Linux without a Boot Loader?

Greg Ungerer
gerg@snapgear.com

Why bother?

- One less package to support
- Reduce startup times
- Save on flash space
- Reduce hardware support code duplication
- Make Linux *the* boot loader!

Boot Loaders

- U-boot, redboot, blob, etc
- Initialize and setup hardware
- Boot linux kernel (or any other OS)
- Recover from bad flash contents
- Update flash
- Debug hardware?

Does Linux need a boot loader?

- Yes (currently :-)
- No power up hardware initialization
- No memory initialization code
- No startup vectors or CPU setup

Can we modify Linux to boot?

- Yes!
- Surprisingly easy to do
- Changes are very hardware specific
- Mostly done in assembler start up code

Code Changes Needed

- CPU initialization
- DRAM initialization
- BUS initialization
- Move kernel to RAM*

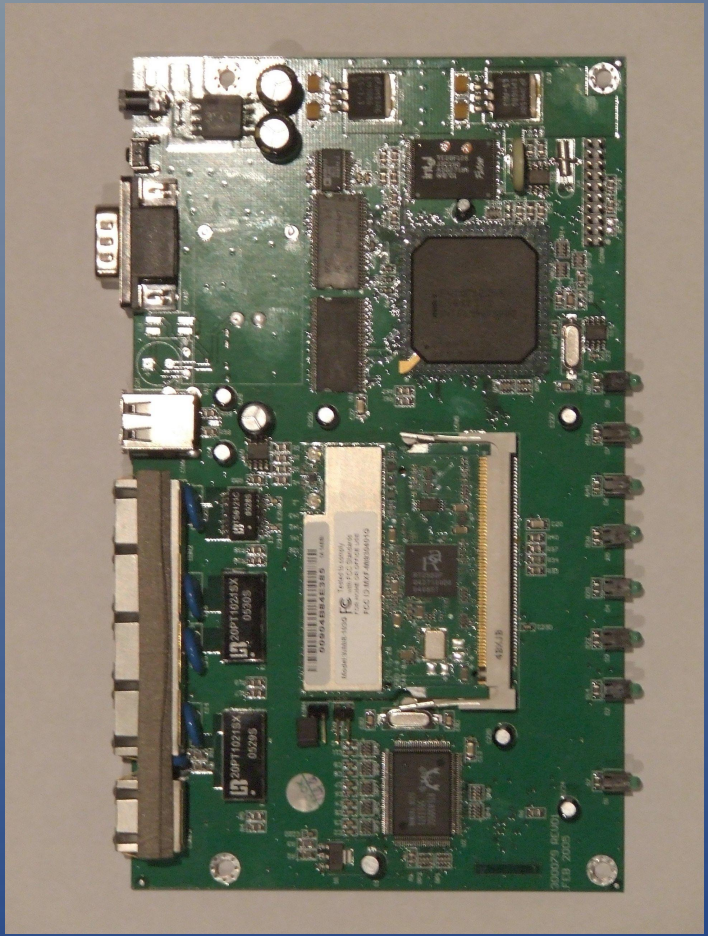
uClinux ColdFire 5272

- Coldfire 5272 66MHz CPU
- 2MB NOR flash
- 8MB DRAM
- 2 ethernet interfaces
- 1 serial port

uClinux ColdFire 5272

- XIP kernel, 1237476 bytes
- ROMfs (XIP apps), 505856 bytes
- m5272sim.h PLATFORM_SETUP macro
- ~50 lines of assembler added

IXP425 XSCALE (ARM)

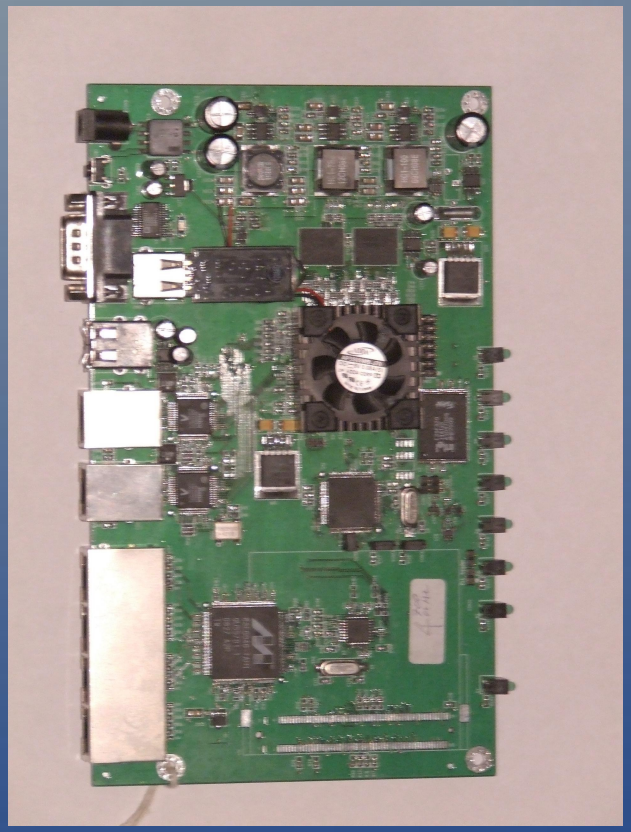


- Intel IXP425 266MHz CPU
- 16MB NOR flash (using 2MB)
- 32MB SDRAM
- 4 port ethernet 10/100 switch
- 1 10/100 etherenet port
- 1 serial port
- 1 mini-PCI slot

Linux on the IXP425

- zImage copied to RAM, 1032820 bytes
- SQUASHfs root filesystem, 901120 bytes
- Added head-hw-init.S only
- Contains “.start” section
- ~90 lines of assembler

Cavium OCTEON 5020



- Octeon 5020 700MHz CPU
- 4MB NOR flash
- 128MB SDRAM
- 4 port 10/100 ethernet switch
- 2 10/100/1000 ethernet ports
- 1 serial port
- 2 USB interfaces

Linux on the Octeon 5020

- linux.bin copied to RAM, 2998656 bytes!
- SQUASHfs root filesystem, 999424 bytes
- Added head-cn50xx.S, included from head.S
- ~111 lines of assembler
- SMP multiple core startup supported

Is It Useful?

- Maybe?
- Certainly possible!

Questions?