AArch64 Linux Update

Catalin Marinas

Linaro Connect – Dublin 2013

The Architecture for the

the Digital Wo



The Architecture for the Digital World®

Current Status

KVM merged for 3.11

- Marc Zyngier maintainer
- Supporting both AArch64 and AArch32 guests
- Hardware-assisted Stage 2 MMU translation, generic timers, GIC
- kvmtool
- Xen merged for 3.11
 - Stefano Stabellini (Citrix) maintainer
- Hugetlbfs, transparent huge pages merged
 - Code sharing with x86
- SMP booting protocols
 - Spin-table
 - PSCI (Power State Coordination Interface)

Current Status (2)

SoC support for ARMv8 software models

- Versatile Express like
- Code under drivers/
- GIC shared with arch/arm/ and secondary CPU interface initialisation via CPU notifiers
- Generic timers shared with arch/arm/
- Vexpress reset moved to drivers/power/reset/
- Still no DT-aware CLCD driver in mainline
- Initial support for Applied X-Gene SoC
 - .dts file under arch/arm64/
 - Code under drivers/ being pushed via corresponding maintainers
- soc-armv8-model branch no longer needed (unless CLCD support for ARMv8 models is required)



New Developments

- Power management (cpuidle), CPU hotplug
 - Based on PSCI firmware calls
- Support for new ARM IP
 - Processors (Cortex-A53/A57)
 - GICv3 (> 8 CPUs)
 - SMMU
- PCIe
 - Scope for code sharing with PowerPC/MIPS/AArch32
- Optimisations [Linaro]
 - Linux klib (memcpy etc.) based on the Cortex Strings library
- Debugging [Linaro]
 - kprobes, uprobes, ftrace, kgdb
- Kexec, kernel crashdump [Linaro]



New Developments (2)

- UEFI run-time services [Linaro]
- ACPI [Linaro]
- New SoCs
 - Mainly under drivers/
- CPU topology, caches
 - Current development for AArch32
- KVM
 - PMU support [ARM/Linaro]
 - VFIO support [ARM/Linaro]
 - Live migration [Linaro]
 - GICv3 (> 8 vCPUs)
- IOMMU API extensions
 - Stage 2 translation

New Developments (3)

- Ticket spinlocks
- Optimised ASID allocator
- Tagged pointers
 - Top 8-bit used as a tag in user-space pointers
- ILP32



Enabling SoCs for AArch64

- Pre-Linux (firmware, boot loader) requirements
 - Requirements in Documentation/arm64/booting.txt
 - Device tree required
 - Linux (all CPUs) entered at EL2 for virtualisation support
 - PSCI (Power State Coordination Interface) strongly recommended
 - It may not possible on CPUs without EL3 (alternative spin-table method for booting secondary CPUs)
- CPUs, enable-method described via DT
 - cpu_logical_map populated from DT
- Standard devices (described via DT)
 - GIC (v2, v3)
 - Generic timers (compatible = "arm,armv8-timer")
 - Timer frequency specified either via DT or in register (CNTFRQ_EL0, usually set by firmware)



Enabling SoCs for AArch64 (2)

Clock drivers

- Code under drivers/clk/
- Use CLK_OF_DECLARE() and corresponding DT entries
- of_clk_init() called from arm64_device_init() (arch_initcall)
- of_platform_populate() used for additional devices
 - Called from arm64_device_init() (arch_initcall)
- Early printk support
 - Assuming UART port initialised by firmware prior to Linux
 - Simple "printch" function in arch/arm64/kernel/early_printk.c
 - Command line argument: earlyprintk=<name>[,<addr>][,<options>]
- Other device drivers
 - Preferably loadable modules unless essential for booting

Enabling SoCs for AArch64 (3)

- Multi-platform mandatory
- No ARCH_* in arch/arm64/Kconfig
 - Drivers Kconfig entries dependent on ARM64
 - Existing ARCH_* entries to be removed
- arch/arm64/configs/defconfig enables all supported SoCs
- DTS files go under arch/arm64/boot/dts/
- Similar to the AArch32 SoC requirements
 - "Your new ARM SoC Linux support check-list" (Thomas Petazzoni) http://elinux.org/images/a/ad/Arm-soc-checklist.pdf
 - But without arch/arm64/mach-*/ directories

