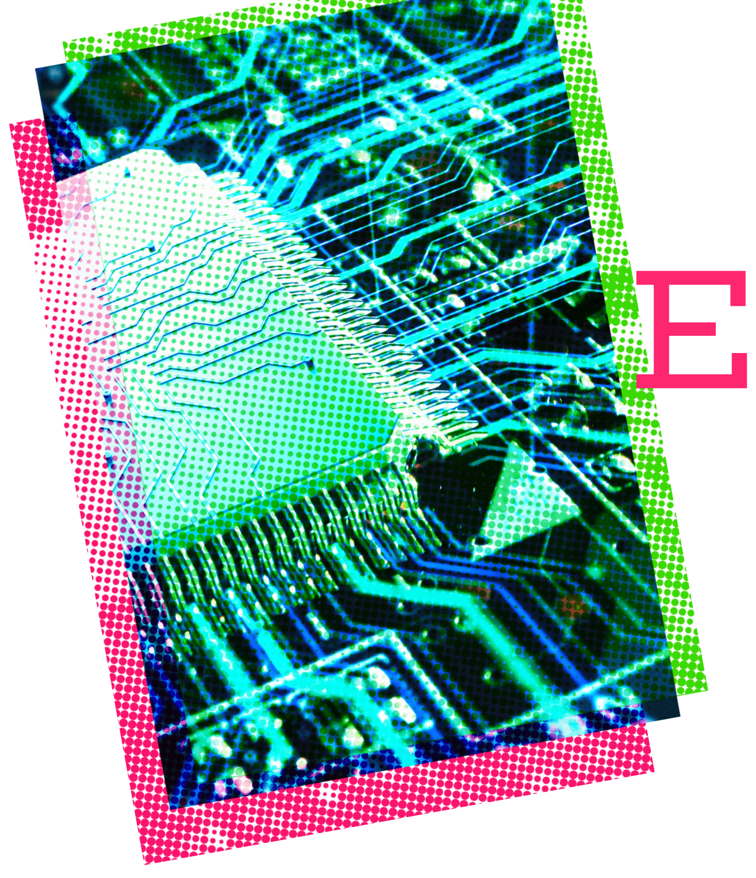

Digression into bhyve

BY CHRISTOPHER R. BOWMAN

In the previous two columns we talked about the Digilent Arty Z7-20 with which I've been experimenting. I find this an interesting board because not only can you toggle primary pins to interface to the outside world, but you can build your own circuits into the fabric and interface them to the processor. To do all this, however, you'll need to use Xilinx/AMDs software to configure and program the chip. Xilinx calls this tool suit Vivado and you can download a version which works with the Zynq chips free of charge from their website.

So, what's the downside? There has to be another shoe waiting to drop somewhere, right? There are two really. First, Vivado only comes in versions for Windows and Linux. Second, the download alone is currently 110GB. For years, I ran the Linux version under VMWare on my Mac and it worked pretty well. But I ended up with several different versions of my virtual machine, each with different versions of the software installed. These VMs were large, starting at 30 gigs or so in the early days and growing most recently to 75 gigs in my most recent installs. Make a couple of versions of these on a couple of machines, and, well, that starts to add up, and I have trouble keeping track of which one is the most recent.

I can use the magic of ZFS to checkpoint my vendor software installs.

I decided I wanted to simplify and centralize. I want to store all the originals of vendor tools I downloaded on my networked home file server, which runs FreeBSD, naturally, and has terabytes of ZFS space. After all, I can't depend on my vendor to continue to offer these past versions after they've moved to newer versions. That's understandable, but as a hobbyist, my work doesn't always proceed at their pace, so I want to make sure I keep the original tool downloads. I wanted my home directory and my working files and projects to be stored on my network server, so they are available from all my machines and backed up like any other data on my file server. I wanted all the unzipped and installed versions of the vendor software to be stored on my file server not in my VMs. This way I can use the magic of ZFS to checkpoint my vendor software installs. My VMs are a light weight install of Linux, any Linux with which I want to experiment. If I need to upgrade or create a new VM, I know there is no work on there, it's all on my file server, so I just need a new basic install and everything else is all still on my file server. I don't have to copy over or reinstall the vendor tools. Since recent upgrades to the compute environment at my home have included a FreeBSD machine with 16 cores and 128 Gigabytes of memory, I want to run these tools on that machine in anticipation of the point in time where my designs get big enough that they take hours to synthesize. As a bonus, I get a setup where someone else looking to replicate my work needs only a single FreeBSD machine.

There seem to be two basic approaches at this point. I could try to get the installer to run under FreeBSD's Linux emulation and run these tools on my FreeBSD box natively without a Linux VM. That would be AWESOME! But I'm not sure it could be done or what problems I would run into or how long that might take me to figure out, and I was in the middle of some projects. This really seems like the best approach, and I'd love to hear if anyone has this working or wants to try it, but I chose an approach that I thought would involve less effort and be more likely to work. I'd heard a lot about bhyve and decided to investigate that. If I get a VM running a version of Linux supported by Vivado natively, I figured that would be the easiest. It only took me an evening or so of reading and another of experimentation and I was shockingly up and running.

I started with the FreeBSD handbook. It's really an amazing resource and kudos to everyone who helped make it as brilliant as it is.

I started with the FreeBSD handbook. It's really an amazing resource and kudos to everyone who helped make it as brilliant as it is. Chapter 24.6. FreeBSD as a Host with bhyve has a really good introduction to using FreeBSD as a host operating system. For the most part, I cobbled together a setup from there and a few other places on the internet. To setup, I created the basic host setup for networking:

```
# kldload vmm

# ifconfig tap0 create up
# ifconfig bridge0 create
# ifconfig bridge0 addm igb0 addm tap0
# ifconfig bridge0 up
```

I'm using the instructions in 24.6.5. Graphical UEFI Framebuffer for bhyve Guests since I don't want to muck around with setting up grub. Using the UEFI Framebuffer also allows me to export the Linux display via vnc. I can connect from my FreeBSD host if I'm using that or from any other machine on my network. Though perhaps I should think a bit more about security.

I downloaded an ISO version of CentOS from before RedHat killed it, and I use the following VM configuration to do the install:

```
# bhyve -c 4 -m 32G -w -H \
        -s 0,hostbridge \
        -s 3,ahci-cd,/u1/ISOs/CentOS/CentOS-7-x86_64-DVD-2009.iso \
        -s 4,ahci-hd,/dev/zvol/zroot/vms/centos7 \
        -s 5,virtio-net,tap0 \
        -s 29,fbuf,tcp=0.0.0.0:5900,w=1920,h=1200,wait \
        -s 30,xhci,tablet \
        -s 31,lpc -l com1,stdio \
        -l bootrom,/usr/local/share/uefi-firmware/BHYVE_UEFI.fd \
        vm0
```

With this I get a 4 core 32-gig virtual machine with the ISO, `/u1/ISOs/CentOS/CentOS-7-x86_64-DVD-2009.iso` mounted in the guest as a CDROM. With this I can run a standard GUI install of Linux which is pretty straight forward. I'm running this on a bare zvol: `/dev/zvol/zroot/vms/centos7`. I did this so I could use ZFS snapshots to snapshot my VM any time I want (initially right after a clean install) so I can roll back at any time. If I put the VM virtual disk on a ZFS files system, my snapshot would apply to anything else on that file filesystem, not just the VM virtual disk image. I also heard using bare zvols can be faster. In hindsight, I could have used a single data set per VM virtual disk image and used a file on that data set as the VM virtual disk image rather than a zvol. If there is one VM per data set, snapshots still apply to just the VM. I'm not sure which is a better approach, and subsequent reading has called into question the zvol vs. file backing speed aspect. My tool run times aren't yet long enough that I'm concerned about eking out every last ounce of performance. I've also heard the nvme devices can be faster than the ahci-hd device for guest OS that support them, but I haven't experimented there. If it becomes an issue, it's relatively easy to create a new VM now that my data and install doesn't reside on the VM.

Now I have a fairly generic CentOS VM with a Vivado installation.

Once I installed CentOS, I configured it to NFS to mount my home directory from my FreeBSD machine, and I ran the Xilinx Vivado tool installation. It's a gui install and pretty straightforward. I just make sure I'm installing to an NFS mounted directory. Given the volume of file operations, I was expecting this to be pretty painful, but it went surprisingly quickly for a tool that ends up at 66 Gigabytes installed. Granted I have a very fast local network. I don't expect to edit the tool install, but I created a snapshot — this too for peace of mind. I don't want to have to run that install again.

When I applied for a Vivado license, I copied the ethernet MAC address that my Linux guest reports. It seems to be stable across reboots, but I'd like to figure out how to configure this so that I can be sure that the MAC address on my VM always matches my Vivado license MAC address.

Now I have a fairly generic CentOS VM with a Vivado installation that I can access via VNC from any machine on my local network. At this point, I also installed a 15-year-old Linux version of *Civilization: Call to Power* to play while my FPGA builds are compiling. I was shocked at how well it works (and how addicted I am to it.)

While most of this works really well, there are a couple differences from my initial setup using VMWare on a Mac. First VMWare supports pass through to the host filesystems. The NFS mounts accomplish basically the same thing, and I can't notice much speed penalty, but I have to configure this in Linux instead of the VMWare gui. Not a big difference--I'm comfortable with that. Where I do find myself wishing for the VMWare solution, is copy and paste. If I select something in the Linux Guest gui, I can't easily copy and paste that to the machine on which I'm running the VNC viewer. This is occasionally a sore point, but since the user filesystems I'm using under Linux are all NFS mounted from FreeBSD, I can edit those files live just as easily from FreeBSD (or any other machine that's NFS mounting them). As a result, I do almost no editing or work under Linux. Mostly, I just switch to the Linux VNC session window to run the Vivado compiles, and I do everything else outside that. It works pretty well.

In the next column, we will start to use Vivado to build our first circuit.

If you've got feedback, complaints or flames on any of this I'd love to hear from you. You can contact me at articles@ChrisBowman.com.

CHRISTOPHER R. BOWMAN first used BSD back in 1989 on a VAX 11/785 while working 2 floors below ground level at the Johns Hopkins University Applied Physics Laboratory. He later used FreeBSD in the mid 90's to design his first 2 Micron CMOS chip at the University of Maryland. He's been a FreeBSD user ever since and is interested in hardware design and the software that drives it. He has worked in the semiconductor design automation industry for the last 20 years.