Design Document for Serial Driver

1 Outline

This document describes the serial driver in Linux kernel of MVF TOWER BOARD (XTWR-VF600) with MVF SoC.

2 Code to be added

In order to implement the serial driver, the following source and header file is added to the Linux Kernel Source tree.

- drivers/tty/serial/serial_mvf.c
  Serial Driver Source File

- arch/arm/plat-mxc/include/mach/mvf_uart.h
  Definitions for the Serial Driver

3 Existing code to be changed

No modification will be made in source code since this is a newly added driver.
However, the following files are changed for the use of driver.

- drivers/tty/serial/Kconfig
  Serial driver Configuration
- drivers/tty/serial/Makefile
  Serial driver Makefile

4 API of new functions

A standard framework will be implemented as a new serial driver.
Name of functions to be implemented are either mvf_xxxxx() or serial_mvf_xxx().

Functions to be implemented are as follows.

mvf_serial_init
  Initialization processing of the driver

mvf_serial_exit

Termination processing of the driver

serial_mvf_probe

Probe processing of the device

serial_mvf_remove

General deletion processing of the device

- uart_ops

  Create the following functions for uart_ops struct.

  Content of such functions conforms to a standard serial driver. Refer to the "Documentation/serial/driver" in Linux Kernel Source Tree.

  mvf_tx_empty

  As a callback function of tx_empty member in uart_ops struct

  mvf_set_mctrl

  As a callback function of set_mctrl member in uart_ops struct

  mvf_get_mctrl

  As a callback function of get_mctrl member in uart_ops struct

  ,

  mvf_stop_tx

  As a callback function of stop_tx member in uart_ops struct

  mvf_start_tx

  As a callback function of start_tx member in uart_ops struct

  mvf_stop_rx

  As a callback function of stop_rx member in uart_ops struct

  mvf_enable_ms

  As a callback function of enable_ms member in uart_ops struct

mvf_break_ctl

As a callback function of break_ctl member in uart_ops struct

mvf_startup

As a callback function of startup member in uart_ops struct

mvf_shutdown

As a callback function of shutdown member in uart_ops struct

mvf_set_termios

As a callback function of set_termios member in uart_ops struct

mvf_type

As a callback function of type member in uart_ops struct

mvf_release_port

As a callback function of release_port member in uart_ops struct

mvf_request_port

As a callback function of request_port member in uart_ops struct

mvf_config_port

As a callback function of config_port member in uart_ops struct

mvf_verify_port

As a callback function of verify_port member in uart_ops struct

- console struct

  Create the following functions as console struct member functions of serial driver.

  mvf_console_write

  As a callback function of writel function

  mvf_console_setup

  Console initialization

Console Initial Setting

Speed :	9600bps (TBD: device clock-dependent)
Data bits:	8bits
Parity:	none
Stop bits:	1bit

- dev_pm_ops

  Create the following functions as dep_pm_ops struct functions for Power Management. (Power Management will be implemented in Release 4.)

  . suspend = serial_mvf_suspend

  suspend processing (TBD: Release 4)

  .resume = serial_mvf_resume()

  resume processing (TBD: Release 4)

  For Release 1, minimum functions for console I/O will be implemented. UART driver implementation will be done on Release 2.

  Other than above, create functions to be used locally in this source as static function or macro.

5 Expected register settings

This serial driver will implement RS232, and not implement ISO-7816 and CEA709.1B. Therefore, registers to be set are as follows.

UART_BDH: UART Baud Rate Register
UART_BDL: UART, Baud Rate Register
UART_C1: UART Control Register 1
UART_C2: UART Control Register 2
UART_S1: UART Status Register 1
UART_S2: UART Status Register 2
UART_C3: UART Control Register 3
UART_D: UART Data Register

For the one below, set 0.

UART_C4: Control Register

6 Expected functionality and usage

The second release of this serial driver will be implemented with the Asynchronous Serial ports (minimum 2: 1 RS232 /w RTS/CTS/DSR/DTR control signals, 1 RS232 TXD/RDX/RTS/CTS) as a standard UART function.

It will be used as a console or a tty serial device.

This driver will enable all serial ports (6CH).

Serial port No.1 is used for console output.

7 Any other pertinent information

None