CIRCUIT CELLAR ONLINE

LEARNING THE ROPES

Ingo Cyliax

Virtex Proto Board

Although Ingo had originally planned to follow up his last article about multipliers, he ran into a little problem. OK, so it was a big problem. But, that's to our benefit.

I originally planned to follow my last piece with an article about how to design with multipliers. Well, to shorten the story, the software I intended to use in designing some of the multiplier examples didn't work.

The longer version of the story is that, because I was working on a project, I needed to upgrade my FPGA design software to the latest and greatest version. Of course, because it runs under Windows, this doesn't always work the way you think it will. Such was the case with the new installation, so I decided to downgrade to the original version. Unfortunately, it doesn't work at all now. At this point, I needed to uninstall both versions by removing all traces of *.dlls*, executables, environment variables, and registry entries, or just resort to reinstalling Windows. I'm sure you've been there at one time or another. Someday they'll bring out these design tools under Linux. At least I can dream.

My design environment under Windows is a mess, but the Linux environment on my laptop still works well, enabling me to write this article. This month, I'll talk about the Virtex prototyping board I'm using for the project that got me in trouble—the Virtual Workbench 300 (VW-300) from the Virtual Computer Corporation (VCC).

THE BOARD

The board is a prototyping board for the Xilinx Virtex series FPGA. Virtex is Xilinx's high-end FPGA, boasting densities of up to one million gates (more in the Virtex-E). Remind me to write a piece about how to compute realistic gate densities in FPGAs that actually mean something.

Virtex has some nice architectural features that make it suitable for large system-on-a-chip designs that include 32-bit processor cores, peripherals, and high-performance digital signal processing. There are several types of memory from SelectRAM that can be used for small register files (i.e., BlockRAM), which are large 4-Kb

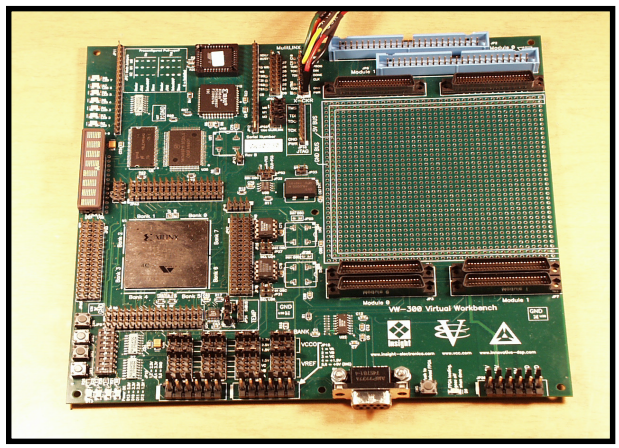

Photo 1—*Look at all this neat stuff on such a small board.*

Standard	Input Ref. Voltage	Output Source Voltage	Termination Voltage
LVTTL	N/A	3.3	N/A
LVCMOS	N/A	2.5	N/A
PCI	N/A	3.3	N/A
GTL	0.8	N/A	1.2
GTL+	1.0	N/A	1.5
HSTL Class I	0.75	1.5	1.5
HSTL Class III	0.75	1.5	1.5
HSTL Class IV	0.75	1.5	1.5
SSTL3 Class I/II	1.5	3.3	1.5
SSTL2 Class I/II	1.125	2.5	1.125
CTT	1.5	3.3	1.5
AGP	1.32	3.3	N/A

Table 1—*Here is a list of all the I/O interfaces that can be supported by Virtex. Each bank's V_{IO} and V_{REF} power supplies can be wired to various voltages to help implement these.*

RAM modules available on-chip. BlockRAMs are highly flexible because they can be configured as either single- or dual-ported memory with either 1 or 16-bit wide ports. Other than that, they are of the look-up table-based CLB architecture, with plenty of routing resources to connect them with fast carry support between adjacent CLBs.

The I/O on Virtex chips is perhaps the best feature. There are several banks of I/O, each bank powered by its own I/O power supply and input reference voltage. With the appropriate power supplies, various signaling standards can be implemented. Table 1 shows the supported types.

Of course, the inputs are also 5-V tolerant if it's selected and have programmable pull up/pull down and weak keepers.

To make interfacing even easier, Virtex has several delay locked loops (DLL). These can be used to match internal clock signals to external clocks in order to reduce the effects of on/off chip latencies at high clock rates.

NO RE-FLOW OVEN?

Virtex FPGAs are available in BGA packages. BGA packages are the package of choice for large chips these days. Of course, like many new package styles, they are harder to mount when prototyping or building one-offs. BGAs have small solder balls on the bottom of the chip that match up with small solder pads on the PCB. The BGAs are placed on the PCB and then need to be heated in an oven for the solder balls to melt and bond with the solder pads. Of course everyone has a re-flow oven these days.

Because I don't have a re-flow oven handy and didn't have enough time to track down a facility that could handle mounting these packages, I had to buy a prototype board for this project. The project, incidentally, was demonstrating a 32-bit processor core I developed for my day job. I needed a Virtex board with a large enough chip and some external memory pre-wired, along with a prototyping area for all the extra stuff.

After searching Optimagic's web site, which is great for FPGA resources, I found that the VW-300 fit the bill, so I ordered one.

Besides the Virtex chip, the VW-300 has a lot of memory, a clock, and connector resources. In addition, there are LEDs, switches, push buttons, 8-digit alphanumeric LED display, and other neat stuff all on an 5.5″ × 5.5″ board (see Photo 1).

There are three memories on the VW-300. A 256K × 8-bit flash memory is used to store startup configuration for the FPGA. From the factory, it has a demo that scrolls continual advertising over the alphanumeric display. It's nice to have something running so you can make sure everything is OK (after you've taken it out of the box and built a power supply). The flash memory can be reprogrammed by removing it from its socket (it's PLCC footprint flash memory) and programming in a programmer. A CPLD is used to interface the flash memory to the Virtex part to perform Xilinx's SelectMAP partial reconfiguration protocol. Because the CPLD is flash memory-based, I suppose it could be reprogrammed to perform other functions.

BURST MODE

Besides the flash memory, the VW-300 also has a 256K × 18-bit synchronous burst mode fast static RAM (SBRAM). This is a memory module normally used for off-chip L2 processor cache and supports burst mode operations. The cycle time of the memory installed is 6 ns (166 MHz), which means it can transfer at a peak rate of over 332 MBps. Because the SRAM is fast but not too large, an 80-ns 1M x 16 x 4 SDRAM is also included onboard.

The SRAM and DRAM memories are devices attached to the FPGA. By itself, the Virtex does nothing special with the memories. You would have to design in a SRAM or DRAM controller into the design loaded on the FPGA in order to use the memory. In this project, you can use the SDRAM as traditional program and data memory, and the SRAM as fast buffer memory or cache memory. The Xilinx web site has cores that can be used to control burst mode SRAM and SDRAM in a Virtex design.

Because the Virtex also has on-chip memory block and small look-up table-based register files, you can use a full spectrum of memory hierarchy—from small single- and dual-ported register files (SelectRAM) running at sub-nanosecond access times, to larger memory block (BLOCKRAM) running at less than 5-ns access times. To get to off-chip memory, you have to figure the latency it takes to get in and out of the chip and protocol issues, in addition to the access times of the memory.

For example, to access SBRAM you have to send the address and wait for the data to burst out of the memory sequentially. Although the peak burst

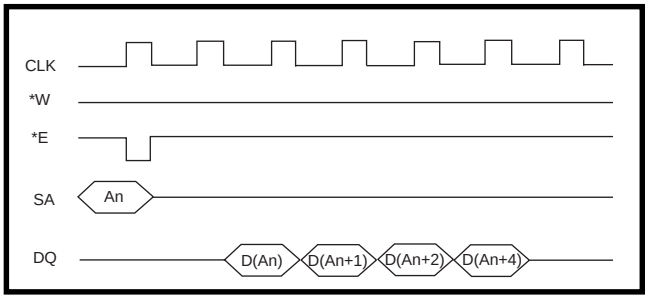

Figure 1—*Here you can see the timing diagrams for SBRAM access.*

rate is fast at 166 MHz, the total access time to get four words is at least five cycles. Figure 1 shows timing diagrams for SBRAM access. In contrast, BlockRAM works as true random access memory, not in burst mode. In any case, with this board you have all the options covered.

MORE OPTIONS

The VW-300 also has several clocking options. It has two Dallas Semiconductor DS1073 econo-oscillators (100 MHz and 66 MHz). These can be set up to divide the oscillator frequency by two, which is the default. In addition to these oscillators, the board has a Cypress ICD2053B adjustable oscillator, which can be programmed using a serial protocol either from the FPGA or externally. By default, this oscillator runs at 16 MHz. As if this isn't enough, there is also an extra oscillator site where you can install your own oscillator module.

Another neat feature is the temperature sensor module. This is a Max1617 die temperature sensor. The Virtex provides a diode-based die temperature output that works with this sensor module. The module converts the temperature reading and lets you read the temperature through a SMBUS interface, either by the FPGA or externally.

Measuring the die temperature is useful for several reasons. Because FPGAs are user programmable, computing power consumption (and power dissipation) is complex. During system testing, you can use the die temperature, and by knowing the ambient air temperature as well as heat transfer coefficients for the package and current air humidity, you can estimate the power consumption and power dissipation.

Also, because this type of Virtex has so many flip-flops and internal tristate buffers, which can cause fights if not properly designed, you can actually overheat the chip at normal operating temperatures. A temperature sensor can be used to detect these conditions and shutdown the chip if the temperature exceeds safe levels. I suppose a boring application of a temperature sensor might be to drive a variable speed fan to cool the chip optimally.

If you're building systems on a chip, you might want to use serial communication channels to talk with other computers or a terminal. This board includes a TTL to RS-232 level adapter and a DE-9 connector to perform this. For example, this project uses an RS-232 port for a background debugger type interface.

Building a UART in FPGA is not hard. I used to teach students in undergraduate hardware labs to do this. I'll cover designing UARTs in a later article.

If you like switches and lights, it has those as well. There is an 8-position DIP switch and eight surface-mount LEDs you can read and light up with the FPGA. There are also four pushbutton switches. However, the neatest peripheral on this board is the Infineon IPD2133 8-character 5 × 7 dot matrix aphanumeric display.

To the FPGA, the display looks like a small SRAM device. There is an address bus and a data bus. The simplest way to use it is to address one of the eight digits (address 0–7) and write a 7-bit ASCII code to it. The chip converts the ASCII code using an internal character ROM and displays it on the 5 × 7 array for that digit. Not bad. Figure 2 shows a block diagram of the internal display.

You can use this display to show a hexadecimal representation of internal registers, assuming you add logic to your design to use the display. You can also use it to provide a small scrolling text console for your project.

Besides onboard peripherals, there are a variety of headers and connectors that can be used to interface external modules, devices, and logic analyzer inputs to the Virtex on this board. In particular, there are several mezzanine connectors for third party CODEC modules from Insight, as well as A/D and D/A Omnibus modules from Innovative Integration. All of the I/Os are documented in the user manual, and pin configuration files for the FPGA can be downloaded from VCC's web site.

There are several configurations that are supported on this board for the Virtex chip. I already mentioned the flash PROM for doing SelectMAP configuration, and there are jumpers and headers for serial PROM, JTAG, Xchecker, and MultiLINX configuration. The manual, schematics, and datasheet for some of the components are available on the web site. After you have registered for your board, everything you'd expect from a prototyping board opens up to you.

UPON ARRIVAL

After the board arrived, it became obvious that I was missing some-

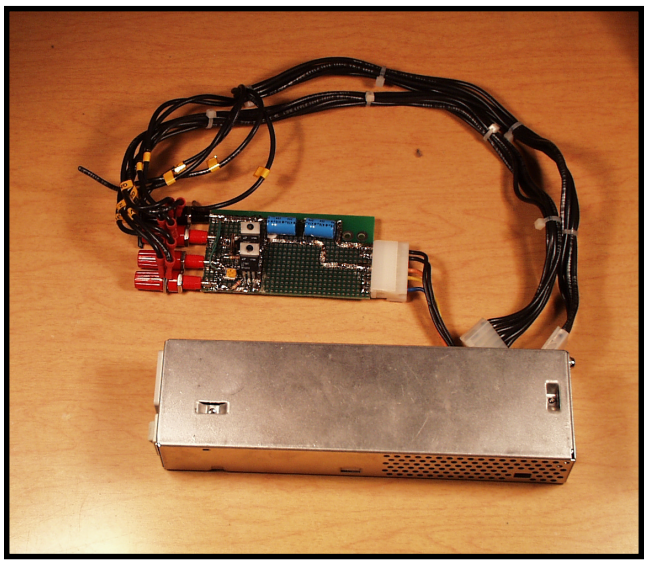

Photo 2—*After the voltages were adjusted and everything was hooked up, I was ready to try it out.*

thing. It turns out that the Virtex chip is a 2.5-V core logic device. That means all of the internal logic on this chip operates from a 2.5-V power supply. Virtex-E FPGA core actually runs at 1.8 V. Furthermore, most of the memories and peripherals work at 3.3 V and some, like the RS-232 converter, run from a 5-V supply. However, there are no onboard regulators. VCC does have a matching power supply available for this board that takes care of it all.

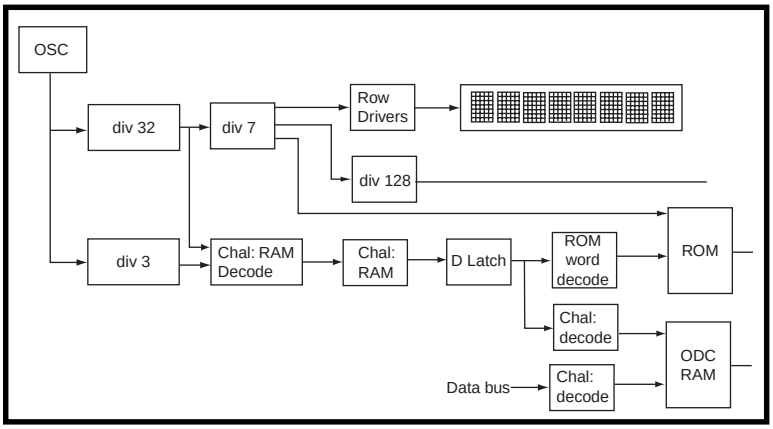

Figure 2—*The internal display has an address bus and data bus and provides you with many options for use.*

Building power supplies isn't rocket science (for the most part), so I figured it shouldn't be a big deal. Especially, because voltage regulators don't come in BGA packages, and I can solder these parts myself. Taking a quick survey however, I discovered that 3.3-V regulator ICs are readily available and 2.5-V regulators are harder to come by. I settled on using Linear Technologies' LT-1587. These devices are available in varied fixed voltages, as well as an adjustable version. I chose the adjustable version that is available in three terminal TO-220 packages and quickly cobbled up a power supply on a perforated circuit board that matches the schematics in Figure 3. Photo 2 shows what this looks like.

Because the V_{IN}/V_{OUT} differential has to be between 1.5 and 5.75 V, I chose a small 5-V switcher that I scrounged up from an old Macintosh. After carefully adjusting the two voltages (3.3 and 2.5 V), I hooked up everything you see in Photo 2 and tried it for the first time. It was nice to see the demo design come up the first time. The compiled project was downloaded through an Xchecker cable and everything came up as expected with the board.

Although I'm typically a strong advocate of building your own stuff, it's nice to be able to use off-the-shelf prototyping boards. Usually when I build my own boards, I spend a significant amount of time debugging my aspect of the design. In this case, it was definitely a timesaver, especially because it will probably take several attempts to fine-tune any BGA-based home assembly techniques. I did figure out that my kitchen stove achieves re-flow temperatures in "clean" mode. Apparently the normal thermostatic control is disabled in this mode, and the oven runs up to its maximum temperature. At this point, however, I'm hesitant to try a re-flow with a $300 FPGA in my kitchen oven. If I take on such a task in the future, I'll be sure to let you know whether or not I'm successful.

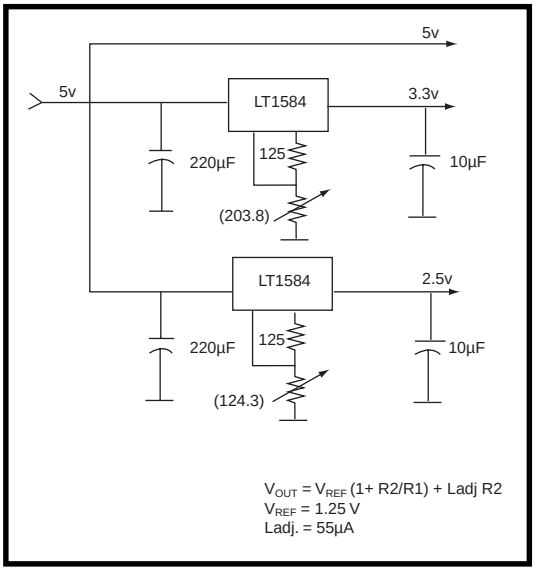

Figure 3—*Here you can see a schematic of the power supply I quickly threw together.*

Ingo Cyliax is the Sr. Hardware Engineer at Derivation Systems Inc. (DSI) where he designs and builds embedded systems and hardware components. DSI is the leader in formally synthesized FPGA cores and specializes in embedded Java technology. Ingo has been writing on various topics ranging from real-time operating systems to nuts and bolts hardware issues for several years.

SOURCES

Virtual Workbench 300 (VW-300)
Virtual Computer Corporation
(818) 342-8294
Fax: (818) 342-0240
www.vcc.com

OptiMagic's Programmable Logic Jump Station
Optimagic
(831) 687-0415
Fax: (408) 701-7007
www.optimagic.com