Linux kernel
on Kwikbyte AT92RM9200

Architecture, CPU and machine

- ✓ In the source tree, each architecture has its own directory arch/arm for the ARM architecture
- ✓ This directory contains generic ARM code
- ✓ boot, common, configs, kernel, lib, mm, nwfpe, vfp, oprofile, tools
- ✓ And many directories for different CPU families
- ✓ mach-* directories : mach-pxa for PXA CPUs, mach-imx for Freescale iMX CPUs, etc.
- ✓ Each of these directories contain
- ✓ Support for the CPU
- ✓ Support for several boards using this CPU
- ✓ Some CPU types share some code, in an entity called a *platform*
- ✓ plat-omap contains common code from mach-omap1 and mach-omap2

Source code for KB9202

- ✓ arch/
- ✓ arm/
- ✓ mach-at91/
- ✓ CPU-specific code for the AT91RM9200 at91rm9200.c, at91rm9200_time.c, at91rm9200_devices.c
- ✓ Board specific code

  board-kb9202.c
- ✓ For the rest of this presentation, we will focus on board support only

Configuration

- ✓ A configuration option must be defined for the board, in arch/arm/mach-at91/Kconfig

```
config MACH_KB9200
        bool "KwikByte KB920x"
        depends on ARCH_ATRM9200
        help
         Select this if you are using KwikByte's KB920x board.
         <http://kwikbyte.com/KB9202_description_new.htm>
```

- ✓ This option must depend on the CPU type option corresponding to the CPU used in the board
- ✓ Here the option is ARCH_ATRM9200, defined in the same file
- ✓ A default configuration file for the board can optionally be stored in arch/arm/configs/. For our board, it's kb9202__defconfig

Compilation

➢ The source files corresponding to the board support must be associated with the configuration option of the board

```
obj-$(CONFIG_MACH_KB9200)      += board-kb9202.o
```

```
obj-$(CONFIG_ARCH_AT91RM9200)  += at91rm9200.o at91rm9200_time.o
at91rm9200_devices.o
```

Machine structure

- ✓ Each board is defined by a machine structure
- ✓ The word « machine » is quite confusing since every mach-* directory contains several machine definitions, one for each board using a given CPU type
- ✓ For the KB9202 board, at the end of arch/arm/mach-at91/board-kb9202.c

```
MACHINE_START(KB9200, "KB920x")
        /* Maintainer: KwikByte, Inc. */
        .phys_io          = AT91_BASE_SYS,
        .io_pg_offst      = (AT91_VA_BASE_SYS >> 18) & 0xfffc,
        .boot_params      = AT91_SDRAM_BASE + 0x100,
        .timer            = &at91rm9200_timer,
        .map_io           = kb9202_map_io,
        .init_irq         = kb9202_init_irq,
        .init_machine     = kb9202_board_init,
MACHINE_END
```

Machine structure macros

- ✓ MACHINE_START and MACHINE_END
- ✓ Macros defined in arch/arm/include/asm/mach/arch.h
- ✓ They are helpers to define a struct machine_desc structure stored in a specific ELF section
- ✓ Several machine_desc structures can be defined in a kernel, which means that the kernel can support several boards.
- ✓ The right structure is chosen at boot time

Machine type number

- ✓ In the ARM architecture, each board type is identified by a machine type number
- ✓ The latest machine type numbers list can be found at http://www.arm.linux.org.uk/developer/machines/download.php
- ✓ A copy of it exists in the kernel tree in arch/arm/tools/mach-types
- ✓ For the KB9202 board
  ```
  kb9200    MACH_KB9200         KB9200 612
  ```
- ✓ At compile time, this file is processed to generate a header file, include/asm-arm/mach-types.h
- ✓ For the KB9202 board
  #define MACH_TYPE_KB9200 612
- ✓ And a few other macros in the same file

Machine type number

- ✓ The machine type number is set in the MACHINE_START() definition
  MACHINE_START(KB9200, "KB920x")
- ✓ At run time, the machine type number of the board on which the kernel is running is passed by the bootloader in register *r1*
- ✓ Very early in the boot process (arch/arm/kernel/head.S), the kernel calls __lookup_machine_type in arch/arm/kernel/head-common.S
- ✓ __lookup_machine_type looks at all the machine_desc structures of the special ELF section
- ✓ If it doesn't find the requested number, prints a message and stops
- ✓ If found, it knows the machine descriptions and continues the boot process

System timer

- ✓ The timer field point to a struct sys_timer structure, that describes the system timer
- ✓ Used to generate the periodic tick at HZ frequency to call the scheduler periodically
- ✓ On the KB9202 board, the system timer is defined by the struct at91rm9200_timer structure in at91rm9200_time.c
- ✓ It contains the interrupt handler called at HZ frequency
- ✓ It is integrated with the clockevents and the clocksource infrastructures
- ✓ See include/linux/clocksource.h and include/linux/clockchips.h for details

map_io()

- ✓ The map_io() function points to kb9202_map_io(), which
- ✓ Initializes the CPU using at91rm9200_initialize()
- ✓ Map I/O space
- ✓ Register and initialize the clocks
- ✓ Configures the debug serial port and set the console to be on this serial port
- ✓ Called at the very beginning of the C code execution
- ✓ init/main.c: start_kernel()
- ✓ arch/arm/kernel/setup.c: setup_arch()
- ✓ arch/arm/mm/mmu.c: paging_init()
- ✓ arch/arm/mm/mmu.c: devicemaps_init()
- ✓ mdesc->map_io()

init_irq()

- ✓ init_irq() to initialize the IRQ hardware specific details
- ✓ Implemented by kb9202_init_irq() which calls at91rm9200_init_interrupts() in at91rm9200.c, which mainly calls at91_aic_init()
- ✓ Initialize the interrupt controller, assign the priorities
- ✓ Register the IRQ chip (irq_chip structure) to the kernel generic IRQ infrastructure, so that the kernel knows how to ack, mask, unmask the IRQs
- ✓ Called a little bit later than map_io()
- ✓ init/main.c: start_kernel()
- ✓ arch/arm/kernel/irq.c: init_IRQ()
- ✓ init_arch_irq() (equal to mdesc->init_irq)

init_machine()

- ✓ init_machine() completes the initialization of the board by registering all platform devices
- ✓ Called by customize_machines() in arch/arm/kernel/setup.c
- ✓ This function is an arch_initcall (list of functions whose address is stored in a specific ELF section, by levels)
- ✓ At the end of kernel initialization, just before running the first userspace program init:
- ✓ init/main.c: kernel_init()
- ✓ init/main.c: do_basic_setup()
- ✓ init/main.c: do_initcalls()
- ✓ Calls all initcalls, level by level

init_machine() for KB9202

- ✓ For the KB9202 board, implement in kb9202_board_init()
- ✓ Registers serial ports, USB host, USB device, SPI, Ethernet, NAND flash, 2IC, buttons and LEDs
- ✓ Uses at91_add_device_*() helpers, defined in at91rm9200_devices.c
- ✓ These helpers call platform_device_register() to register the different platform_device structures defined in the same file