Design Document for Base Port

1 Outline

This document describes the base port (including MSL) in Linux kernel of MVF TOWER BOARD (XTWR-VF600) with VF6XX SoC.

2 Code to be added

In order to implement the base port, the following sources and header files are added to the Linux Kernel Source tree.

Files below are added in reference to MX6 files (arch/arm/mach-mx6). However for Release 1, the implementation is kept to a minimal level where serial console, FEC (Ethernet), FB and keypad work. The priority for implementation is sorted out as below and "A"s are implemented selectively and selectively for Release 1.

A ... Essential
B ... Necessary in the future (Release2 or later)
C ... Possibly necessary
D ... Unnecessary for this CPU

- arch/arm/mach-mvf/Kconfig (A)
  Kernel configuration file for MVF SoC family
  Settings:
  - SoC and board-type selection
  - Device groups to be enabled
  - Support config of devices to be enabled and each device driver

- arch/arm/mach-mvf/Makefile (A)
  Makefile file for MVF SoC family
  Define association between configuration and files to be build

- arch/arm/mach-mvf/Makefile.boot (A)
  Makefile.boot file for MVF SoC family

Definitions of each address:

- zreladdr

  Beginning address of the kernel

  Generally, beginning of RAM address + 0x8000
- params_phys

  Address of parameter that is given by u-boot

  Generally it is beginning of RAM address + 0x100. Implement after checking where u-boot writes.
- initrd_phys

  Beginning address of reading Initrd

  There may be no definition depending on board

  It is often the case to use beginning address of RAM + 0x00800000, however it is TBD for this board.

- arch/arm/mach-mvf/board-twr_vf600.c (A)

  Specific definition for XTWR-VF600 board

  Definition for board-specific info:

  - Device definition
  - Define board-specific initialization function
  - Other

  Definition for callback function:

  - Define board-specific function to change settings (and such) for each device

- arch/arm/mach-mvf/board-twr_vf600.h (A)

  XTWR-VF600-specific header file

  Definition:

  - PAD definition for multi-function pin

- arch/arm/mach-mvf/bus_freq.c (B?)

  Bus frequency setting (low or high) source file for MVF SoC family, however only dummy function is implemented for MX6 series.

  Determined unnecessary for Release 1, but it may become necessary at the time of PM implementation (depending on specification).

- arch/arm/mach-mvf/clock.c (A)

Clock implementation for MVF SoC family

- Define clock tree
- Define callback function for each clock
- Clock initialization

- arch/arm/mach-mvf/cpu.c (C)

  CPU initialization for MVF SoC family

  Details:

  - Initialize IRAM
  - Initialize device controller and such (for MX6)

- arch/arm/mach-mvf/cpu_op-mvf.c (C)

  Mechanism that supports CONFIG_CPU_FREQ and dynamically change CPU clock

  With MX6, it is defined but is not used

  Unnecessary if CPU clock is locked

- arch/arm/mach-mvf/cpu_op-mvf.h (C)

  Same as cpu_op-mvf.h above

- arch/arm/mach-mvf/ cpu_regulator-mvf.c (B?)

  Voltage setting for CPU clock

  - Set regulator to suit CPU clock
  - Modify loop_per_jiffy for clock change

  Unnecessary if CPU clock is locked

  - Necessary if CPU clock becomes variable eventually
  - Note that this setting includes changes in loop_per_jiffies
  - Executed only at boot since this is called as a part of initialization

- arch/arm/mach-mvf/crm_regs.h (A)

  Clock header file for MVF SoC family

  Define PLL and CCM (Clock Controller Module) register

- arch/arm/mach-mvf/devices-vf6xx.h (A)

  Define wrapper macro for device definition

Associating with definitions of devices under arch/arm/plat-mxc/devices/

Define minimal ones for Release 1

- arch/arm/mach-mvf/devices.c (A)

  Device setup source file for MVF SoC family

  - GPIO definition

- arch/arm/mach-mvf/dummy_gpio.c (C)

  Definition of function called by device driver for GPIO control

  - There is a case that it is called by device driver that controls GPIO at the time of device operation.
  - This is basically unnecessary, but become necessary if gpio_xxx_active, gpio_xxx_inactive and such are needed in device driver. (to be empty function with this SoC family)

  Implement as required by device driver from Release 2 or later

- arch/arm/mach-mvf/headsmp.S (D)

  Necessary only when CONFIG_SMP is enabled

  Target SoC is single core, so no SMP implementation is done this time

- arch/arm/mach-mvf/irq.c (A)

  GIC interrupt processing for MVF SoC family

  Initialization of Watchdog address

- arch/arm/mach-mvf/localtimer.c (D)

  Necessary only when CONFIG_SMP is enabled

  Target SoC is single core, so no SMP implementation is done this time

- arch/arm/mach-mvf/mm.c (A)

  Memory map initialization for MVF SoC family

  Initialize secondary cache (if necessary)

- arch/arm/mach-mvf/mvf_anatop_regulator.c (D)

  Regulator control for Anatop

  It is basically unnecessary

  - If regulator control is used, dedicated (additional) source is necessary
  - Must consult with Freescale for details of regulator

- arch/arm/mach-mvf/mvf_fec.c (C)

  Initialization processing of FEC (Fast Ethernet Controller)

  - MAC address acquisition (procedures must be identified) and its setting
  - If MAC address has any problem, use random MAC address
  - Register platform device

- arch/arm/mach-mvf/mvf_wfi.S (B)

  Processing to stop CPU till interrupt occurs

  - Copy mvf_init to iram in cpu.c
  - This process may not be implementable if IRAM is unusable

  Create as necessary at the time of PM implementation

  - Processing to halt CPU when CPU is in the idle state
  - Probably possible only to call cpu_do_idle in arch_idle
  - When a kernel parameter enable_wait_mode is specified, code in this source is called

- arch/arm/mach-mvf/mvf_suspend.S (B)

  Suspend/resume processing around memory at the time or PM control

  - Suspend/resume processing of DDR
  - Low-level power source control around I/O
  - Power source control of secondary cache

  Necessary at the time of PM implementation

- arch/arm/mach-mvf/plat_hotplug.c (D)

  Necessary only when CONFIG_SMP is enabled

  Target SoC is single core, so no SMP implementation is done this time

- arch/arm/mach-mvf/plat_smp.c (D)

  Necessary only when CONFIG_SMP is enabled

  Initialization processing for the second and subsequent CPU in case of multi-core

  Target SoC is single core, so no SMP implementation is done this time

- arch/arm/mach-mvf/pm.c (B)

  Power source control for each device

  Power source control for CPU

  Necessary at the time of PM creation

- arch/arm/mach-mvf/regs-anadig.h (A)

  Header file for ANADIG register definition

  - Define ANADIG register address
  - Define ANADIG register bit field

- arch/arm/mach-mvf/ src-reg.h (A)

  Register definition for SRC (System Reset Controller)

- arch/arm/mach-mvf/system.c (B or D)

  Power source control of CPU after WFI (Wait for Interrupt)

  Soft reset processing for specific device

  May be used depending on PM implementation

- arch/arm/mach-mvf/usb.h (B)

  Definition for USB low-level function

- arch/arm/mach-mvf/usb_dr.c (B)

  Low-level code for USB OTG

  - USB control
  - Associate host and PHY

- arch/arm/mach-mvf/usb_h1.c (B)

  Low-level code for USB host

  - USB control
  - Associate host and PHY

- arch/arm/mach-mvf/usb_h2.c (B)

  Low-level code for USB host

  - USB control
  - Associate host and PHY

- arch/arm/mach-mvf/usb_h3.c (B)

  Low-level code for USB host

  - USB control
  - Associate host and PHY

- arch/arm/plat-mxc/ include/mach/mvf.h

  Header file for MVF SoC family

- arch/arm/plat-mxc/ mvf_time.c

  Low-level code for kernel timer for MVF SoC family

  CortexA5 Global Timer is used as Timer source

- arch/arm/plat-mxc/ mvf_pit_time.c

  Low-level code for kernel timer for MVF SoC family

  VF6XX SoC PIT (Periodic Interrupt Timer) is used as Timer source

  Note:

  This function is enabled when CONFIG_MVF_USE_PIT in kernel configuration is ON, as substitute for mvf_timer.c.

  This function will be excluded from Release 2 as kernel timer implementation is completed with Global Timer.

3 Existing code to be changed

Make alterations to the files below.

- arch/arm/plat-mxc/Kconfig

  Add ARCH_MVF menu

- arch/arm/plat-mxc/devices/platform-xxx.c

  Add resource definition for MVF SoC family or for VF6XX SOC

  Refer to design document of each device driver for details

- arch/arm/plat-mxc/ include/mach/mxc.h

Add definitions for VF6XX SoC and TWR_VF600 board

4 API of new functions

No API is added due to since it has no driver
If names as mx6, mx6q and such are used for functions etc., it will be replaced with mvf and vf6xx.

5 Expected register settings

TBD

6 Expected functionality and usage

This base port is done based on the implementation of MX6 SoC family, thus follows its functions/usage as well.

7 Any other pertinent information

None