

# Experiences Building a Coherent Shared Memory Machine: RAMP-White

Hari Angepat and Derek Chiou

Electrical and Computer Engineering

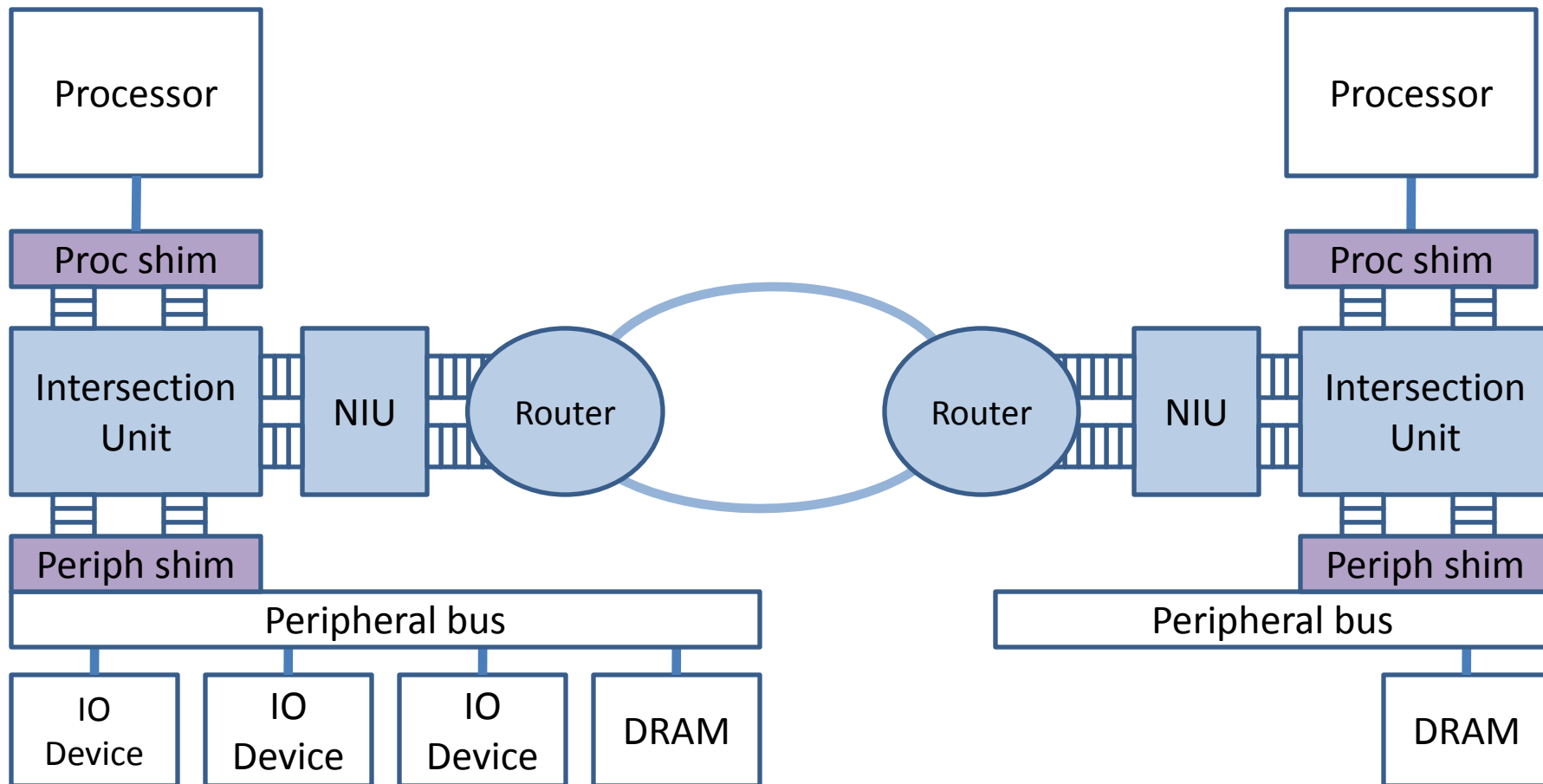
University of Texas at Austin

IBM Technical Contact: Volker Strumpfen, IBM Austin Research Laboratory

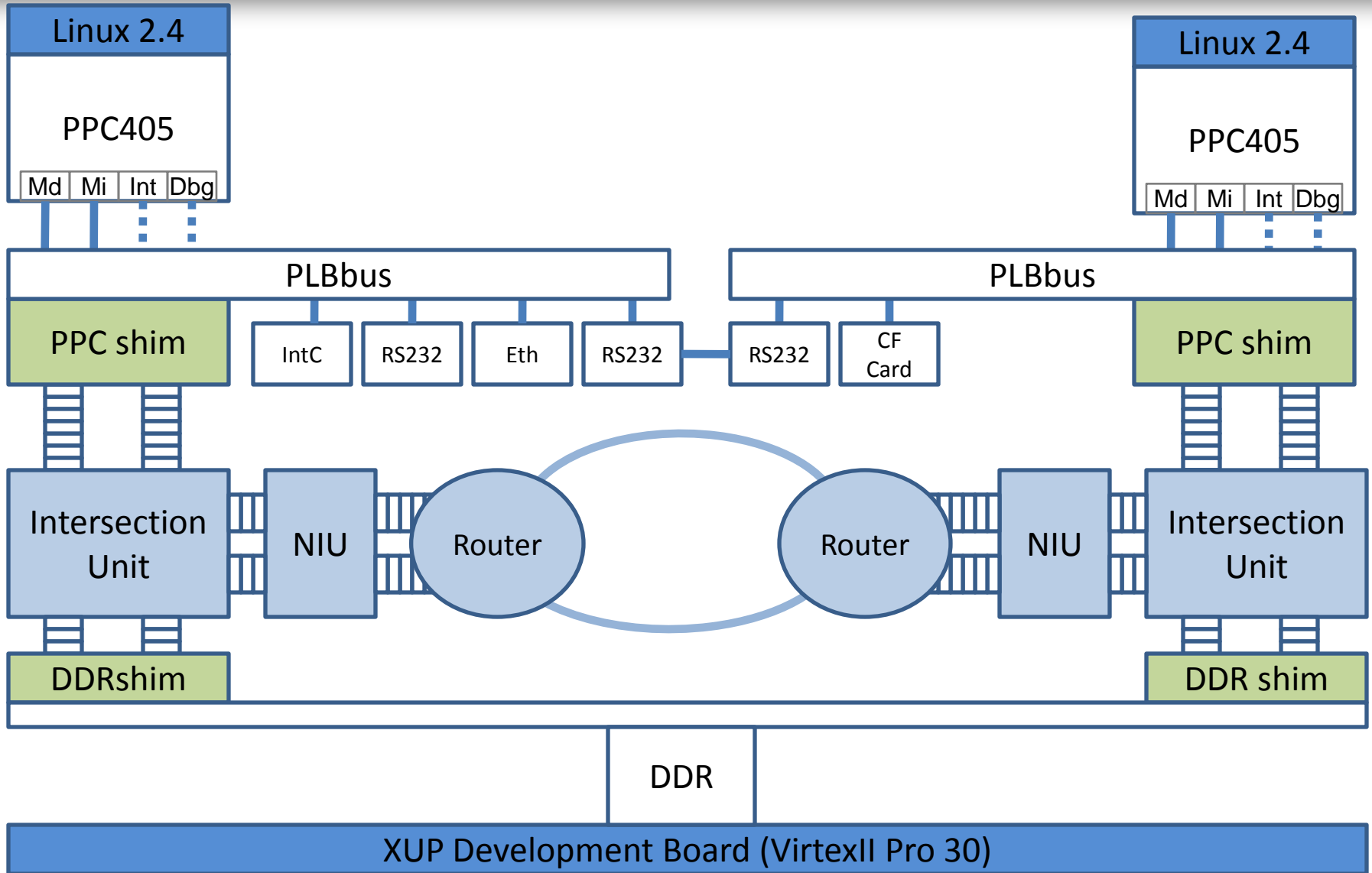
# RAMP White

- Cache coherent shared memory multiprocessor emulator
- Intended to scale to 1024 cores
- Implemented using FPGAs
- Single image operating system support
- Support for existing programming models
  - Programming libraries
  - Legacy applications

# RAMP White



# Prototype 1



# Software Stack

- Limitations of multi-image kernels
  - Requires custom porting of software framework
  - Restricted access to peripheral devices
- Kernel Proxy
  - Still requires custom porting of frameworks
- SMP Linux
- Plan 9

- Issues:
  - PPC405 hard macro
    - Interrupt pins only mechanism of invoking external control
  - Boot from reset
    - MP interrupt controller (IPI's, broadcast)
  - Caches
    - Incoherent, externally accessible via interrupt handler
  - TLB management
    - Software managed TLB must be explicitly kept consistent
- Verification of modified HW/SW components
  - Non-trivial task to get correct, with a number of performance limitations

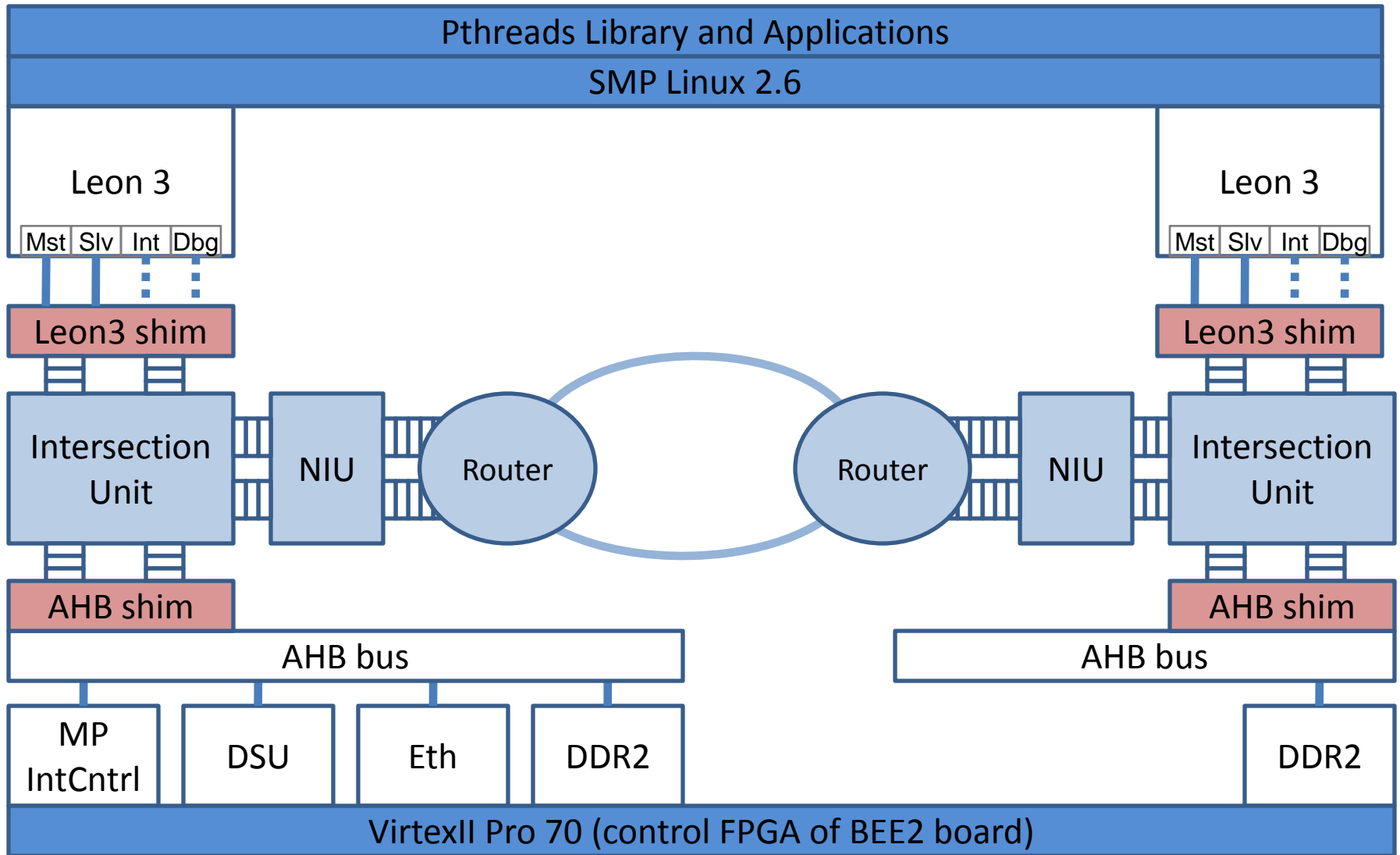
# Plan9

- Smaller, lower complexity operating system
- Allows flexible sharing of physical resources
  - memory, ethernet, disk, cpu servers
- Allows easy task execution/debugging on remote cores
- Leverage resurgent interest in using Plan9 for HPC (IBM port to Bluegene)
  - Porting to PPC405 RAMP-White node heavily assisted by Eric Van Hensbergen

# Prototype 2

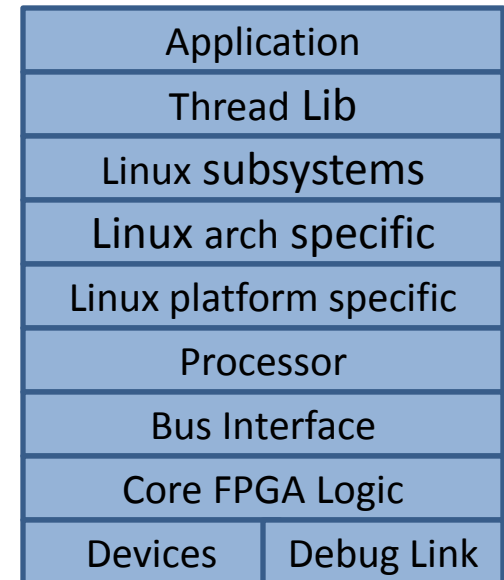
- Gaisler Research open source IP library
  - GRLIB Components {serial, ethernet, ddr, jtag, etc}
  - Supports their Sparc soft-core processor (Leon3)
  - **Maintain a Linux SMP 2.6 port for Leon3**
- Migration to BEE2 Development Platform
  - 5 \* VirtexII Pro70, inter-FPGA links
  - JTAG programmed of bitstream
  - Ethernet based debug for kernel loading/debug

# Prototype 2



# Debugging HW/SW Codesign Platforms

- Rapid prototyping VS verification effort
  - System-level behavior often only seen in HW
  - Software simulation speeds limit coverage
- Codesigned HW/SW
  - Verification stack spans multiple levels
- Hardware Development
  - Bluespec HDL
  - Chipscope debugging
  - Multiple Platform Configurations



# Future Work

- Debugging scalable support multi-FPGA links
- Development of a simple micro-coded coherency engine
- Expand cache hierarchy with soft-core cache models
- Complete Plan9 port with support for shared memory between nodes

# Questions?

- Demo Videos:
  - Prototype dual PPC405 on XUP
  - Prototype dual Leon3 on BEE2

# Dual PPC405 on XUP

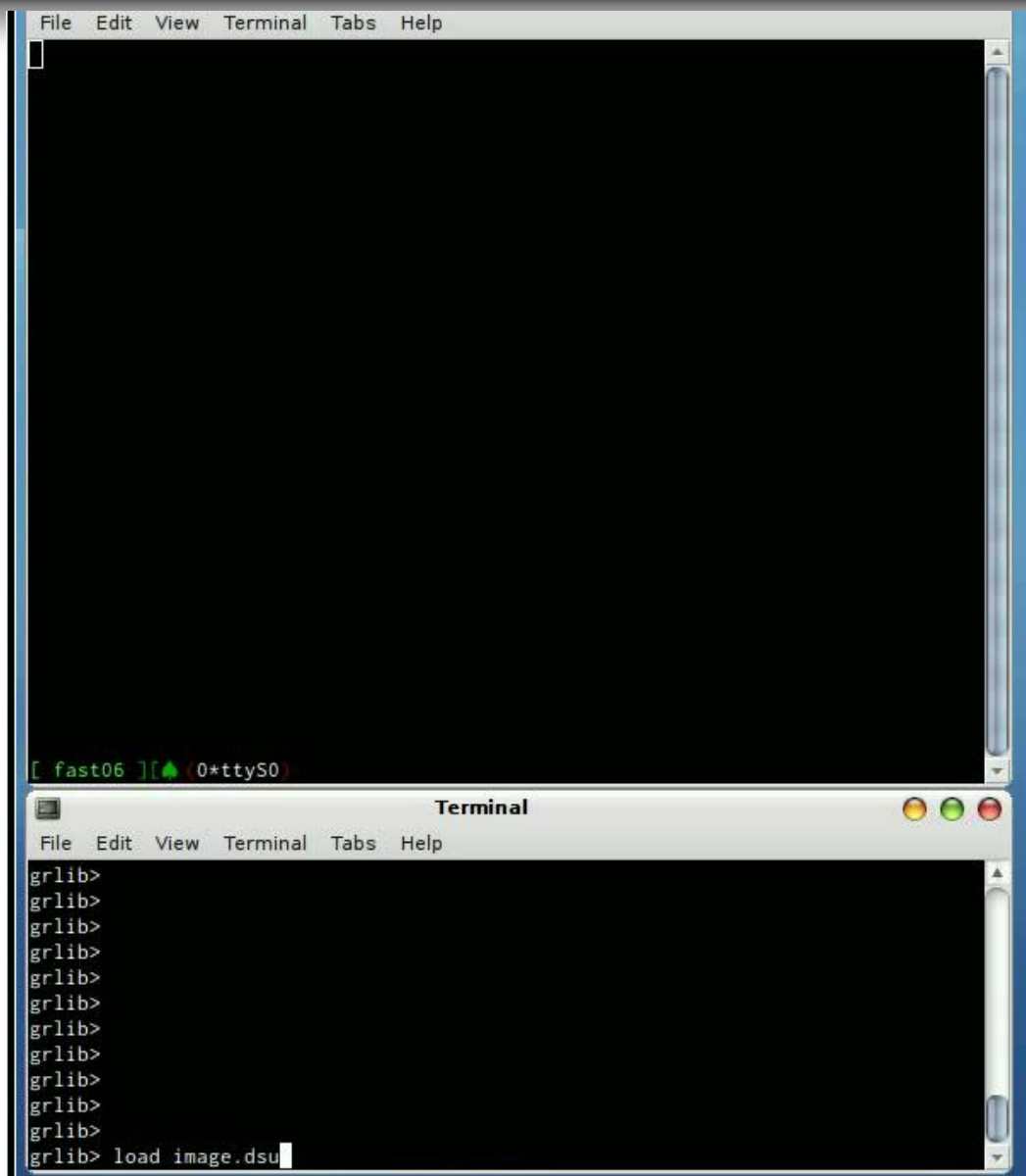
```
- angepat@wolverine: ~/bin
File Edit View Terminal Tabs Help
{flash}[root]> ls
PPC0 devmem.elf p0.elf sw
{flash}[root]>

- angepat@wolverine: ~
File Edit View Terminal Tabs Help
{ppc2}[root]> ls
PPC1 devmem.elf p1.elf sw
{ppc2}[root]>
```

[ wolverine ] [ (0\*ttyS0) ] [ 30/10 9:14 ]

[ flash ] [ (0\*ttyS1) ] [ 30/10 7:14 ]

# Dual Leon3 on BEE2



```
File Edit View Terminal Tabs Help  
[ fast06 ] [0*ttyS0]
```

```
Terminal  
File Edit View Terminal Tabs Help  
glibc>  
glibc>  
glibc>  
glibc>  
glibc>  
glibc>  
glibc>  
glibc>  
glibc>  
glibc>  
glibc>  
glibc>  
glibc>  
glibc> load image.dsu
```

# Questions?