Application Notes

Amlogic SDHC Controller User Guide

Revision 0.1

AMLOGIC, Inc.

2518 Mission College Blvd
Santa Clara, CA 95054
U.S.A.
www.amlogic.com

Content

Revision History

Revision	Revised Date	By	Changes
0.1	Aug.20, 2014	Wengang.Lei	Initial release draft

1. Introduction

The SDHC refers to the Secure Digital Host Controller, which can transfer data from the system memory to the SD and MMC bus, and vice versa.

The SD Host Controller (SDHC) has the following features:

1) Supports SD Spec 3.01, eMMC Spec 4.5x , support SDSC, SDHC, SDXC memory card
2) Provides UHS-I both 3.3V and 1.8V signaling,and supports DS, HS, SDR12, SDR25, SDR50, SDR104 speed mode
3) Has AHB slave and master interfaces , supports DMA and PIO mode
4) Back compatible to SD Spec 2.0 and 1.0
5) Only supports SD mode, do not support SPI mode
6) Supports 1-bit, 4-bit, 8-bit bus width mode(8-bit bus width only for MMC)

2. Configuration

The SDHC can support MMC, SD card and SDIO devices. User should do the correct configuration before using it. The configuration file is in the common/arch/boot/dts/amlogic/ folder.

Take k200b as an example:

Open *common/arch/boot/dts/amlogic/meson8_k200b_1G_emmc_sdhc.dtd*
And find

```
sdhc{
        compatible = "amlogic,aml_sdhc"
        dev_name = "aml_sdhc.0";
        reg = <0xc1108e00 0x3c>;
        pinctrl-names = "sdhc_sd_clk_cmd_pins", "sdhc_sd_all_pins",
"sdhc_emmc_clk_cmd_pins", "sdhc_emmc_all_pins", "sdhc_sdio_clk_cmd_pins",
"sdhc_sdio_all_pins";
        pinctrl-0 = <&sdhc_sd_clk_cmd_pins>;
        pinctrl-1 = <&sdhc_sd_all_pins>;
        pinctrl-2 = <&sdhc_emmc_clk_cmd_pins>;
        pinctrl-3 = <&sdhc_emmc_all_pins>;
        pinctrl-4 = <&sdhc_sdio_clk_cmd_pins>;
        pinctrl-5 = <&sdhc_sdio_all_pins>;
        //pinctrl-6 = <&sd_1bit_pins>;
```

```
        // DEVICE="sd"
        // L2 PROP_STR = "status"
        // L3 PROP_U32 = "port"
        // L2 PROP_STR = "pinname"
        // L3 PROP_U32 = "ocr_avail"
        // L2 PROP_STR 3 = "caps"
        // L3 PROP_U32 = "f_min"
        // L3 PROP_U32 = "f_max"
        // L3 PROP_U32 = "f_max_w"
        // L3 PROP_U32 = "max_req_size"
        // L2 PROP_STR = "gpio_dat3"
        // L2 PROP_STR = "jtag_pin"
        // L2 PROP_STR = "gpio_cd"
        // L2 PROP_STR = "gpio_ro"
        // L2 PROP_U32 = "irq_in"
        // L2 PROP_U32 = "irq_out"
        // L2 PROP_U32 = "card_type"
                sd{
                        status = "okay";
                        port = <4>;                     /**0:sdhc_a, 1:sdio_b, 2:sdio_c, 3:sdhc_a,
4:sdhc_b, 5:sdhc_c */
                        pinname = "sd";
                        ocr_avail = <0x00200080>; // 3.3:0x200000, 1.8+3.3:0x00200080
                        caps =
"MMC_CAP_4_BIT_DATA","MMC_CAP_MMC_HIGHSPEED","MMC_CAP_SD_HIGHS
PEED","MMC_CAP_UHS_SDR12","MMC_CAP_UHS_SDR25","MMC_CAP_UHS_SDR
50","MMC_CAP_UHS_SDR104";
                        f_min = <300000>;
                        f_max = <100000000>;
                        max_req_size = <0x20000>;               /**128KB*/
                        gpio_dat3 = "CARD_4";
                        //jtag_pin = "CARD_0";
                        gpio_cd = "CARD_6";
                        //gpio_ro = "GPIOZ_0";
                        irq_in = <3>;
                        irq_out = <5>;
                        card_type = <5>; /* 0:unknown, 1:mmc card(include eMMC), 2:sd
card(include tSD), 3:sdio device(ie:sdio-wifi), 4:SD combo (IO+mem) card, 5:NON sdio
device(means sd/mmc card), other:reserved */
                };

        //$$ DEVICE="emmc"
        //$$ L2 PROP_STR = "status"
        //$$ L3 PROP_U32 = "port"
        //$$ L2 PROP_STR = "pinname"
        //$$ L3 PROP_U32 = "ocr_avail"
        //$$ L2 PROP_STR 4 = "caps"
        //$$ L3 PROP_U32 = "f_min"
        //$$ L3 PROP_U32 = "f_max"
        //$$ L3 PROP_U32 = "f_max_w"
        //$$ L3 PROP_U32 = "max_req_size"
        //$$ L2 PROP_STR = "gpio_dat3"
        //$$ L2 PROP_U32 = "card_type"
```

```
                emmc{
                        status = "okay";
                        port = <5>;                     /**0:sdio_a, 1:sdio_b, 2:sdio_c, 3:sdhc_a,
4:sdhc_b, 5:sdhc_c */
                        pinname = "emmc";
                        ocr_avail = <0x00200080>; // 3.3:0x200000, 1.8+3.3:0x00200080
                        caps =
"MMC_CAP_8_BIT_DATA","MMC_CAP_MMC_HIGHSPEED","MMC_CAP_SD_HIGHS
PEED", "MMC_CAP_NONREMOVABLE","MMC_CAP_ERASE",
"MMC_CAP_HW_RESET"; // MMC_CAP_NEEDS_POLL -- for detect,
MMC_CAP_NONREMOVABLE -- for eMMC/TSD
                        caps2 = "MMC_CAP2_HS200_1_8V_SDR";
                        f_min = <300000>;
                        f_max = <100000000>;
                        max_req_size = <0x20000>;               /**128KB*/
                        gpio_dat3 = "BOOT_3";
                        card_type = <1>; /* 0:unknown, 1:mmc card(include eMMC), 2:sd
card(include tSD), 3:sdio device(ie:sdio-wifi), 4:SD combo (IO+mem) card, 5:NON sdio
device(means sd/mmc card), other:reserved */
                };

        // DEVICE="sdio"
        // L2 PROP_STR = "status"
        // L3 PROP_U32 = "port"
        // L2 PROP_STR = "pinname"
        // L3 PROP_U32 = "ocr_avail"
        // L2 PROP_STR 4 = "caps"
        // L3 PROP_U32 = "f_min"
        // L3 PROP_U32 = "f_max"
        // L3 PROP_U32 = "max_req_size"
        // L2 PROP_U32 = "card_type"
                sdio{
                        status = "okay";
                        port = <3>;                     /**0:sdio_a, 1:sdio_b, 2:sdio_c, 3:sdhc_a,
4:sdhc_b, 5:sdhc_c */
                        pinname = "sdio";
                        ocr_avail = <0x00200080>; // 3.3:0x200000, 1.8+3.3:0x00200080
                        caps =
"MMC_CAP_4_BIT_DATA","MMC_CAP_MMC_HIGHSPEED","MMC_CAP_SD_HIGHS
PEED","MMC_CAP_NONREMOVABLE",
"MMC_CAP_UHS_SDR12","MMC_CAP_UHS_SDR25","MMC_CAP_UHS_SDR50","M
MC_CAP_UHS_SDR104";
                        f_min = <300000>;
                        f_max = <100000000>;
                        max_req_size = <0x20000>;               /**128KB*/
                        card_type = <3>; /* 0:unknown, 1:mmc card(include eMMC), 2:sd
card(include tSD), 3:sdio device(ie:sdio-wifi), 4:SD combo (IO+mem) card, 5:NON sdio
device(means sd/mmc card), other:reserved */
                };
        };
```

The detailed configuration parameters is described as below:
Status : the initial status of device.

Port : the slot where the device in.
Pinname : the name of the device.
Ocr_avail : the voltage which the device can operate on.
Caps : host capabilities.(refer to common/include/linux/mmc/host.h for more details)
Caps2 : more host capabilities.(refer to common/include/linux/mmc/host.h for more details)
F_min : the minimum operating frequency the host can support.
F_max : the maximum operating frequency the host can support.
Max_req_size : the maximum transfer size in one request the device supports.
Card_type : the type of the device.
Gpio_dat3 : the name of gpio which is used as data3.(only valid for SD and eMMC)
Jtag_pin : the name of gpio which is used as jtag detect.(only valid for SD)
Gpio_cd : the name of gpio which is used as card detect.(only valid for SD)
Gpio_ro : the name of gpio which is used as card WP status detect.(only valid for SD)
Irq_in : the irq number for card inserting detect.(only valid for SD)
Irq_out : the irq number for card extracting detect.(only valid for SD)
User should do the correct configuration according to the actual situation of the device.