CIRCUIT CELLAR ONLINE

FEATURE ARTICLE

Robert Kondner

Debunking Common Engineering Myths

Robert takes us through several areas of engineering that people often have misconceptions about. He divides the black-and-white from the gray and shows us that engineering and engineering management should not be mistaken as the same.

One detail often overlooked in the field of product development is that engineering and engineering management are two different tasks. Although most engineering tasks can usually be reduced to black-and-white questions of volts, amps, and seconds, engineering management questions often hover in a gray cloud of compromise. A well-managed project identifies and resolves the gray areas as early as possible and understands the consequences of doing so.

Feeding this gray area and slowing the process of making decisions that affect a product's design is an array of well-heeled vendors of components, tools, and services. How many salespeople want to talk to the VPs of engineering (affectionately known as "decision makers") instead of an engineer? Why? It is an attempt to help shape how those gray clouds contract into hard decisions that directly affect the generation of purchase orders.

Unfortunately, years of consistent bugle calls from numerous vendors have made their way into project management logic. The purpose of this article is to expose the fallacies in some of these areas.

MARKET WINDOW/PRODUCT LIFE

If you could save all the time to market that most FPGA vendors claim, then simply switching vendors two or three times during the course of a project would have your FPGA logic finished before you started! Obviously this doesn't work. The middle of a project is no time to switch tools and still remain on schedule. But, a schedule is required for a successful project, and there definitely are decision points and drop-dead dates.

Consider the real constraints of your project when selecting tools and components, and the major learning curves associated with using these new tools and parts. The safest approach is likely to be the same approach used in your last project. New technology is great, but be prepared to take some lumps during the learning process.

Not all engineered electronic technology fades away within six months of the design start. Most firms have a number of ancient designs using 6502 and Z80 processors or something similar. Although these older designs lack some of the features or speed of newer devices, the technology in this design is often the starting point for the next model. Technology incorporated in a design often outlives the product, and the technology developed is often the lifeblood of the organization. If quality technology is desired by management, both time and money must be allocated to its development.

Engineering management does the organization a great disservice if, in the drive to get a new product out the door, long-term migration is not considered. It is expensive to restart a

finished project or to add legs and brains that were once considered bells and whistles. The old phrase "Do it right the first time" has to win out over "Keep it simple."

These technology development issues are not questions to be asked as you hit the data books (or Internet) looking for implementation details. A successful project will consider these up front with the knowledge that the developed technology will probably have a significant life. If management cannot deal with these issues, it's time to change management.

OBSOLETE PARTS

If I had a nickel for every vendor that told me I was using an obsolete part, I might not be able to retire, but I could buy a nice lunch. The salespeople alone are not to blame—manufacturers pushing new products are feeding the frenzy. From a feature standpoint, a manufacturer's new part might make the previous part obsolete for a new design, but it does not imply that the older part should not be used in a new design.

REDUCING PART COUNT REDUCES COST

I always get a chuckle when someone selling a new wiz-bang component takes the total assembly cost and divides by the component count to show the value of reducing part count. I chuckle because some contract board assembly is based on component cost, not component count. Using a more expensive component will increase assembly cost. A good assembly shop will consider component counts, but as a portion of the quoted cost, the savings on one or two components is simply not important.

Even major savings in PCB space can be difficult to justify. If changing a component can reduce PCB layer count, then go for it. However, using a couple of 50-mil SOICs in place of a single, large 20-mil QFP will bring a smile to your assembly folks. Whether or not that smile means a lower assembly cost is another question. Unless a board is tight, 50-mil SOICs are easier to route, place, solder, and test.

OVER THE WALL VS. CONCURRENT DESIGN

We hear a lot about concurrent engineering. I believe that the best environment in which to develop a product is one with good communications. In the real world, however, not all individuals have the skills to be involved in all areas of expertise. Any good schedule has points called milestones, which usually include a set of deliverables to the next phase. Good milestone examples include a netlist or mechanical drawing. One advantage of over-the-wall structures is, if you pass something incomplete, it bounces back into your lap.

It has been my privilege to pass projects to qualified PCB-layout and mechanical people. It has always been my experience that giving them both information and independence results in incredible productivity. If you want to win the Super Bowl, you need good players who know how and when to pass the ball.

CODE REUSE AND OBJECT ENVIRONMENTS

I must admit that I have a bias here that is primarily based on the observation of a limited number of projects. The worst failure among code projects has been with large-scale OOPs code. Maybe they would have failed faster in another environment—large code projects are always risky. Just for the record, I like OOP Visual tools. I have even used Delphi to write a few small applications, and I love the tool. I have looked at Visual Basic and Visual C++, and I hold both in high esteem.

I have a problem with the fact that after a logic function is coded into a particular OOP environment, it's impossible to reuse it in another environment. Not all projects have a Win32 system with virtual memory management, event manager. Most OOP systems use a library with more APIs than any programmer could ever hope to use, much less fully understand. Lots of code goes into PIC processors and the like. With these small environments, anything more than a few lines of C quickly moves to assembly. A good project should produce reusable code, whether it is in a WIN32 or PIC environment.

CHARGE *N* TIMES THE COMPONENT COST

When I hear someone say something like, "In order to make a profit, you need to charge five times the component cost," I see this as a signal that this person has never developed a written business plan. Perhaps a product uses a few resistors, then the component cost is practically zero. Obviously a company building a PC graphic card needs a different pricing strategy. No quick rule replaces the simple logic of knowing your market, customer, and costs.

THE END RESULT

Getting an electronic design project done right and on time wins out over using new technology every time. Careful selection of new technology is important especially when considering learning curves—be prepared to take some lumps. Remember that the smoke and mirrors used in selling technology products become an occupational hazard for otherwise well-intentioned salespeople. Insist that management provide a long-term plan for any new designs. Make sure you generate cost estimates on doing functions now or later. Adding more functions at the first cut of a design delays milestones and adding things later involves product rework—both are real costs. Follow the basic premise of "Keep it simple but do it right the first time." ■

Robert Kondner has more than 20 years of experience designing small products and embedded systems. Many applications have used ARM processors with LCD graphic panels in portable or vehicle environments. You may reach him at sales@index designs.com.

Circuit Cellar, the Magazine for Computer Applications. Reprinted by permission. For subscription information, call (860) 875-2199, subscribe@circuitcellar.com or www.circuitcellar.com/subscribe.htm.