Systems Integration

You must remember that unless you are working in basic research, you are not inventing new technology, you're integrating existing technology. Unless you are working at the semiconductor physics level, you are doing systems integration at some level. From a project at any scale, even up to large manufacturing, refinery, telecommunications facilities. It is all systems integration. From stuffing a box with a couple PCBs, a power supply, some user interface controls, the level we're talking about in this book, to a hundred of equipment racks stuffed with video and audio gear to cover the Olympics, or provide a programming service like DirecTV, the goal is the same. That is to get all the parts involved to work as a cohesive system in the most efficient, and usually the most economical way possible.

As we will cover in much greater detail there is a general order that a project, be it a system comprising many products working together, or a simple single new product that follows. We briefly touched upon the process in the book's introduction. But let's briefly touch upon how prototyping is done today.

Prototyping today

From the financial standpoint, less capital is required and there are more options to raise money. From a development standpoint there's no need to develop custom components, there are many off-the-shelf base designs in the form or development boards, and widespread, easy to integrate RF modules, and built in web server interfaces. For manufacturing there is no need to set up your own factory anymore, as all can be outsourced. When you have a product to ship there are more ways to market and sell your product. As we will see shortly almost all off-the-shelf development boards are FPGA or CPLD based, and they come with some limitations.

While CPLD is good for simpler electronic circuitry, or for "glue" circuitry between other separate system blocks, they can't handle complex electronic processing. The good thing about CPLDs is that they are instant on, verses a FPGA's requirement to load its configuration from external RAM. The fact that the configuration is stored externally means you will need to obfuscate that data in some way. While some FPGAs have built in flash memory that it automatically boots from, often CPLDs are used to instigate FPGA startup.

If you need to do a consumer product, the considerations change considerably. Unless it is to be a high-end product that can stand the cost, or will see limited consumer demand, ASICs generally make more sense than FPGAs in these situations. To be economical many consumer products require an ASIC to integrate the products operation to keep the price of the end product low enough to meet the consumer market. Any very high volume, low target cost product must use an ASIC. Why? The electronic performance is optimized with a product specific ASIC, and the cost per unit is much lower. In addition, an ASIC is more energy efficient, and these can have a combination of digital, as well as analog circuitry. While FPGA might have internal analog circuitry such as PLLs, ADCs, etc, designs rendered on FPGA based products can only be configured into digital blocks. The entry cost of ASIC development has gone up to the $2,000,000 mark at least. So, easy entry to the consumer market may not be so easy!

So, ASIC based products have very high initial development costs, while FPGA has extremely little up-front hardware cost, the total cost/volume line is much flatter for ASICs than the much steeper FPGA line. At some point on the volume axis the two lines obviously cross. The decision from a purely engineering and sales perspective will the volume reach that crossover point. For a startup the decision is not so simple as the upfront money might not be so simple to raise.

It should be noted that FPGAs are often used to prove the concept and design before that configuration is rendered and etched at an ASIC fab. Today there are many pure-play ASIC fab houses. Pure play means they accept the designs submitted by others and produce the resulting ASIC. Even many large companies no longer do their own IC fab. Fabless companies include Broadcom, AMD, Nvidia, and Qualcomm. Even companies with their own fab operations outsource some production to these houses, such as Intel, and TI. PLD companies link Altera and Xilinx also do the same.

These fab houses often provide simulation software/hardware black box that allows a new design to be refined and verified before it goes out for fab. While these are not cheap, often in the $50K range, it might allow a startup to survive on its initial funding, long enough to prove the concept, before requiring a second infusion of cash from a VC, etc. Second rounds of funding are often even more difficult than the first. This is because VCs will make several small bets, and then see which ones become the most promising before making substantially larger bets.

You must also realize that the electronics may not be the most expensive part of a product and so each product must be analyzed based on its volume/pricing and other components.

VOLUME/PRICE*	LOW VOLUME	MEDIUM VOLUME	HIGH VOLUME
LOW COST	OFF THE SHELF SPECIAL FUNCTION	FPGA AND PROCESSOR	ASIC
MEDIUM COST	MICROCONTROLLER WITH DISCRETE SEMICONDUCTORS	FPGA WITH EMBEDDED PROCESSOR	HARD WIRED FPGA AND SPECIAL FUNCTION
HIGH COST	MICROPROCESSOR WITH FPGA	MICROPROCESSOR WITH COMPLEX PERIPHERALS	HARD WIRED FPGA SPECIAL FUNCTION AND MICROPROCESSOR

Consumer market – high volume, low price, short lived.

Industrial market – Medium volume, medium to high price and technology. Long life cycle
Military/aircraft – low volume, extremely high tech, very long support/life cycles