

Chip Design

Tools, Technologies & Methodologies

www.chipdesignmag.com

Max's Chips & Dips - October/November 2009

Rocket Your Way to FPGA Debugging Success

Now engineers can see how an FPGA design behaves in the physical chip—running as it will in-system—while still having access to all of the capabilities and flexibility of a software simulator.

In these hard economic times, everybody is trying to do more with less. The United States Postal Service (USPS) is no exception. Recently, it became necessary to create a new type of mail-processing machine, which would be installed at over 200 USPS mail hubs across America. One solution would have been to create a new device from the ground up at an estimated cost of around \$400,000 per unit. As an alternative, it was decided to refurbish existing mail-processing equipment that had originally been created for quite a different task. These existing units had already been decommissioned and warehoused. Instead of scrapping these machines, they were modified to perform the new task for only \$60,000 per unit, thereby saving the USPS (and the American postal customer) close to \$70 million.

The problem lay in creating a sophisticated control unit that could perform real-time data processing and control operations at blinding speed. This unit also would have to be created in such a way that it could be linked via a hardware interface to the mail-processing machine on one side and via USB 2.0 to a PC running Linux on the other. And all of this had to be achieved in a very tight timeframe.

THE INCREASING DEMAND FOR SPEED AND EFFICIENCY

From watching old films on television, many of us continue to cherish a mental image of our local postmaster casually sorting the day's mail by hand into wooden cubbyholes. This was done in preparation for a mail carrier to deliver it to our homes. Things just don't work this way anymore. Today, the USPS is the third-largest employer in the country behind the Department of Defense and Wal-Mart with more than 34,000 facilities, a fleet of over 220,000 vehicles (the largest civilian fleet in the world), and over 600,000 employees.

Processing an average of around 670 million pieces of mail a day, USPS carriers deliver to more than 149 million residences,

businesses, and post-office boxes. Not surprisingly, performing this task quickly and efficiently demands hard work and dedication from USPS employees backed by the latest and greatest in technological infrastructure.

When you post a letter, it's quickly transported to the nearest major mail hub. As soon as it arrives at the hub, the mail is fed into a machine that captures an image of its handwritten or printed address. This image is transmitted to powerful computers that perform incredibly sophisticated Optical Character Recognition (OCR) in real time. Based on this, additional software determines the destination that the originator of this piece of mail had in mind. It then accesses a national database to "fill in the gaps." The result is translated into a POSTNET barcode that is printed onto the mail as it passes at high speed through the machine. This barcode is subsequently used to guide the mail throughout the remainder of its journey.



Figure 1: The original CSBCS machines are pictured in the warehouse.

Next, the mail is sorted with respect to whichever major hub is closest to its ultimate destination. It is then transported to that hub. In the not-so-distant past, the mail would eventually end up at some main local mail facility. Here, the mail would be fed into a Carrier Sequence Bar Code Sorter (CSBCS), which would read its barcode and sort the mail into the Carrier Route Sequence (see Figure 1). That is, the mail would be sorted based on the order in which it would be delivered by the carriers serving that community.

The old CSBCS machines were around 6 ft. wide, 4 ft. tall, and 12 ft. long. Approximately 4000 of these units were deployed at local facilities around the country. These machines performed their duties admirably for many years. But time moved on and they were eventually superseded by new technological developments. Now, the task of sorting the mail into the Carrier Route Sequence is performed at the central hubs. Each hub is equipped with an advanced Delivery Barcode Sorter (DBCS). The size of these monster devices varies depending on the installation. But they can easily be several hundred feet in length. Following the introduction of the DBCS machines, the original CSBCS units were decommissioned and stored in a warehouse facility in Topeka, KS. Destined for the scrapheap, their future looked bleak until....

WE HAVE A PROBLEM

Although the modern DBCS machines are capable of sorting 40,000 pieces of mail per hour, they can perform this task only if they can actually read the barcode labels that have been sprayed onto the mail. Unfortunately, the barcodes are unreadable around 4% of the time. In some cases, a piece of mail has images and/or too much writing on it, which results in too little space to spray the barcode. Alternatively, the mail may be colored, which means that there isn't enough contrast to read the barcode without error.

This failed mail has to be processed by hand at a rate of only 300 pieces per hour. In addition to being time consuming, prone to error, and—let's face it—not a lot of fun, this hand-sorting has to be performed again and again at the various hubs and local mail facilities. Typically, it has to be done four or five times before the mail reaches its ultimate destination.

To address this issue, it was decided that a new machine should be created. This machine would do the following: process the rejected mail; re-scan the original address and transmit it to the computers that perform the optical character recognition and determine the destination; and apply a sticky label to the mail and spray the POSTNET barcode onto this sticky label. This sounds easy if you say it quickly. But it becomes a little trickier when you discover that this is all occurring while the mail is being transported at 4 meters per second. Initially, the thought was to build this new machine

from the ground up. But early evaluations showed that this would cost around \$400,000 per machine, which would have been far too expensive.

THE "GREEN" SOLUTION

As an alternative, Brent Raney, manager of technology development and applications, and Mike Amato, manager of engineering software management at USPS Headquarters Engineering, championed the idea of retrofitting the existing CSBCS machines. In addition to providing major cost savings to the USPS, this "green" solution would be much kinder to the environment. A team of "in-house" engineers was commissioned to design the upgrade. Meanwhile, the USPS Central Repair Facility in Topeka, KS was tasked with performing the refurbishing effort and applying the upgrade to the old equipment.

The project required a mixture of mechanical, electronic, and software design disciplines. Much of the original CSBCS machines—the feeder, transport belt, optical reader, and sorter—could be reused. A new section was to be added to the existing machine to perform the enhanced functions. This new section was to have its own dedicated electronics and computer for control.

Jerry Pender, electronic engineer, was tasked with the electronic portion of the design. It was to be based on a field-programmable gate array (FPGA) because the FPGA's massively parallel nature could satisfy the requirements of the computationally intensive control and data-processing algorithms in real time. Jerry's colleague, Bill Storey, was to develop the computer control software. "We needed a PC platform computer interface, but didn't want to spend a lot of time and money designing one," says Jerry. Jerry had designed specialized ISA and PCI bus interface cards in the past, but felt that there should be an easier way. He had been toying with the idea of using USB 2.0 as a machine control interface for quite some time. In researching the idea, however, Jerry found that there are a lot of pesky little details involved with USB. "On the surface, USB seemed so simple. But once you got into it, it was so much more involved," he notes. Jerry knew that if he could somehow couple USB 2.0 with an FPGA, it would be the ultimate implementation platform and a match made in heaven.

Jerry started to research the problem and quickly discovered Opal Kelly. Opal Kelly's FPGA Integration Modules connect a user's hardware design implemented in an FPGA directly to the user's software implemented on a PC. Opal Kelly handles all of those pesky little USB details, such as the enumeration process, low-level drivers, and bus timeouts. They even provide a control-panel software package called FrontPanel. It allows testing of the hardware design (to some extent) at one's desk without using the actual application software. It also allows

the engineer to build some level of confidence before putting it into the actual application. Opal Kelly even used Xilinx platform FPGAs, which Jerry had been working with for years, so there was no learning curve. “I realized within seconds that this is exactly what I had been searching for. If I had written a wish list of everything I wanted, I could not have done better,” comments Jerry.

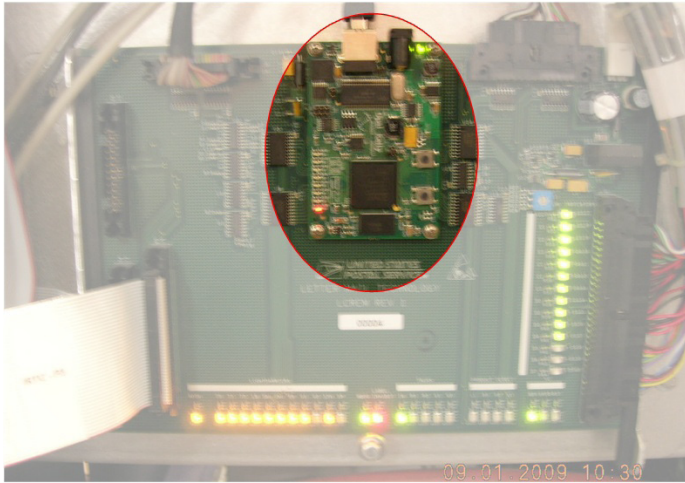


Figure 2: Here is a close-up of the Opal Kelly XEM3050 module in the new machine.

Jerry used Opal Kelly’s XEM3050 module in his interface to the PC and Bill used Linux Fedora for the application software operating system (see Figure 2). The system was required to track mail as it moved through the various processes on a transport running at 4 meters per second. Each piece of mail needed to be monitored in real time so that the correct mail address was used.

Linux is not a real-time operating system and even high-speed USB 2.0 has inherent latencies. So how is this combination capable of performing real-time process control? “The answer lies in the design of the application,” says Jerry. “We made sure that all incoming events are time-stamped (or in this case, position-stamped) in the FPGA before they are passed along to the control software as inputs. All outgoing data and control signals from the control software are sent to the FPGA in advance, where external real-time signals are then used to trigger the functions at the appropriate time.”

“The only requirement is that the CPU and control software be fast enough to process these events without missing any deadlines,” notes Jerry. “Our transport moves at the rate of 10 mailpieces per second and there can be up to 20 mailpieces in the processing queue at any given time. Bill has run timing-analysis software along with his control application software, and has found the processing speed to be significantly more than adequate for the application.” The FPGA also allowed the team to build a simple mailpiece image-capture system, which was transferred to the Linux system via the USB interface.

One really useful aspect of the Opal Kelly module is that it allows both data and control signals to pass into and out of the computer—all over a single twisted pair of USB wires. As Jerry points out, “We never even had to take the cover off of the computer.” Opal Kelly also provides simple mechanisms to monitor and/or control signals, monitor and/or control events, and receive and/or output data.



Figure 3: This picture shows the team standing in front of one of the retrofitted machines.

Jerry continues, “Without this system, we would have had to purchase more expensive hardware and develop two special-purpose boards.” Jerry estimates that going with the Opal Kelly solution reduced design and debugging time by six months. The team commenced the project in September 2008 and had a fully working prototype up and running by January 2009 (see Figure 3). At a cost of only \$60,000 to retrofit an existing CSBCS machine, this solution—when applied to the 200 units that will be deployed in USPS mail hubs across America—will save the USPS (and the American postal customer) close to \$70 million. (For those who are interested, the Opal Kelly XEM3050 module costs just \$750 and the FrontPanel software, which runs on the Linux, Mac, and Windows operating systems, is included for free.)

As Jerry says, “There is beauty in simplicity. Using the Opal Kelly module made things really, really simple. It has been a pleasure (even fun) to work with this solution! (Whoops, please don’t let my boss hear that!)” ♦

Clive (Max) Maxfield is author of *Bebop to the Boolean Boogie (An Unconventional Guide to Electronics)* and *The Design Warrior’s Guide to FPGAs (Devices, Tools, and Flows)*. Max also is the co-author of *How Computers Do Math*, featuring the pedagogical and phantasmagorical virtual DIY Calculator (www.DIYCalculator.com).

