast upgrade path that enables designers to integrate functions and minimize/eliminate MCU, DSP, and other components from the board.

Designing with Device Reliability

With products shipping in volume, TCO concerns remain present in the form of maintenance and replacement costs. FPGAs such as the MAX 10 or Cyclone series have a reputation for quality and reliability. Device reliability over time means lower product maintenance in the field, which helps reduce the maintenance cost component of TCO, so limited resources instead can be spent on developing new products. Intel maintains in-house reliability stress, failure analysis, and associated staffs to support qualification of all new die and packaging technology families, of which all qualification procedures meet all Joint Electron Device Engineering Council (JEDEC) requirements. Intel regularly achieves greater than 20-year useful life under nominal operating conditions.

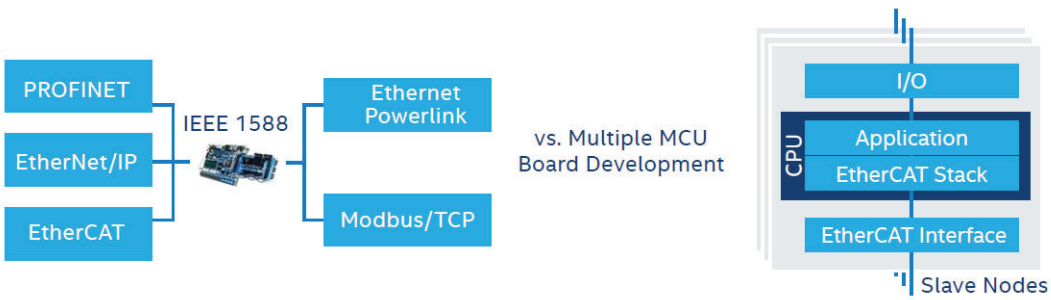

Figure 4. Softing GmbH on Using a Single FPGA Platform to Support Multiple Industrial Ethernet Protocol Standards with a Single Generic API

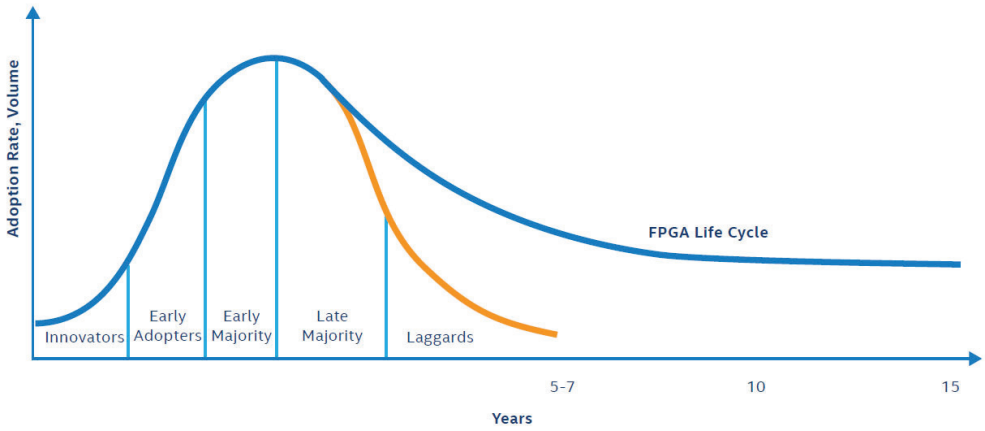

Figure 5. Life Cycles of a Typical Intel FPGA vs. a MCU/DSP Device

Taking Advantage of Longer Life Cycles

More than just product reliability contribute to long-term TCO. Compared to the life cycles of Intel FPGAs, many MCU and DSP devices have significantly shorter life cycles, typically in the five-to-seven-year range, because their vendors tend to obsolete mature devices much sooner that Intel, as shown in Figure 5. Intel's strategy is to support long life cycles, typically up to 15 years, instead of obsolescing mature products, thereby helping customers avoid the higher cost of obsolescence management.

Long FPGA life cycles align well to the longevity of industrial products and help customers mitigate design risk, thereby reducing their TCO. Given Intel's extensive product families, designers can move up to the next device class or migrate their design to a new FPGA family when they exceed the capabilities of their current FPGAs (see Table 1)

FPGAs for Industrial Applications

As the intelligent drives example illustrated, Intel devices such as the Cyclone IV E, Cyclone V GX, Cyclone V, and MAX 10 FPGAs provide the design flexibility to address hardware and software changes including support for multiple product SKUs using the same hardware platform, IP reuse, obsolescence-proof designs, and inventory management. All these factors play important roles in achieving bottom line profitability. Depending on their market needs, customers requiring more performance than the MAX 10 or Cyclone series FPGAs can migrate to larger Intel products, such as the Arria® family of FPGAs.

For applications requiring external host processors for performance reasons, designers are moving to mainstream architectures such as the Intel® ATOM™ processor and other PCI Express* (PCIe*)-based processors. Designers can leverage the strong software echo-systems of such processor architectures. They can also use the high-speed PCIe interfaces included on these processors to communicate with an I/O companion chip to integrate an array of peripherals and I/O options and support any variant of Industrial Ethernet protocols, SATA, and other IPs. Figure 6 illustrates the flexibility of the FPGA I/O companion chip architecture.

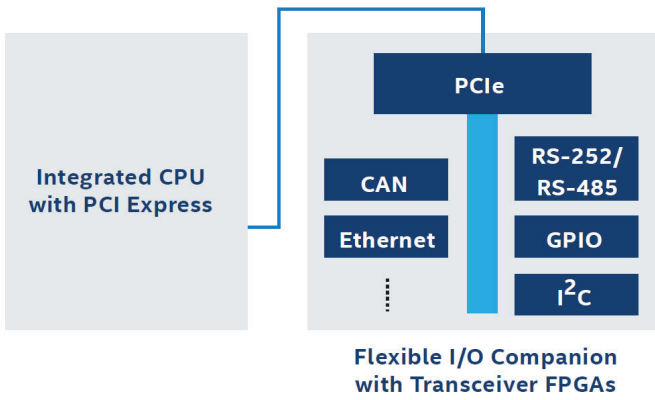

Figure 6. Cyclone FPGA with Transceivers for I/O Companion Chip Applications

FPGA	Logic Elements	Total Memory (Kb)	18x18 Multipliers	Transceiver I/Os	PCIe* Hard IP Block	User I/Os	ADC	User Flash (Kb)
Cyclone V GX FPGA	31,500 - 301,000	1,468 - 15,137	102 - 684	3 - 12	1 - 2	208 - 560	n/a	n/a
Cyclone V SX SoC	25,000 - 110,000	1,678 - 6,748	72 - 224	6 - 9	2	326 - 469	n/a	n/a
Cyclone V ST SoC	85,000 - 110,000	4,847 - 6,748	174 - 224	9	2	469	n/a	n/a
MAX 10 FPGA	2,000 - 50,000	108 - 1,638	16 -144	n/a	n/a	27 - 500	0 - 2	96 - 512
Arria® V FPGA	75,000 - 504,000	8,463 - 27,046	480 - 2,312	9 - 36	1 - 2	416 - 704	n/a	n/a

Table 1. Cyclone and MAX 10 FPGAs

Conclusion

Whether designers are making product enhancements, replacing products due to feature or component obsolescence, or launching new product variations, perhaps from scratch, MAX 10 and Cyclone series FPGAs offer the design flexibility to help designers achieve their ultimate goal: product success with profitability. Aligning to an Intel FPGA methodology to complement or replace existing standard product designs will help companies achieve lower TCO, as measured by development, enhancement, replacement, and maintenance costs over the lifetime of the product.

Where to Get More Information

For more information about Intel and MAX 10 FPGAs, visit www.intel.com/content/www/us/en/products/details/fpga/max/10.html

For more information about Intel and Cyclone FPGAs, visit www.intel.com/content/www/us/en/products/details/fpga/cyclone.html

1 https://www.intel.com/content/www/us/en/industrial-automation/programmable/applications/overview.html

2 https://www.intel.com/content/www/us/en/content-details/650497/a-flexible-solution-for-industrial-ethernet.html

3 https://www.intel.com/content/www/us/en/content-details/650481/developing-functional-safety-systems-with-t-v-qualified-fpgas.html

4 https://www.intel.com/content/www/us/en/products/details/fpga/cyclone.html

5 https://www.intel.com/content/www/us/en/products/details/fpga/nios-processor/v.html

6 https://www.intel.com/content/www/us/en/products/details/fpga/development-tools.html

7 https://www.intel.com/content/www/us/en/support/programmable/support-resources/fpga-training/overview.html

altera™
An Intel Company

WP-01122-1.1