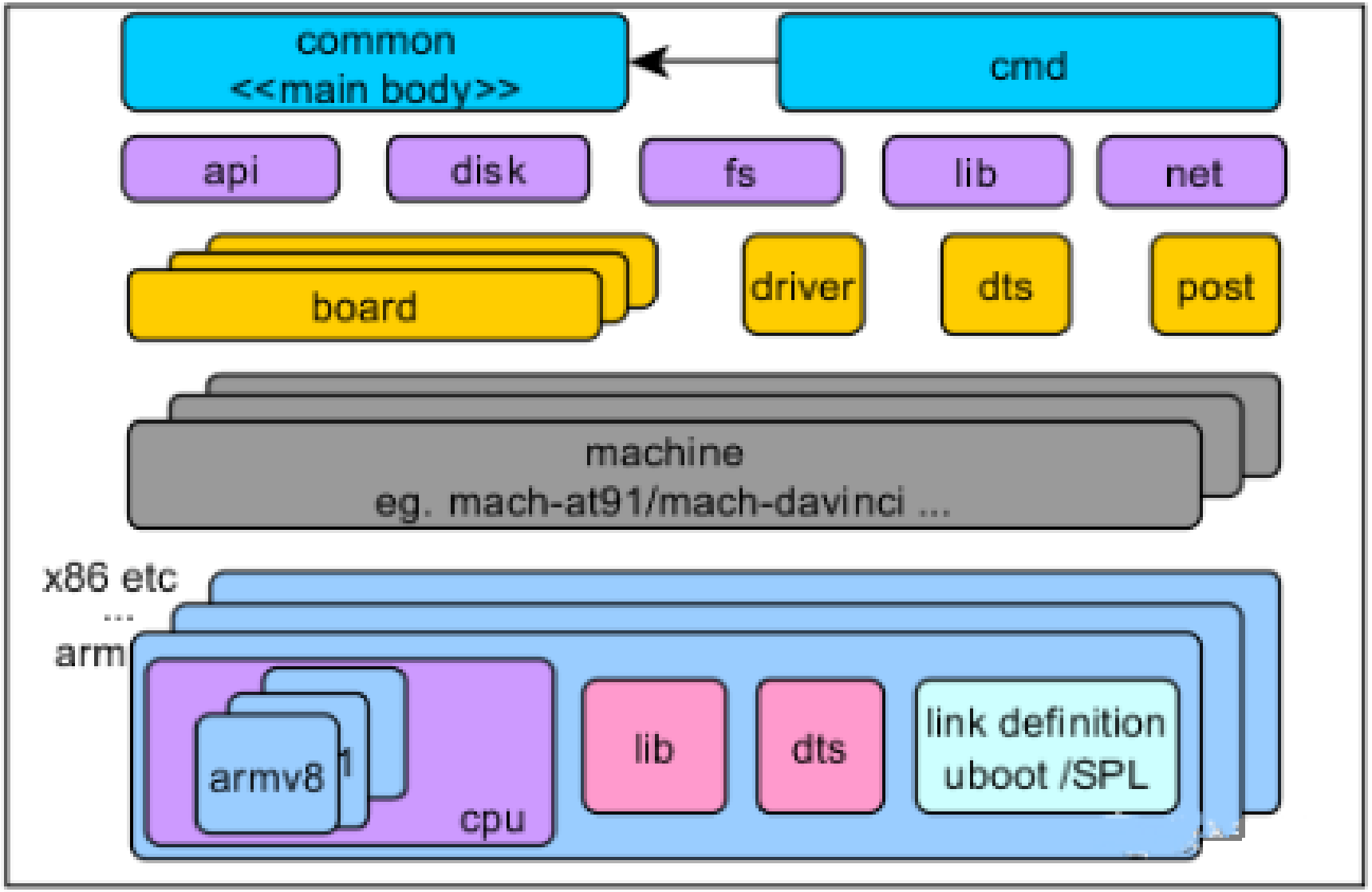
Embedded Linux Porting (C4)

Bootloader (U-Boot)

Agenda

- U-BOOT Architecture
- U-BOOT Commands (L)
- U-BOOT EVN Variables
- U-BOOT Directory Structure
- U-BOOT Code Flow
- U-BOOT Commands (P)

U-Boot ARCH


common
<<main body>>
cmd
api
disk
fs
lib
net
board
driver
dts
post
machine
eg. mach-at91/mach-davinci ...
x86 etc
...
arm
armv8
cpu
lib
dts
link definition
uboot /SPL

U-Boot Commands

Information Commands	
help	*print online help*
bdinfo	*print Board Info structure*
coninfo	*print console devices and information*
flinfo	*print FLASH memory information*

Basic Commands	
version	*print monitor version*
echo	*echo arguments to console*
reset	*perform RESET of the CPU*
sleep	*delay execution for some time*
cls	*Clear screen*

Environment Variables Commands	
env	*environment handling commands*
printenv	*print environment variables*
setenv	*set environment variables*
editenv	*edit environment variable*
saveenv	*save environment variables to persistent storage*

Memory Commands	
mtest	*simple RAM test*
md	*echo arguments to console*
mm	*memory modify (auto incrementing)*
mw	*memory write (fill)*
nm	*memory modify (constant address)*
base	*print or set address offset*
crc32	*checksum calculation*
cp	*memory copy*

U-Boot Commands

Download	& BOOT Commands
loadb	*load binary file over serial line (kermit mode)*
loady	*load binary file over serial line (ymodem mode)*
loads	*load S-Record file over serial line*
Ping	*send ICMP ECHO REQUEST to network host*
bootp	*boot image via network using BOOTP/TFTP protocol*
dhcp	*invoke DHCP client to obtain IP/boot params*
tftpboot	*boot image via network using TFTP protocol*
nfs	*boot image via network using NFS protocol*
boot	*boot default, i.e., run 'bootcmd'*
bootm	boot application image from memory
Nboot	boot from NAND device
go	start application at address 'addr'
fatload	load binary file from a FAT file system
Ext2load	load binary file from a Ext2 filesystem

HW Subsytem	
gpio	manipulate gpios
i2c	I2C sub-system control
mmc	MMC sub system
usb	USB sub-system control
ftd	flattened device tree utility commands
mtdparts	define flash/nand partitions
eeprom	EEPROM sub-system control
nand	NAND sub-system control
flinfo	print FLASH memory information
erase	erase FLASH memory
sf	*Serial Flash sub-system*

U-BOOT Env Variables

U-BOOT Environment Variables	
ipaddr	Board IP Address
serverip	Server IP Address
bootenv	Kernel Command line Arguments
bootcmd	Default command executed by u-boot to boot the system

U-BOOT Source

Browse Source: https://github.com/rugged-board/uboot-rba5d2x

Download U-Boot for RuggedBOARD

$ wget https://github.com/rugged-board/uboot-rba5d2x/archive/uboot-rba5d2x.zip

Or

$ git clone https://github.com/rugged-board/uboot-rba5d2x.git

U-BOOT Directory Structure

phytec@phytec: ~/phytec/uboot-rba5d2x

```
phytec:uboot-rba5d2x$ ls
api          disk           include      post         u-boot               u-boot.lds
arch         doc            Kbuild       README       u-boot.bin           u-boot.map
board        Documentation  Kconfig      scripts      u-boot.cfg           u-boot-nodtb.bin
boot.bin     drivers        lib          spl          u-boot.cfg.configs   u-boot.srec
cmd          dts            Licenses     System.map   u-boot.dtb           u-boot.sym
common       env            MAINTAINERS  tags         u-boot-dtb.bin
config.mk    examples       Makefile     test         u-boot-dtb.img
configs      fs             net          tools        u-boot.img
```

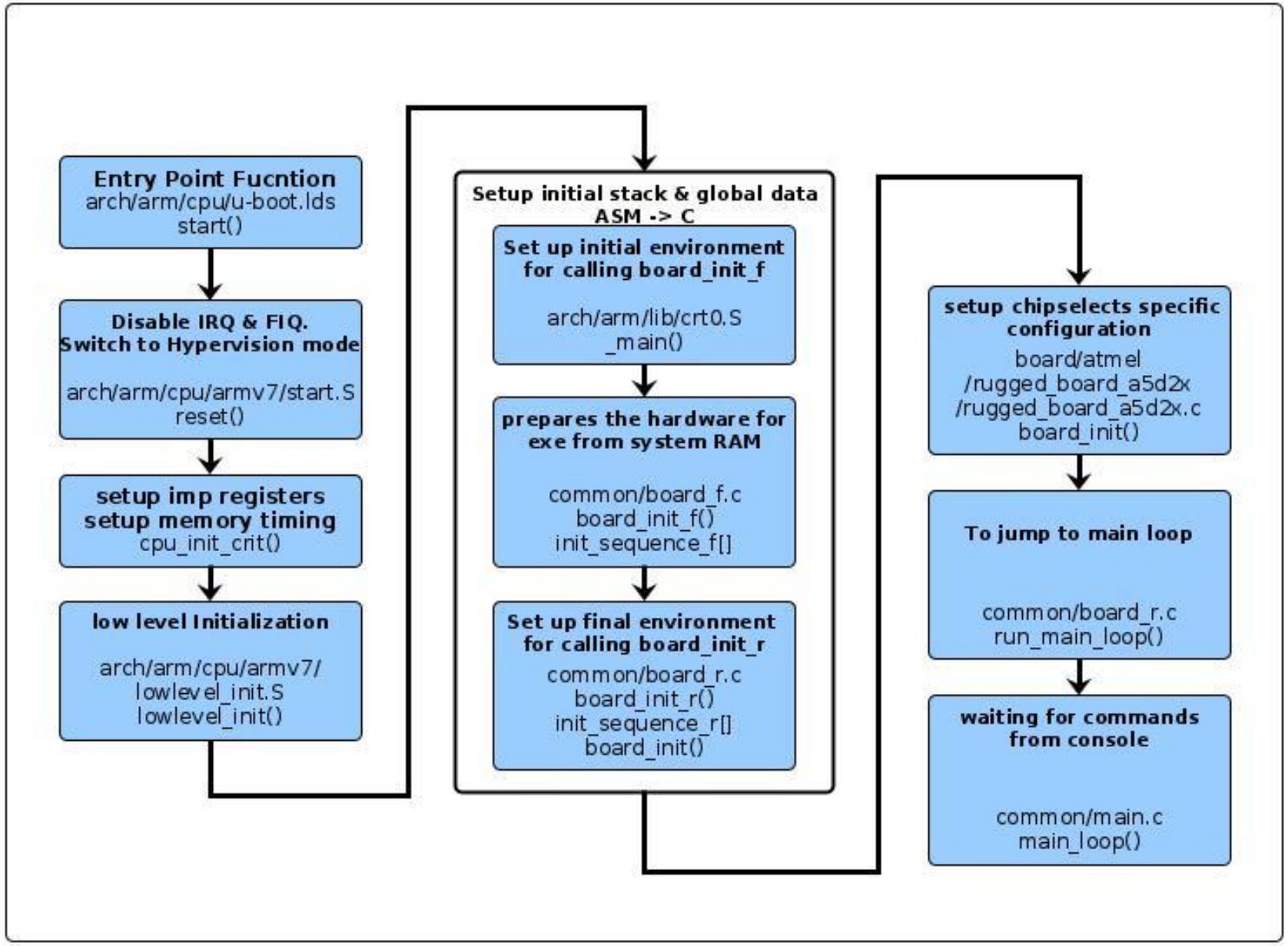
U-BOOT Dir Structure

uboot/arch/arm/cpu	Arch & Core specific code, u-boot.lds armv7/start.S, cpu.c
uboot/arch/arm/mach-at91	SOC specific code, armv7/sama5d2_devices.c
uboot/arch/arm/dts	device tree directory consists of device tree files for SOC, SOM, SBC *sama5d2.dsi , rb_a5d2x.dtsi, rugged_board_a5d2x.dts*
uboot/board/atmel/rugged_board_a5d2x	Board directory contains board files with syntax <vendor>/<boardname> boardname.c called board file.
uboot/configs	Contains board default configuration file used to configure uboot for a specific board. <boardname_defconfig> for ruggedboard we have two files for NOR: rugged_board_a5d2x_qspiflash_defconfig & for SDCARD: rugged_board_a5d2x_mmc1_defconfig
uboot/drivers	Contains bus drivers & device drivers (gpio, serial, i2c, spi, mmc, usb, net) at91_gpio.c, atmel_usart.c, at91_i2c.c, atmel_sdhci.c, atmel_spi/qspi.c, at91_emac.c Device Driver: rtc/ds1307.c, misc/i2c_eeprom.c …

U-BOOT Dir Structure

uboot/common	U-Boot Bootloader SW Logic and main code
uboot/cmd	U-Boot Command Line Interface Implementation
uboot/fs	File systems implemented/supported by u-boot
uboot/net	Network sub-system/stack of u-boot
uboot/Documentation	U-Boot documentation
uboot/post	U-Boot Power On Self test function
uboot/env	Environment variable SW module

U-BOOT Code Flow



U-BOOT Code Flow

board_init_f()

- initf_bootstage	/* uses its own timer,so doesn't need DM */
- arch_cpu_init	/* basic arch cpu dependent setup */
- mach_cpu_init	/* SoC/machine dependent CPU setup */
- get_clocks	/* get CPU and bus clocks (etc.) */
- timer_init	/* initialize timer */
- env_init	/* initialize environment */
- init_baud_rate	/* initialze baudrate settings */
- serial_init	/* serial communications setup */
- console_init_f	/* stage 1 init of console */
- dram_init	/* configure available RAM banks */

U-BOOT Code Flow

board_init_r()

- board_init	/* Setup chipselects */
- set_cpu_clk_info	/* Setup clock information */
- initr_nand	/* initialize flash */
- initr_mmc	/* initialize fmmc */
- console_init_r	/* fully init console as a device */
- arch_misc_init	/* miscellaneous arch-dependent init */
- misc_init_r	/* misc platform-dependent init */
- interrupt_init	/*set up exceptions */
- initr_enable_interrupts	/* enable exceptions */
- initr_ethaddr	/* setup ethernet */
- board_late_init	/* board late initialization */
- run_main_loop	/* jump to main loop & waiting for commands from console */

U-Boot Ends with:

main.c

will run the default boot command ***autoboot_command(s);***

or

start the command line ***cli_loop();***

Open Discussions

Connect-us