

配置 i.MX 6 系列处理器上的 USB

作者: 飞思卡尔半导体有限公司

1 简介

i.MX6系列是新一代应用处理器，具有改进的性能和特性。这些处理器上的USB控制器与以往i.MX部件上的USB控制器完全一样，但使用的比较逻辑、收发器和时钟生成机制则不同。

本文对i.MX6处理器上的USB子系统及其配置要求进行深入地介绍。但是，未在文中描述用于USB传输的控制器编程。

目录

1. 简介	1
2. USB 模块概述	2
3. 存储器映像参考	4
4. 针对 USB 操作配置系统	5
5. 修订历史记录	14



1.1 定义、首字母缩略词和缩写词

- HSIC—High Speed Inter Chip （高速片间）
- UTMI—USB 2.0 Transceiver Macro cell Interface （USB 2.0收发器宏单元接口）
- OTG—On-The-Go
- EOP—End of Packet （数据包结尾）
- SOF—Start of Frame （帧（微帧）起始令牌包）

2 USB模块概述

i.MX6系列包括9款器件，主要通过存在或可使能的CPU内核数量来识别。除了这些特性的差异之外，片上外设也有区别。

本章节将描述构成i.MX 6系列处理器USB组合的功能。

2.1 i.MX 6Dual/6Quad与i.MX 6DualLite/Solo

i.MX 6Dual/6Quad、i.MX 6DualLite/Solo、i.MX 6SoloLite和i.MX 6SoloX都具有相同的USB子系统，包括PLL时钟生成器，USB控制器内核和收发器。下面的结构框图显示了基本的功能和时钟。

每个USB控制器内核使用两个独立的时钟：

- AHB/IPG时钟用于CPU或存储器。IPG时钟可以驱动大多数USB内核逻辑，包括寄存器接口。AHB时钟与IPG时钟同步，用于DMA接口进行数据或描述符传输。AHB和IPG时钟均来自于系统PLL，并且共享时钟门控的控制。
- 收发器时钟 (Xcvr_clk) 在收发器中生成。该时钟来自USBn_PLL_480_MHz时钟。收发器时钟与USB总线的数据传输速率同步，由控制器用来提供与协议和端口相关的时钟逻辑。

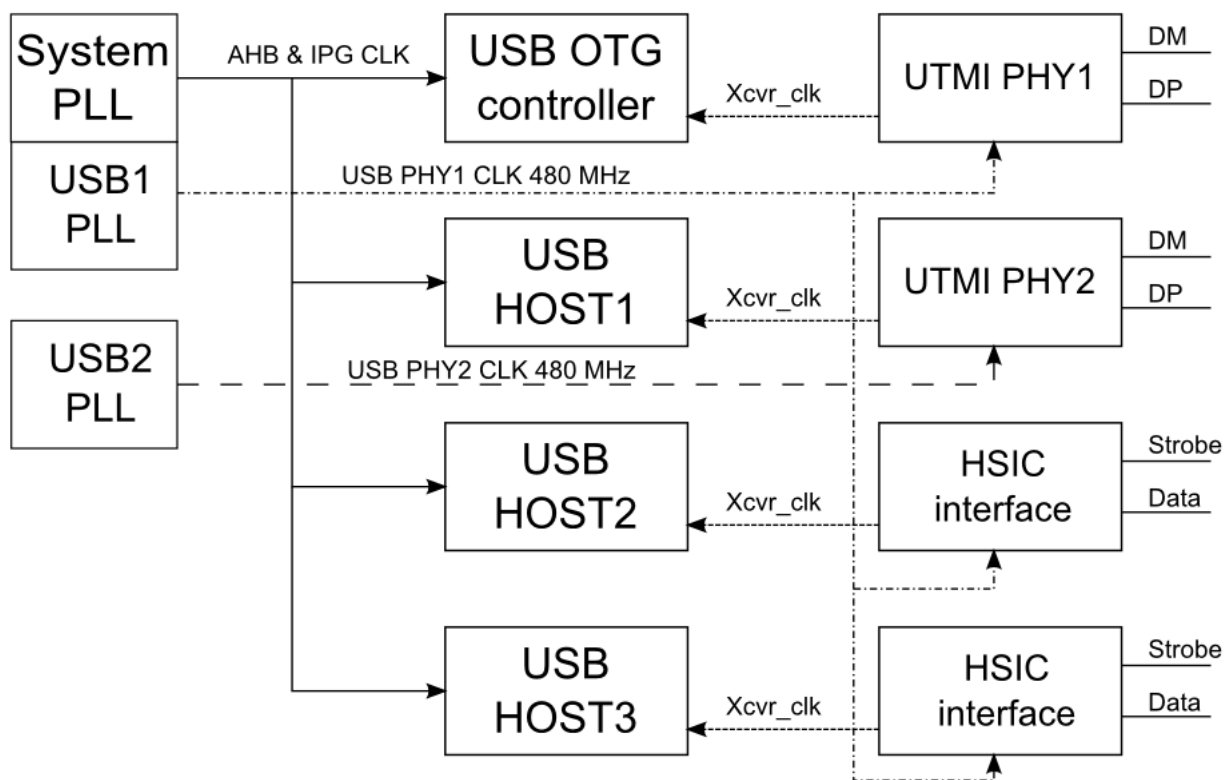


图1. i.MX 6Dual/6Quad与i.MX 6DualLite USB组合

所有USB控制器内核使用相同的AHB/IPG时钟，该时钟通过CCM寄存器CCM_CCGR6的bit 1和bit 0 (usboh3_clk_enable)使能。

USB1 PLL分别为OTG控制器上的UTMI收发器，以及HOST2和HOST3控制器上的HSIC接口提供480 MHz时钟。USB2 PLL与USB1 PLL相同，为UTMI PHY2收发器提供480 MHz时钟。

注意，USB规范要求信号速率为480 MHz +/- 500 ppm。因此，这些时钟必须为480 MHz。

2.2 i.MX 6SoloLite与i.MX 6SoloX

这些处理器上的USB子系统具有两个带有UTMI收发器的OTG控制器内核和一个带有HSIC接口（用于连接板上外设）的主机控制器。双OTG控制器使该处理器可用于需要2个USB设备端口的应用。

i.MX 6SoloLite/6SoloX的PLL、USB控制器和收发器的功能与i.MX 6Dual/6Quad上的功能相同。唯一的区别在于它们的实例化。

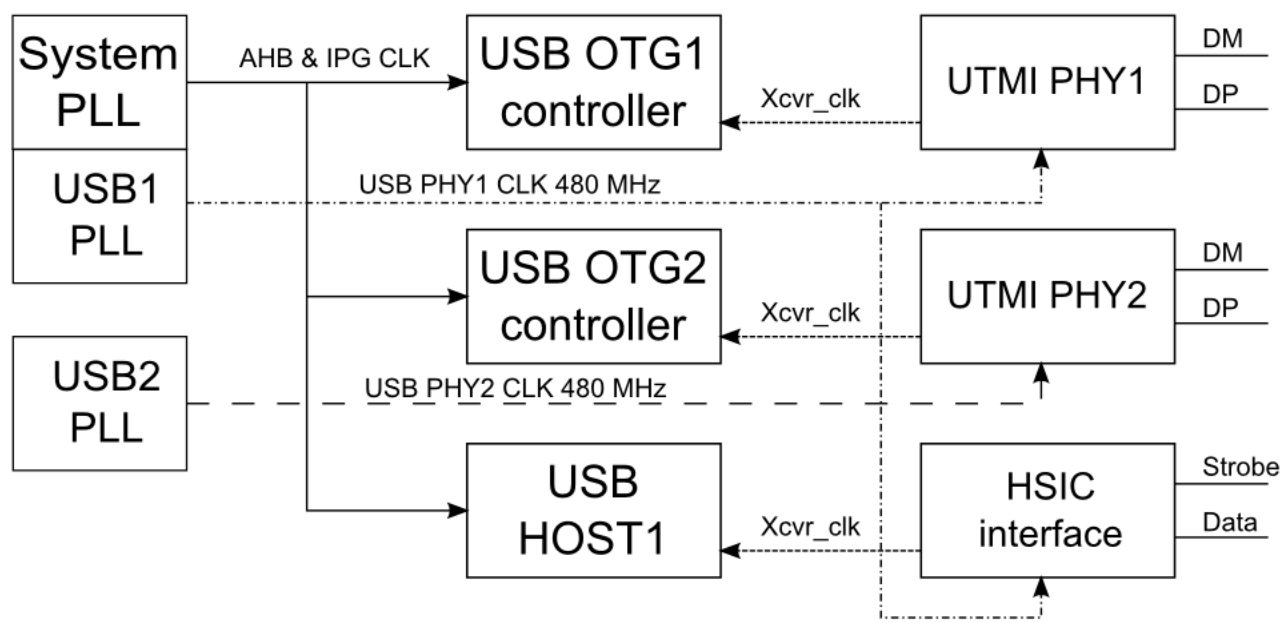


图2. i.MX 6SoloLite与i.MX6SoloX USB组合

这些时钟与i.MX 6Dual/6Quad的时钟相同。源自系统PLL的一个AHB/IPG时钟供所有3个USB控制器使用。OTG1控制器上的UTMI PHY和Host1 HSIC接口共用一个480 MHz PLL，而OTG2控制器则使用一个专用的480 MHz PLL。

3 存储器映像参考

针对USB相关寄存器的存储器映像范围很大，分布于多个模块。

下表提供了模块的基地址，并引用了参考手册中关于寄存器详情的章节。

表 1. i.MX 6Dual/6Quad 与 i.MX 6DualLite

实例	基地址	说明	参考手册章节
USBC_UOG1	0x02184000	USB OTG 控制器，带有 UTMI 收发器	USB 内核存储器映像
USBC_UH1	0x02184200	USB 主机控制器，带有 UTMI 收发器	USB 内核存储器映像
USBC_UH2	0x02184400	USB 主机控制器，带有 HSIC 接口	USB 内核存储器映像
USBC_UH3	0x02184600	USB 主机控制器，带有 HSIC 接口	USB 内核存储器映像
USBC_NON_CORE	0x02184800	集成特定寄存器	USB 非内核存储器映像
USBPHY1	0x020C9000	UTMI PHY，连接到 OTG 控制器	USB PHY 存储器映像
USBPHY2	0x020CA000	UTMI PHY，连接到 H1 控制器	USB PHY 存储器映像
USB_ANALOG	0x020C81A0	充电检测器和 VBUS 检测	USB 模拟存储器映像

表 2. i.MX6 SoloLite 与 i.MX6 SoloX

实例	基地址	说明	参考手册章节
USBC_UOG1	0x02184000	USB OTG 控制器，带有 UTMI 收发器	USB 内核存储器映像
USBC_UOG2	0x02184200	USB OTG 控制器，带有 UTMI 收发器	USB 内核存储器映像
USBC_UH1	0x02184400	USB 主机控制器，带有 HSIC 接口	USB 内核存储器映像
USBC_NON_CORE	0x02184800	集成特定寄存器	USB 非内核存储器映像
USBPHY1	0x020C9000	UTMI PHY，连接到 OTG1 控制器	USB PHY 存储器映像
USBPHY2	0x020CA000	UTMI PHY，连接到 OTG2 控制器	USB PHY 存储器映像
USB_ANALOG	0x020C81A0	充电检测器和 VBUS 检测	USB 模拟存储器映像

4 针对USB操作配置系统

4.1 AHB/IPG时钟

AHB/IPG时钟来自于系统PLL，在配置USB控制器后运行。只需要通过设置CCM_CCGR6寄存器的bit 0和bit 1就可以使能该时钟。当CPU进入一个新的功耗模式时，可通过4种设置自动启动/停止时钟。将这两位置为11b，可以在CPU进入停止模式之前，使时钟保持使能状态。更多详细信息，请参见参考手册中关于CCM的部分。

4.2 USB_n PLL

i.MX6系列处理器包含两个PLL，用于产生USB模块所需的480MHz时钟。虽然该PLL可以配置为其他频率，但当用于产生USB时钟时，就必须配置为480MHz。可以通过CCM_ANALOG_PLL_USB1配置USB1 PLL，通过CCM_ANALOG_PLL_USB2配置USB2 PLL。

PLL必须按照如下顺序进行初始化：

1. 使能PLL。
2. 将POWER位置位，以对PLL供电。
3. 等待PLL_LOCK位有效。
4. 将BYPASS位清零。这将会使PLL从Bypass模式切换到输出模式。
5. 将EN_USB_CLKS置位。这样将使能用于PHY的PLL 9相时钟输出。

附注

在系统进入暂停或者USB进入低功耗模式时，PLL的ENABLE和POWER位应保持置位。当系统进入低功耗模式时，USB控制器也会使PLL进入低功耗模式。

4.3 UTM_n PHY

与i.MX5 USB收发器类似，i.MX6处理器上的UTMI PHY是一个UTMI+ Level 3 PHY。这意味着它支持主机、外设和OTG操作所需的所有功能。

但是，i.MX6系列上的UTMI PHY与i.MX5处理器上的PHY存在明显的差别。下面的章节中将讨论其配置要求。

4.3.1 功耗和时钟

PHY提供几种功耗控制选项。大部分功能可以单独使能/禁用。注意，所有寄存器操作均要求PHY时钟处于使能状态。因此，在进行任何其他编程前，应清零时钟门控信号。

建议顺序为：

- 置位USBPHY_CTRL.SFTRST—将PHY置于复位状态。
- 清零USBPHY_CTRL.CLKGATE—使能PHY中的时钟。
- 清零USBPHY_CTRL.SFTRST—将PHY从复位状态释放。

这时，PHY处于低功耗模式，并为进一步的配置做好准备。

- 清零USBPHY_PWD寄存器—将PHY从低功耗模式释放。

此时PHY可用于通信。

4.3.2 USB PHY低功耗模式

当USB总线挂起时，UTMI收发器可置于低功耗模式。在这种模式中，PHY的所有驱动器和接收器均可关闭。而全速单端接收器需要用来检测总线上的唤醒条件，不能关闭。

i.MX6系列处理器的UTMI收发器在USBPHY_PWD寄存器中有多个功耗控制位，允许关闭收发器的不同部分。这种精细控制仅适用于试验用途。对于一般USB操作，所有电源域必须使能（位清零）。

注意，设置USBPHY_PWD寄存器中的掉电位并不能停止PHY时钟或PLL。

4.3.2.1 进入低功耗模式

在端口或者总线进入休眠模式后，除了需要设置USB控制器的PORTSC寄存器中的PHCD位以外，在i.MX6处理器中，还必须将PHY的PWD位置位，且时钟必须在PHCD位置位后马上开始门控。

该程序在主机和设备模式之间略有不同。

对于主机模式：

- 禁用USBPHY_x_CTRL寄存器中的HS断接检测器。
- 将PORTSC寄存器中的SUSP位置位。
- 等待SUSP位有效（延迟到当前事务结束时该位才能被置有效）
- 将PORTSC寄存器中的PHCD位置位，使PHY进入低功耗模式。
- 将USBPHY_PWD寄存器中的PWD位置位。

- 将USBPHY_CTRL寄存器中的CLKGATE位置位，以停止PHY时钟。
- 当总线进入空闲状态（J-状态）300 μ S 后，将PORTSC中的WKDC位置位，以避免虚假的断接唤醒。

对于设备模式：

- 等待USB控制器发出挂起中断。
- 将PORTSC寄存器中的PHCD位置位。
- 将USBPHY_PWD寄存器中的PWD位置位。
- 将USBPHY_CTRL寄存器中的CLKGATE位置位，以停止PHY时钟。

4.3.2.2 退出低功耗模式

由于软件禁用了时钟并使掉电模式强制进入低功耗模式，因此为了PHY唤醒，软件必须重新使能时钟，并将掉电位清零。

若要实现这一点，在进入低功耗模式后必须使能唤醒中断。在检测到唤醒中断时，软件必须马上执行如下动作，从低功耗挂起中恢复。

- 将对应USB_USB_x_CTRL寄存器中的WIE位清零，以清除中断请求
 - PHCD、CLKGATE和PWD位将在唤醒事件的触发下自动清零

若要在CPU控制下退出低功耗模式（无USB唤醒事件）

- 将PORTSC寄存器中的PHCD位清零
 - CLKGATE与PWD将会自动清零
- 将PORTSC中的FPR置位，以强制恢复
- 等待FPR清零，然后重新使能HS断接检测器

4.3.3 操作设置

一般来说，PHY在操作时应作为UTMI+ Level 3 PHY。在此模式下，PHY将使能所有功能。USB控制器应基于自身的操作模式自动选择适当的PHY设置。

4.3.3.1 通用控制—SBPHY_CTRL寄存器

下面的设置对于基本的USB操作来说十分重要。

ENUTMILEVEL3位和ENUTMILEVEL2位

为使PHY完全支持USB，USBPHY_CTRL寄存器中的ENUTMILEVEL3和ENUTMILEVEL2必须置位。UTMI+ Level 2增加了对直接相连的低速设备的支持。当控制器以主机模式对低速设备（如USB鼠标）进行操作时，需要这种支持。UTMI+ Level 3增加了对直接相连的需支持低速设备的全速集线器的支持。

关于UTMI+的详细信息，请参见UTMI+规范。但是该规范仅对成员公司开放。从usb.org网站上可以获取含有类似相关信息的白皮书“*UTMI+ White Paper*”。

ENHOSTDISCONDETECT位

主机断接（Host-Disconnect）功能向控制器发出通知：在某个活动的高速端口上检测到断接情况。该功能仅在控制器运行于高速主机模式时使用。检测的实现方式：通过高速断接包络检测器测量在40位SOF EOP中最后8位时DM/DP上的差分电平。

此位在控制器用于主机模式时必须置位。

附注

此位必须在USB总线进入挂起模式前清零。一旦USB控制器回复总线后，该位必须马上置位。

4.3.3.2 发送器设置—SBPHY_TX寄存器

发送器中的配置选项可允许更改HS驱动器的电流和高速收发器的终端电阻。在一般情况下，不应该更改默认值。

终端电阻：TXCAL45

可以通过TXCAL45DP和TXCAL45DM位域来配置高速终端电阻。这可以用来弥补由外部阻性元件在信号通路中所造成的损耗，例如ESD保护和扼流圈。但在使用时应谨慎，并通过眼图测量来验证。增大终端电阻值将会增大连接器的DM/DP信号电平；同时还会影响到信号通路上的差分阻抗，导致反射造成的信号失真。因此，不建议更改HS终端电阻。

下面的示例用来说明终端电阻对系统的影响。图3显示的是一个USB驱动器的等效原理图，带有本地和远程终端电阻，以及一个串联电阻，表示数据通路中的外部元件。图4显示的是同一电路，但是远程端为驱动端，而本地端为接收端。

高速驱动器是一个电流源，驱动17.78 mA的电流(I_d)通过本地和远程收发器的终端电阻。在理想情况下，本地和远程终端电阻将会平分驱动电流，使得生成的信号电平为400 mV。

当信号通路上放置了外部元件时，驱动电流将不再被接收器和发送器平分，因为通过接收器支路的电流将变小。

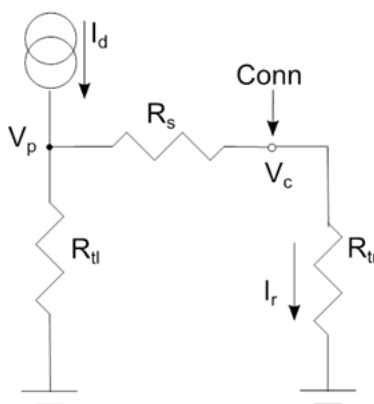


图3. 驱动时的HS终端

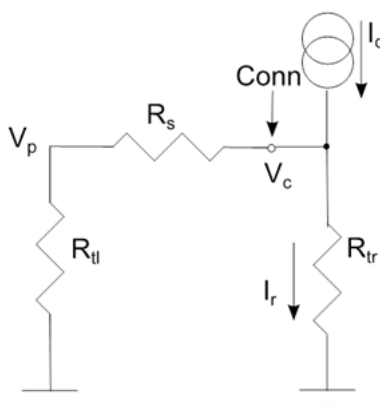


图4. 接收时的HS终端

上图中各符号分别表示：

- V_p —收发器引脚上的电压
- V_c —USB连接器处的电压
- R_{tl} —本地终端电阻，标称值为 $45\ \Omega$
- R_{tr} —远程终端电阻，标称值为 $45\ \Omega$
- R_s —串联电阻
- I_d —发送器的驱动电流为 17.78 mA

USB连接器处的电压：

$$V_c = I_d * R_{tl} * R_{tr} / (R_{tl} + R_{tr} + R_s)$$

器件引脚处的电压：

$$V_p = I_d * R_{tl} * (R_{tr} + R_s) / (R_{tl} + R_{tr} + R_s)$$

当 $R_s = 0$ 且 $R_{tl} = R_{tr}$ 时， V_p 和 V_c 均变为 $I_d * R_t/2$ ，达到一个平衡的工作状态。

示例：数据通路中的串联电阻为8 Ω

这个示例说明了串联电阻值在数据通路中的直流影响，以及调节HS终端电阻的补偿作用。

在默认设置下，器件引脚和USB连接器处的电压分别为：

$$V_C = 17.78 * 45 * 45 / (45 + 45 + 8) = 367 \text{ mV}$$

$$V_P = 17.78 * 45 * (45 + 8) / (45 + 45 + 8) = 433 \text{ mV}$$

从直流角度来看，该数据仍然在规定范围（400 mV +/- 10%）内，但是其他信号伪差的裕度却显著减小。

当试图通过调整终端电阻来补偿Rs上的压降时，系统响应如下：

若要使USB连接器的电压为400 mV，那么通过Rtr的电流必须为½ Id。若要达到这个值，Rtl必须等于Rtr + Rs，使每个支路（图3）分别接收50%的总电流。

应用于本示例中，则Rtl变为45 + 8 = 53 Ω，于是得到的电压如下：

$$V_C = 17.78 * 53 * 45 / (53+45+8) = 400 \text{ mV}$$

$$V_P = 17.78 * 53 * (45 + 8) / (53+45+8) = 471 \text{ mV}$$

此时USB连接器的电压恰好为400 mV，但请注意，驱动器引脚的电平变为471V。这使断接检测器的触发点裕度减小了71 mV。

断接检测器是一个主机端功能，可在SOF传输过程中测量差分电平。用来发出断接信号的触发电平位于525 mV至625 mV之间。

因此，驱动器输入/输出引脚的信号电平越高，断接检测器的噪声裕度就越小。

当从远程设备（图4）接收时，相同的终端设置会得到如下结果：

$$V_C = I_d * R_{tr} * (R_{tl} + R_s) / (R_{tl} + R_{tr} + R_s) = 17.78 * 45 * (53 + 8) / (53 + 45 + 8) = 460 \text{ mV}$$

$$V_C = I_d * R_{tr} * (R_{tl} + R_s) / (R_{tl} + R_{tr} + R_s) = 17.78 * 45 * (53 + 8) / (53 + 45 + 8) = 460 \text{ mV}$$

在接收时器件引脚的电压刚好为400 mV。远程端的发送电平为460 mV，再次减小了主机断接电平的裕度。

更令人关注的问题是，这种调整会对差分阻抗带来哪些影响。

为避免信号失真，整个信号通路中的差分阻抗必须是90 Ω。但在我们的示例中，USB连接器的单端阻抗为61 Ω（53 Ω Rtl加上8 Ω串联电阻），或者差分阻抗为122 Ω。USB连接器阻抗的不连续性产生的反射系数为-0.15，也就是说发出的能量有15%将被反射回来，这会对眼图产生不利影响。

高速驱动器电流：D_CAL

这一字段允许调整高速驱动器的参考电流。降低阻值将会使驱动电流增大，因而传输信号的幅度也会增大。

调整信号幅度可以用来补偿信号通路上的损失。从直流角度来看，其作用将与调整终端电阻的作用相同，但对动态行为的影响明显较小。

用同样的例子来分析其作用：

没有补偿时，电压为：

$$V_c = 17.78 * 45 * 45 / (45 + 45 + 8) = 367 \text{ mV}$$

$$V_p = 17.78 * 45 * (45 + 8) / (45 + 45 + 8) = 433 \text{ mV}$$

在USB连接器处(V_c)需要400 mV，因此电压必须乘以一个系数 $400 / 367 = 1.09$ 或增加9%。

补偿后，实际驱动电流为 $17.78 * 1.09 = 19.38 \text{ mA}$ 。在此情况下，电压为：

$$V_c = 19.38 * 45 * 45 / (45 + 45 + 8) = 400 \text{ mV}$$

$$V_p = 19.38 * 45 * (45 + 8) / (45 + 45 + 8) = 471 \text{ mV}$$

这与调整终端电阻的结果相同。

在接收时，没有补偿，因此电压为：

$$V_p = I_d * R_{tr} * R_{tl} / (R_{tl} + R_{tr} + R_s) = 17.78 * 45 * 45 / (45 + 45 + 8) = 367 \text{ mV}$$

$$V_c = I_d * R_{tr} * (R_{tl} + R_s) / (R_{tl} + R_{tr} + R_s) = 17.78 * 45 * (45 + 8) / (45 + 45 + 8) = 432 \text{ mV}$$

接收端看到仍然是367 mV，远高于为接收器所设的最低电平150 mV。注意，从连接器处来看，这种方式会影响传输包络检测器的触发电平。例如，I/O引脚处为125 mV，而在连接器处变为147 mV。这样可能会导致进行合规性测试时发生接收器灵敏度冲突。在这种情况下，必须调节传输包络检测器的触发电平。

调整驱动电流而不是终端电阻的益处在于差分阻抗。由于终端电阻仍是45 Ω ，USB连接器处的差分阻抗为 $(45 + 8) * 2 = 106 \Omega$ ，可得驱动时的反射系数为-0.08。因此，仅8%的能量被反射，使得眼图中的失真将会更小。

4.3.3.3 接收器设置—SBPHY_RX寄存器

i.MX6 USB收发器中的HS接收器有针对传输包络检测器和断接检测器的可调参数。

传输包络检测器：ENVADJ

传输包络检测器可判定一个有效信号何时出现。在当信号变得很弱，无法可靠接收时，它会关闭接收器（静噪）并PHY中强制空闲状态。USB规范规定静噪电路可以在150 mV激活，而在100 mV及以下必须有效。

包络检测器默认的断路电平为125 mV。可以向下调节25 mV，或向上调节两步（每步12.5 mV）。通常情况下，不应更改该电平。

主机断接检测器：DISCONADJ

与传输包络检测器类似，主机断接检测器也可以限制有效信号电平的范围。当高速设备在高速模式下断开主机时，将会在设备端失去连接到GND的45 Ω 终端，因此所有的驱动电流（17.78 mA）将通过主机端的45 Ω 终端。因此，在传输过程中信号电平变为800 mV。断接检测器将信号电平与预定义的启动点进行比较，如果信号电平超过了断路电平，则将以信号告知主机控制器存在断接情况。

根据USB 2.0规范，断路电平的范围是525 mV-625 mV。i.MX6收发器的默认启动电压是575mV，可以在568 mV-587 mV范围内按照每步6.25 mV进行调节。

4.3.4 电池充电检测

根据USB充电规范中的定义，i.MX6处理器上的UTMI PHY支持电池充电检测。

对于一般USB操作，必须禁用充电检测。充电检测器上的电阻将（启用时）会造成端接电阻的不平衡，并会影响信号质量。

充电检测功能是通过USB_ANALOG_USBN_CHRG_DETECT和USB_ANALOG_USBN_CHRG_DETECT_STAT寄存器中的位来控制。

关于检测过程的详细说明请参见i.MX6系列参考手册中的USB-PHY章节。

4.3.5 VBUS检测

VBUS检测主要用于电池供电的设备，用来检测与主机的连接。

i.MX6的VBUS检测电路可以检测VBUS电平，这样USB设备就可以判断此时是否与主机或充电器连接。同时，当USB设备端没有检测到VBUS电平时，就可以关闭USB控制器的时钟和电源，从而降低功耗。

VBUS检测电路也可以用于检测连接断开信号。当USB总线空闲超过3 s时，USB控制器会产生一个挂起中断，USB设备驱动软件可以等待这个挂起中断，一旦中断产生，就检测VBUS_VALID，如果VBUS有效，则说明主机已挂起总线，但连接却没有断开；如果VBUS无效，则说明连接已被断开。需要注意的是，由于VBUS电容的影响，在连接断开3 s之后，VBUS仍然可能高于VBUS有效阈值，因此，建议系统软件启动DISCHARGE_VBUS以加速放电过程。

4.3.6 HSIC接口

i.MX 6Dual/6Quad和i.MX 6DualLite上的USB控制器USB_UH2和USB_UH3，以及i.MX 6SoloLite和i.MX 6SoloX上的USB_UH1均设计为通过HSIC接口连接板载外设。这些控制器默认配置为通过HSIC接口进行操作。

因为HSIC使用DATA和STROBE信号，而普通USB使用带有时钟恢复功能的差分信号，所以HSIC接口的端口不能直接连接到普通USB端。普通USB设备可以通过一个带有HSIC上行端口的集线器连接到主机控制器的HSIC端口。

关于HSIC的更多信息，请参见规范文件“High-Speed Inter-Chip USB Electrical Specification”（高速芯片间USB电气规范），此文件可在USB 2.0规范的压缩档案内找到。

4.3.7 i.MX 6Dual/6Quad与i.MX 6DualLite上的IOMUX配置

HSIC端口没有专用的I/O引脚。该接口基于低压CMOS，与LPDDR2 DRAM I/O类似。若要使用HSIC端口，必须对IOMUX以及I/O引脚进行配置。下面的表3中显示了针对i.MX 6Dual/6Quad与i.MX 6DualLite HSIC操作的I/O设置。

表 3. i.MX 6Dual/6Quad 与 i.MX 6DualLite HSIC I/O 设置

HSIC 信号	IOMUX 寄存器	HSIC 模式	寄存器	地址
H3_STROBE	IOMUXC_SW_MUX_CTL_PAD_RGMII_RXC	ALT 0	0000 0000h	020E 0084h
H3_DATA	IOMUXC_SW_MUX_CTL_PAD_RGMII_RX_CTL	ALT 0	0000 0000h	020E 006Ch
H2_STROBE	IOMUXC_SW_MUX_CTL_PAD_RGMII_TX_CTL	ALT 0	0000 0000h	020E 0074h
H2_DATA	IOMUXC_SW_MUX_CTL_PAD_RGMII_TXC	ALT 0	0000 0000h	020E 0058h
I/O 引脚设置				
H3_STROBE	IOMUXC_SW_PAD_CTL_PAD_RGMII_RXC		0001 3030h	020E 0398h
H3_DATA	IOMUXC_SW_PAD_CTL_PAD_RGMII_RX_CTL		0001 3030h	020E 0380h
H2_STROBE	IOMUXC_SW_PAD_CTL_PAD_RGMII_TX_CTL		0001 3030h	020E 0388h
H2_DATA	IOMUXC_SW_PAD_CTL_PAD_RGMII_TXC		0001 3030h	020E 036Ch
	IOMUXC_SW_PAD_CTL_GRP_DDR_TYPE_RGMII	DDR_SEL = 10b	0000 0030h	020E 0790h

4.3.8 i.MX 6SoloLite的IOMUX配置

由于I/O接口的区别，i.MX 6SoloLite上的设置略有不同。表4显示了用于该端口HSIC操作的寄存器设置。

表 4. i.MX 6SoloLite HSIC I/O 设置

HSIC 信号	IOMUX 寄存器	HSIC 模式	寄存器	地址
H_DATA	IOMUXC_SW_MUX_CTL_PAD_HSIC_DAT	ALT0	0000 0000h	020E 0154h
H_STROBE	IOMUXC_SW_MUX_CTL_PAD_HSIC_STROBE	ALT0	0000 0000h	020E 0158h
I/O 引脚设置				
H_DATA	IOMUXC_SW_PAD_CTL_PAD_HSIC_DAT	LPDDR2-CMOS	0008 3030h	020E 0444h
H_STROBE	IOMUXC_SW_PAD_CTL_PAD_HSIC_STROBE	LPDDR2-CMOS	0008 3030h	020E 0448h

4.3.9 i.MX 6SoloX的IOMUX配置

表5 显示了用于该端口HSIC操作的寄存器设置。

表 5. i.MX 6SoloX HSIC I/O 设置

HSIC 信号	IOMUX 寄存器	HSIC 模式	寄存器	地址
USB_H_DATA	IOMUXC_SW_MUX_CTL_PAD_USB_H_DATA	Alt 0	0000 0000h	020E 02A4h
USB_H_STROBE	IOMUXC_SW_MUX_CTL_PAD_USB_H_STROBE	Alt 0	0000 0000h	020E 02A8h
—	I/O 引脚设置	—	—	—
USB_H_DATA	IOMUXC_SW_PAD_CTL_PAD_USB_H_DATA	LPDDR2-CMOS	0008 3030h	020E 05ECh
USB_H_STROBE	IOMUXC_SW_PAD_CTL_PAD_USB_H_STROBE	LPDDR2-CMOS	0008 3030h	020E 05F0h

5 修订历史记录

表6 列出了此应用笔记的修订历史记录。

表 6. 修订历史记录

修订编号	日期	重大变更
修订版 0	10/2012	初始版本。
修订版 1	02/2015	添加了关于 i.MX 6SoloX 的引用资料。

How to Reach Us:

Home Page:

Freescale.com

Web Support:

Freescale.com/support

本文档中的信息仅供系统和软件实施方使用 Freescale 产品。本文并未明示或者暗示授予利用本文档信息进行设计或者加工集成电路的版权许可。Freescale 保留对此处任何产品进行更改的权利，恕不另行通知。

Freescale 对其产品在任何特定用途方面的适用性不做任何担保、表示或保证，也不承担因为应用程序或者使用产品或电路所产生的任何责任，明确拒绝承担包括但不限于后果性的或附带性的损害在内的所有责任。Freescale 的数据表和/或规格中所提供的“典型”参数在不同应用中可能并且确实不同，实际性能会随时间而有所变化。所有运行参数，包括“经典值”在内，必须经由客户的技术专家对每个客户的应用程序进行验证。Freescale 未转让与其专利权及其他权利相关的许可。Freescale 销售产品时遵循以下网址中包含的标准销售条款和条件: freescale.com/SalesTermsandConditions.

Freescale, the Freescale logo, and Kinetis are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. ARM is the registered trademark of ARM Limited. ARM Cortex-M0+ is the trademark of ARM Limited. All other product or service names are the property of their respective owners.

© 2015 Freescale Semiconductor, Inc.

© 2015 飞思卡尔半导体有限公司。

Document Number: AN4589
Rev 1, 03/2015

