

# ARC — new first class citizen in U-Boot

Porting U-Boot to yet another ARChitecture

Alexey Brodkin

October 13<sup>th</sup>, 2014

# Agenda

- Where we started
- Why move?
- First steps
- Great rebase
- Clean-up
- Review
- Being a custodian
- Next steps
- Things to improve in U-Boot

# Where we started

- Pre-historic  
1.1.3, 1.3.3 in internal SVN
- Middle ages  
2009.01, 2009.06, 2011.03 in internal SVN
- Modern history  
2012.07, 2013.01 on GitHub  
<https://github.com/foss-for-synopsys-dwc-arc-processors/u-boot>



# Why move?

- Get up-to-date:
  - a. drivers
  - b. filesystems
  - c. features
  - d. fixes
- Escape manual rebasing of ARC patches
- Make ARC more visible
- Aren't we as cool as others?



# First steps

- Added support for new board
- Fixes for used drivers:
  - a. DesignWare SD/MMC
  - b. DesignWare I2C
  - c. DesignWare GMAC
- Fixes for CPUs with caches

# Great rebase

- Switch to common board
  - a. Init calls - which ones to use?
- Relocation is required!
  - a. No assembly, please (steal from x86)
  - b. “Elf32\_Rel” but not “Elf32\_Rel”
  - c. Middle-endian on little-endian
- PIC
  - a. Initial success
  - b. CONFIG\_NEEDS\_MANUAL\_RELOC
  - c. Not enough MANUAL\_RELOC
- No more PIC, use PIE

```
board_rc x
746  ->  intr_addr_map,
747  #endif
748  #if defined(CONFIG_BOARD_EARLY_INIT_R)
749  ->  board_early_init_r,
750  #endif
751  ->  INIT_FUNC WATCHDOG_RESET
752  #ifdef CONFIG_LOGBUFFER
753  ->  intr_logbuffer,
754  #endif
755  #ifdef CONFIG_POST
756  ->  intr_post_backlog,
757  #endif
758  ->  INIT_FUNC WATCHDOG_RESET
759  #ifdef CONFIG_SYS_DELAYED_ICACHE
760  ->  intr_icache_enable,
761  #endif
762  #if defined(CONFIG_SYS_INIT_RAM_LOCK)
763  ->  intr_unlock_ram_in_cache,
764  #endif
765  #if defined(CONFIG_PCI) && defined(CONFIG_
766  -> /*
767  -> * Do early PCI configuration.
768  -> * because PCU resources are
769  -> */
770  -> intr_pci,
771  #endif
772  #ifdef CONFIG_W10BOND_B3C553
773  -> intr_w83c553f,
774  #endif
775  #ifdef CONFIG_ARCH_EARLY_INIT_R
776  -> arch_early_init_r,
777  #endif
778  -> power_init_board,
779  #ifdef CONFIG_SYS_NO_FLASH
780  -> intr_flash,
781  #endif
782  -> INIT_FUNC WATCHDOG_RESET
783  #if defined(CONFIG_PPC) || defined(CON
784  -> /* initialize higher level par
785  -> cpu_init_r,
786  #endif
787  #ifdef CONFIG_PPC
788  -> intr_spi,
789  #endif
790  #if defined(CONFIG_X86) && defined(CON
791  -> init_func_spi,
792  #endif
793  #ifdef CONFIG_CMD_NAND
794  -> intr_nand,
795  #endif
796  #ifdef CONFIG_CMD_ONENAND
797  -> intr_onenand,
798  #endif
799  #ifdef CONFIG_GENERIC_MMC
800  -> intr_mmc,
801  #endif
802  #ifdef CONFIG_HAS_DATAFLASH
803  -> intr_dataflash,
804  #endif
805  -> intr_env,
806  -> INIT_FUNC WATCHDOG_RESET
807  -> intr_secondary_cpu,
808  #ifdef CONFIG_SC3
809  -> intr_sc3_read_eeprom,
810  #endif
811  #ifdef CONFIG_HERMES
812  -> intr_hermes,
board_fc x
807  -> if (ret)
808  -> return ret;
809  #endif
810
811  -> return 0;
812 }
813
814 static init_func_t init_sequence_f[] = {
815 #ifdef CONFIG_SANDBOX
816 -> setup_ram_buf,
817 #endif
818 -> setup_mon_len,
819 -> setup_fdt,
820 -> trace_early_init,
821 #if defined(CONFIG_MPC85xx) || defined(CONFIG_M
822 -> /* TODO: can this go into arch_cpu_init
823 -> probecpu,
824 #endif
825 -> arch_cpu_init, -> /* basic arch.c
826 #ifdef CONFIG_X86
827 -> cpu_init_f, -> /* TODO(sj@chr
828 #ifdef CONFIG_OF_CONTROL
829 -> find_fdt, -> /* TODO(sj@chr
830 #endif
831 #endif
832 -> mark_bootstage,
833 #ifdef CONFIG_OF_CONTROL
834 -> fdtdec_check_fdt,
835 #endif
836 -> initf_malloc,
837 -> initf_dm,
838 #if defined(CONFIG_BOARD_EARLY_INIT_F)
839 -> board_early_init_f,
840 #endif
841 -> /* TODO: can any of this go into arch.c
842 #if defined(CONFIG_PPC) && !defined(CONFIG_8xx
843 -> get_clocks, -> /* get CPU and
844 #if defined(CONFIG_TQM8xxL) && !defined(CONFIG_
845 -> && !defined(CONFIG_TQM85D)
846 -> adjust_sdram_tbs_8xx,
847 #endif
848 -> /* TODO: can we rename this to timer_in
849 -> init_timebase,
850 #endif
851 #if defined(CONFIG_ARM) || defined(CONFIG_MIPS)
852 -> timer_init, -> /* initialize t
853 #endif
854 #ifdef CONFIG_SYS_ALLOC_DPRAM
855 #if defined(CONFIG_CPM2)
856 -> dpram_init,
857 #endif
858 #endif
859 #if defined(CONFIG_BOARD_POSTCLK_INIT)
860 -> board_postclk_init,
861 #endif
862 #ifdef CONFIG_FSL_ESDHC
863 -> get_clocks,
864 #endif
865 -> env_init, -> /* initialize e
866 #if defined(CONFIG_Bxx_CPUCLK_DEFAULT)
867 -> /* get CPU and bus clocks according to
868 -> get_clocks_866,
869 -> /* adjust sdram refresh rate according
870 -> sdram_adjust_866,
871 -> init_timebase,
872 #endif
873 -> init_baud_rate, -> /* initialize b
```

# Clean-up

- Unused headers and legacy code
- Linux stuff
  - a. Some definitions differ: CPU\_BIG\_ENDIAN vs SYS\_BIG\_ENDIAN
  - b. \_\_KERNEL\_\_ is obsolete in U-Boot
  - c. Dead/unused in U-boot parts of headers
  - d. Re-format even code from upstream kernel

# Review

- Recent port as a reference (OpenRISC)
- More clean-up
- Explain unexpected arch specifics
  - a. AUX regs are in separate address space.
- Accessors saga
  - a. “inb”, “inl”, “inw”; “outb”, “outl”, “outw”
  - b. “IO\_READ8”, “IO\_READ16”, “IO\_READ32”;  
“IO\_WRITE8”, “IO\_WRITE16”, “IOWRITE32”
  - c. “readb”, “readl”, “readw”; “writeb”, “ritel”, “writew”
- Upstream never sleeps



# Being a custodian

- Repository on <http://git.denx.de/>
  - a. It's you who decides what is acceptable
  - b. Move sets of patches upstream at once
- Wiki on [www.denx.de/wiki/U-Boot](http://www.denx.de/wiki/U-Boot)
  - a. Save handy tips and tricks for yourself
    - How to manage custodian repo
  - a. Provide useful informations for users
    - Tools for your architecture
    - How to run your U-Boot on target

# Next steps

- ISAv2 support (ARC EM, ARC HS)
  - a. Incompatible vector table
- More boards
  - a. nSIM OSCI
  - b. EMSK v1/v2
- SPL
  - a. No printf
  - b. Avoid usage of libcommon and friends



# Things to improve in U-Boot

- More generalization
  - a. Init calls
  - b. Accessors
  - c. Relocation functionality
- Wrappers for printf
  - a. Debug
  - b. Info
  - c. Errors

# Thank You

**SYNOPSYS**<sup>®</sup>  
Accelerating Innovation

