Case Study:
Some i.MX8X boards eMMC boot failure issue

Yangrui

Aug 2023

Issue Phenomenon

Issue Phenomenon

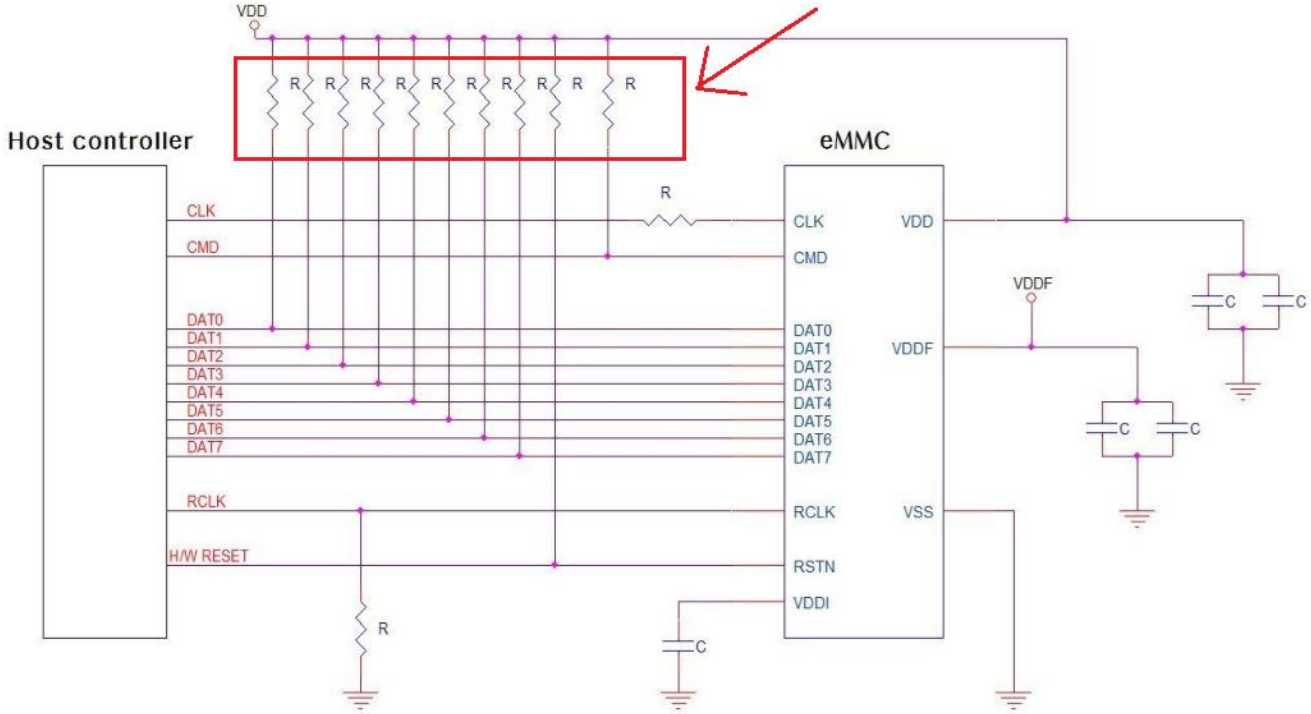
1. Customer some i.MX8X boards black screen and boot failed issue in their manufacture line;
2. The failure rate is 100% in issue boards, but the issue board can boot normally after “heat gun” test, or hand touch eMMC CMD and DATA lines;
3. Add debugging logs, find SW halt in bootloader of their QNX’s uSDHC driver, dead loop in CMD_INHIBIT and DATA_INHIBIT, means uSDHC CMD and DATA lines are always busy and can’t use when this issue happened;

1 CDIHB	Command inhibit (DATA) This status field is generated if either the DAT Line Active or the Read Transfer Active is set to 1. If this field is 0, it indicates that uSDHC can issue the next SD / MMC Command. Commands with a busy signal belong to Command Inhibit (DATA) (for example. R1b, R5b type). Changing from 1 to 0 generates a Transfer Complete interrupt in the Interrupt Status register. **NOTE:** The SD host driver can save registers for a suspend transaction after this field has changed from 1 to 0. 0b - Can issue command that uses the DATA line 1b - Cannot issue command that uses the DATA line
0 CIHB	Command inhibit (CMD) If this status bit is 0, it indicates that the CMD line is not in use and uSDHC can issue a SD / MMC command using the CMD line. This field is set also immediately after the Transfer Type register is written. This field is cleared when the command response is received. Even if the Command Inhibit (DATA) is set to 1, commands using only the CMD line can be issued if this field is 0. Changing from 1 to 0 generates a Command Complete interrupt in the Interrupt Status register. If uSDHC cannot issue the command because of a command conflict error (see Command CRC Error) or because of a Command Not Issued By Auto CMD12 Error, this field remains 1 and the Command Complete is not set. The Status of issuing an auto CMD12 does not show on this field. 0b - Can issue command using only CMD line 1b - Cannot issue command

Issue Analysis

Issue Analysis

Suggest customer to add external pull-up resistors on eMMC_CMD and eMMC_DAT0~7 as following, this issue has gone in i.MX8X issue boards.



Issue Analysis

eMMC SPEC JESD84-B51 & JESD84-A441 request user to add pull-up resistance in DAT0~7 and CMD as following:

$$C_L = C_{HOST} + C_{BUS} + C_{CARD}$$

and requiring the sum of the host and bus capacitances not to exceed 20 pF (see Table 108).

Table 108 — Capacitance

Parameter	Symbol	Min	Typ	Max	Unit	Remark
Pull-up resistance for CMD	R_{CMD}	4.7		100(1)	Kohm	to prevent bus floating
Pull-up resistance for DAT0–7	R_{DAT}	10		100(1)	Kohm	to prevent bus floating
Internal pull up resistance DAT1–DAT7	R_{int}	10		150	kohm	to prevent unconnected lines floating
Bus signal line capacitance	C_L			30	pF	Single card
Single card capacitance	C_{MICRO}			12	pF	For MMC*micro*
	C_{MOBILE}			18		For MMC*mobile* and MMC*plus*
	C_{BGA}		7	12		For BGA
Maximum signal line inductance				16	nH	$f_{pp} \leq 52$ MHz
VDDi capacitor value	C_{REG}(2)	0.1			µF	to stablize regulator output to controller core logics

(1) Recommanded maximum pull-up is 50Kohm for 1.2V and 1.8V interface supply voltages.A 3V part, may use the whole range up to 100Kohms.

(2) Recommended value for C_{REG} might be different between e•MMC device vendors. Please confirm the maximum value and the accuracy of the capacitance with e•MMC vendor because the electrical characteristics of the regulator inside e•MMC is affected by the fluctuation of the capacitance.

JEDEC Standard No. 84-A441
Page 174

12.6 Bus signal levels

10.3.4 Bus signal line load

The total capacitance C_L of each line of the e•MMC bus is the sum of the bus master capacitance C_{HOST}, the bus capacitance C_{BUS} itself, and the capacitance C_{DEVICE} of the Device connected to this line,

$$C_L = C_{HOST} + C_{BUS} + C_{DEVICE}$$

and requiring the sum of the host and bus capacitances not to exceed 20 pF, see Table 200**Error! Reference source not found.Error! Reference source not found.**.

Table 200 — Capacitance and Resistors

Parameter	Symbol	Min	Typ	Max	Unit	Remark
Pull-up resistance for CMD	R_{CMD}	4.7		100(1)	kΩ	to prevent bus floating
Pull-up resistance for DAT0–7	R_{DAT}	10		100(1)	kΩ	to prevent bus floating
Internal pull up resistance DAT1–DAT7	R_{int}	10		150	kΩ	to prevent unconnected lines floating
Bus signal line capacitance	C_L			30	pF	Single Device
Single Device capacitance	C_{DEVICE}			6	pF	
Maximum signal line inductance				16	nH	$f_{pp} \leq 52$ MHz
V_{DDi} capacitor value	C_{REG}(2)	0.1			uF	To stabilize regulator output when target device bus speed mode is either backward-compatible, high speed SDR, high speed DDR, or HS200.
		1			uF	To stabilize regulator output when target device bus speed mode is HS400.
V_{DDi2} capacitor value (e^2•MMC)	C_{REG2}	1			uF	To stabilize internal regulated voltage

Issue Analysis

Bootloader of customer's QNX uSDHC driver don't enable i.MX8X IOMUX pull-up in EMMC0_CMD and EMMC0_DATA0~7, and customer don't add external pull-up resistors in these eMMC related pads, so DAT0 will become always "Card Is Busy" status as following, i.MX8 uSDHC controller will dead loop in CMD_INHIBIT and DATA_INHIBIT.

- R1b responses

  Some commands, like CMD6, may assert the BUSY signal and respond with R1. If the busy signal is asserted, it is done two clock cycles after the end bit of the command. The DAT0 line is driven low, DAT1-DAT7 lines are driven by the Device though their values are not relevant.

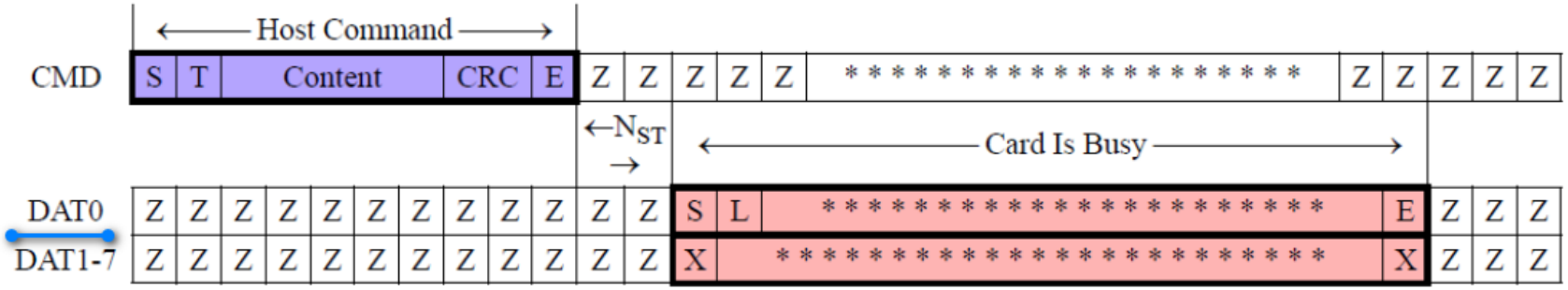


Figure 39 — R1b response timing

Issue Analysis

In i.MX8X Android and Linux, we can find eMMC related pads are configured as IOMUX pull-up as following, so there is no such issue if customer don't add external pull-up resistors in DAT0~7 and CMD when using i.MX8X Android or Linux BSP:

In Uboot:

```
#define ESDHC_PAD_CTRL  ((SC_PAD_CONFIG_NORMAL << PADRING_CONFIG_SHIFT) | (SC_PAD_ISO_OFF << PADRING_LPCONFIG_SHIFT) \
                        | (SC_PAD_28FDSOI_DSE_DV_HIGH << PADRING_DSE_SHIFT) | (SC_PAD_28FDSOI_PS_PU << PADRING_PULL_SHIFT))

#define ESDHC_CLK_PAD_CTRL  ((SC_PAD_CONFIG_OUT_IN << PADRING_CONFIG_SHIFT) | (SC_PAD_ISO_OFF << PADRING_LPCONFIG_SHIFT) \
                        | (SC_PAD_28FDSOI_DSE_DV_HIGH << PADRING_DSE_SHIFT) | (SC_PAD_28FDSOI_PS_PU << PADRING_PULL_SHIFT))
```

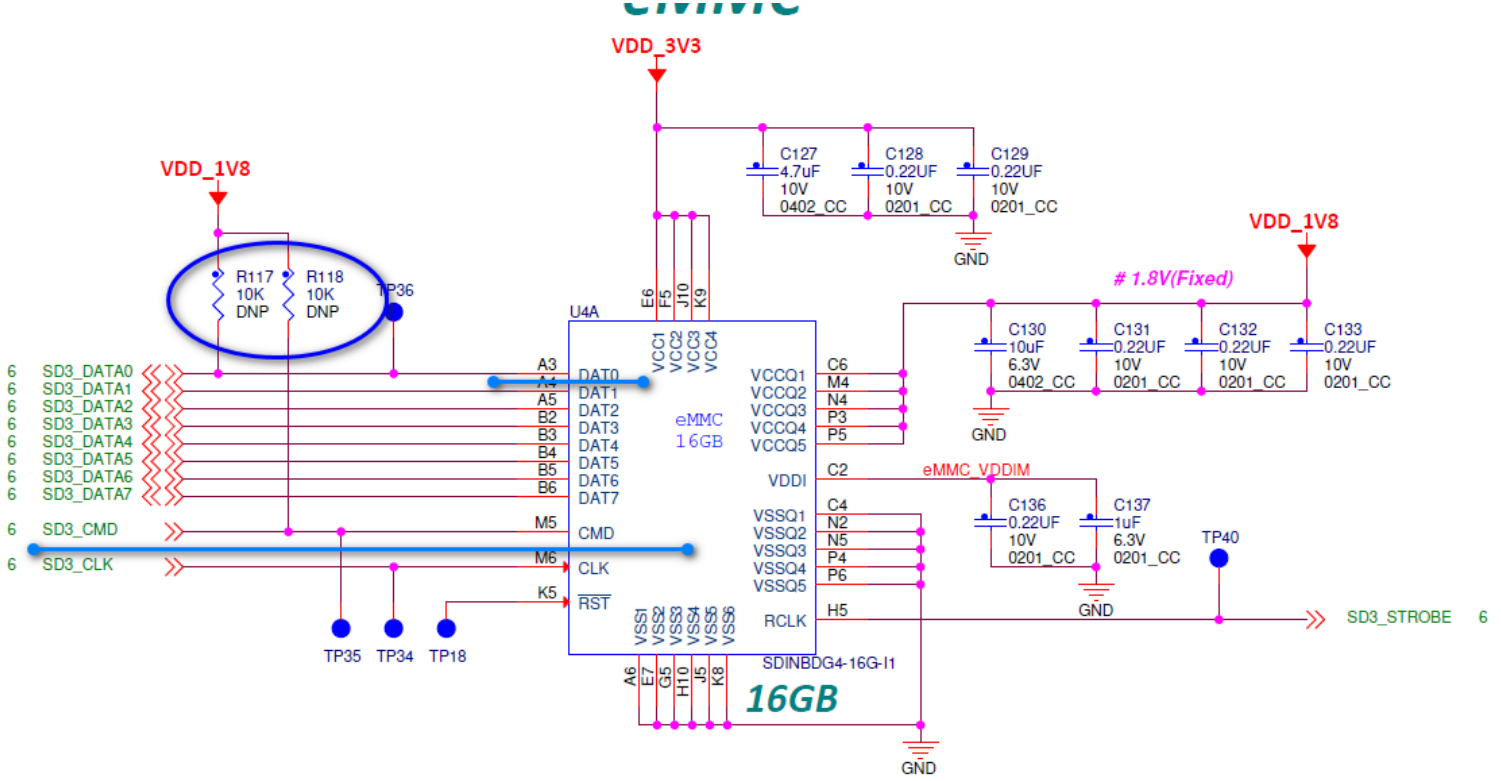
In Kernel device tree:

```
pinctrl_usdhc1: usdhc1grp {
    fsl,pins = <
        SC_P_EMMC0_CLK_CONN_EMMC0_CLK        0x06000041
        SC_P_EMMC0_CMD_CONN_EMMC0_CMD        0x00000021
        SC_P_EMMC0_DATA0_CONN_EMMC0_DATA0    0x00000021
        SC_P_EMMC0_DATA1_CONN_EMMC0_DATA1    0x00000021
        SC_P_EMMC0_DATA2_CONN_EMMC0_DATA2    0x00000021
        SC_P_EMMC0_DATA3_CONN_EMMC0_DATA3    0x00000021
        SC_P_EMMC0_DATA4_CONN_EMMC0_DATA4    0x00000021
        SC_P_EMMC0_DATA5_CONN_EMMC0_DATA5    0x00000021
        SC_P_EMMC0_DATA6_CONN_EMMC0_DATA6    0x00000021
        SC_P_EMMC0_DATA7_CONN_EMMC0_DATA7    0x00000021
        SC_P_EMMC0_STROBE_CONN_EMMC0_STROBE 0x00000041
    >;
};
```

Issue Summary

Issue Summary

Suggest customer to reserve external pull-up resistors on DAT0~7 and CMD pins to comply with eMMC SPEC, at least DAT0 and CMD as following, but we should care about some eMMC don’t have internal pull-up on DAT1~7.




NXP
SECURE CONNECTIONS
FOR A SMARTER WORLD