ARM Device Tree status report

Grant Likely
Secret Lab Technologies Ltd.

October 28, 2010
Embedded Linux Conference Europe
Cambridge, UK
Overview

- Device Tree Overview
- Integration with the Linux device model
- Current State
- Next steps
IT TOOK A LOT OF WORK, BUT THIS LATEST LINUX PATCH ENABLES SUPPORT FOR MACHINES WITH 4,096 CPUs, UP FROM THE OLD LIMIT OF 1,024.

DO YOU HAVE SUPPORT FOR SMOOTH FULL-SCREEN FLASH VIDEO YET?

NO, BUT WHO USES THAT?
Terminology

- **OpenFirmware (OF) Device Tree (DT)**
  - Device representation exported by Open Firmware
  - This presentation is *not* about Open Firmware
- **Flattened Device Tree (FDT)**
  - Firmware-independent device tree encoding
- **Device Tree Compiler (DTC)**
  - Convert between .dts and .dtb
  - Device Tree Source (.dts)
  - Device Tree Blob (.dtb)
    - Tokenized form; used by kernel
- **Bindings**
  - Documentation of how the DT describes hardware
Device Tree – 3 minute overview

• Data structure for describing hardware
• Passed to kernel at boot
  – By firmware, or
  – Linked into boot wrapper
• Alternative to hard-coded platform details
Device Tree Model

• Tree Structure with nodes & properties
  – Nodes give structure
  – Properties add detail
    • Key-value pairs
    • Arbitrary data

• Secondary links - “phandles”
  – Interdependencies aside from natural tree

• Well defined usage conventions
  – 'compatible' property uniquely identifies devices
  – Each 'compatible' value associated with a 'binding'.
  – Common conventions for address ranges, irqs, gpios and others.
Device Tree – Why?

- Multiplatform
- Simplify board ports
- Standardized Firmware → Kernel data passing
Device Tree – Why not?

- Complexity (Does it buy me anything?)
- Learning curve
- More work?
Device Tree – What it is not?

- Doesn't replace board-specific code
  - Simplifies the common-case
  - Method to identify and handle special cases
- Doesn't add features to your platform
- Isn't a boot architecture
  - (but is an important component)
Model - Firmware

- Firmware obtains .dtb and passes to kernel
- How?
  - Options:
    - Load and pass verbatim
    - Load and modify
    - Generate from scratch
      - Here there be Dragons!
  - Kernel doesn't care
Model – ARM Booting

- Device Tree passed r2 instead of ATAGs
  - All firmware data passed within dt structure
- Early init
  - Determine memory
  - Determine machine
  - Boot to mm set up
Model – ARM Booting

- Unflatten .dtb
  - Allocate space for unpacked form
  - Can now directly dereference tree
- Boot to machine_init
- Register devices
Digression: Linux device model

- Busses
- Devices
- Drivers
- Hierarchy of Devices
Digression: Linux device model

Drivers registered against bus types

- root
- platform
  - spi
  - bridge
    - uart1
  - i2c bus
    - eeprom
    - temp
    - rtc
  - pci bus
    - ethernet

- platform_bus_type
- i2c_bus_type
- pci_bus_type
DT & Linux Device Model historical

- SPARC
  - Walk entire tree, register of_device for each node
  - drivers may bind against any node
  - Other bus registrations mirror of_device hierarchy (ie. PCI)
    - Duplicate 'struct device'
DT & Linux Device Model historical

- PowerPC/Microblaze
  - Subset of tree registered as of_devices
    - Typically only mmio devices
  - Drivers may create child busses of different types
  - No device-tree integration with non-of_devices
    - OF wrappers for other busses
DT & Linux Device Model
new approach

- Conceptual flaws of of_platform_bus
  - Duplicate of platform bus
  - DT data applicable to more than just platform devices
  - platform_driver won't bind against an of_device
DT & Linux Device Model

new approach

- DT is *support data*
  - Make available to all devices
- Move probe data
  - *of_node → struct device*
  - *of_match_table → struct device driver*
- Generalize OF-style binding functions
  - Available to any bus type
- Eliminate of_platform_bus_type
  - Merge with platform_bus_type
Current State

- Mainline Infrastructure works
  - Register mapping
  - IRQ mapping (mostly)
  - Integrated with platform, i2c, spi and mdio bus_types

- Board support (minimal)
  - Versatile on QEMU
  - Versatile Express
  - OMAP3
  - i.MX51
  - All out of mainline

- Registering platform devices
- Binding against drivers
Current State

- Added MIPS support in 2.6.37 merge window
- *Almost* added x86 support in 2.6.37
  - OLPC and Xilinx FPGA
  - Last minute changes defer to 2.6.38
- Any arch can add CONFIG_OF
- ARM remains out of mainline
  - Being cautious, nothing else
Next Steps

- Finish board support
- Complement to ARM Multi-Platform
- Documentation
  - How to use it
  - Bindings
  - http://devicetree.org
- Merge it!
Resources

• Secret Lab git tree
  - git://git.secretlab.ca/git/linux-2.6
  - Branch: test-devicetree
  - Unified code, bindings

• Web sites
  - http://devicetree.org
  - https://wiki.ubuntu.com/KernelTeam/ARMDeviceTrees
Acknowledgements

- CE Linux Forum (now Linux Foundation)
- IBM/Freescale/Canonical
- Too many people to mention