

Configuring USB on i.MX 6 Series Processors

by *Freescale Semiconductor, Inc.*

1 Introduction

The i.MX6 family of application processors is a new generation with improved performance and features. The USB controllers on these processors are identical to the ones on older i.MX parts, but the companion logic, transceivers and clock generation are different.

This document gives an in-depth overview of the USB subsystem on i.MX6 processors and its configuration requirements. It does not describe how to program the controller for USB transfers.

Contents

1. Introduction	1
2. USB module overview	2
3. Memory map reference	4
4. Configuring the system for USB operation	5
5. Revision History	14



1.1 Definitions, Acronyms, and Abbreviations

- HSIC—High Speed Inter Chip
- UTMI—USB 2.0 Transceiver Macro cell Interface
- OTG—On-The-Go
- EOP—End of Packet
- SOF—Start of Frame (a token packet indicating the start of a USB (micro)-frame)

2 USB module overview

The i.MX 6 series consists of five families of parts that are primarily identified by the number of CPU cores that are either present or can be enabled. However, besides these characteristic differences, the on-chip peripherals also differ.

This chapter describes the functions that make up the USB complex of the i.MX 6 Series processors.

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

The i.MX 6Dual/6Quad, i.MX 6DualLite/Solo, i.MX 6SoloLite and i.MX 6SoloX have identical USB subsystems including the PLLs for clock generation, USB controller core and transceivers. The block diagram below shows the basic functions and clocks.

Each USB controller core uses two independent clocks:

- The AHB/IPG clocks are used by the logic that interfaces to the CPU or memory. The IPG clock drives most of the USB core logic, including the register interface. The AHB clock is synchronous to the IPG clock and is used by the DMA interface for data and descriptor transfers. Both AHB and IPG clocks are sourced from the System PLL and share controls for clock gating.
- The transceiver clock (Xcvr_clk) is generated in the transceiver. It is derived from the USBn_PLL_480_MHz clock. The transceiver clock is synchronized to the data rate of the USB bus and is used by the controller to clock protocol and port related logic.

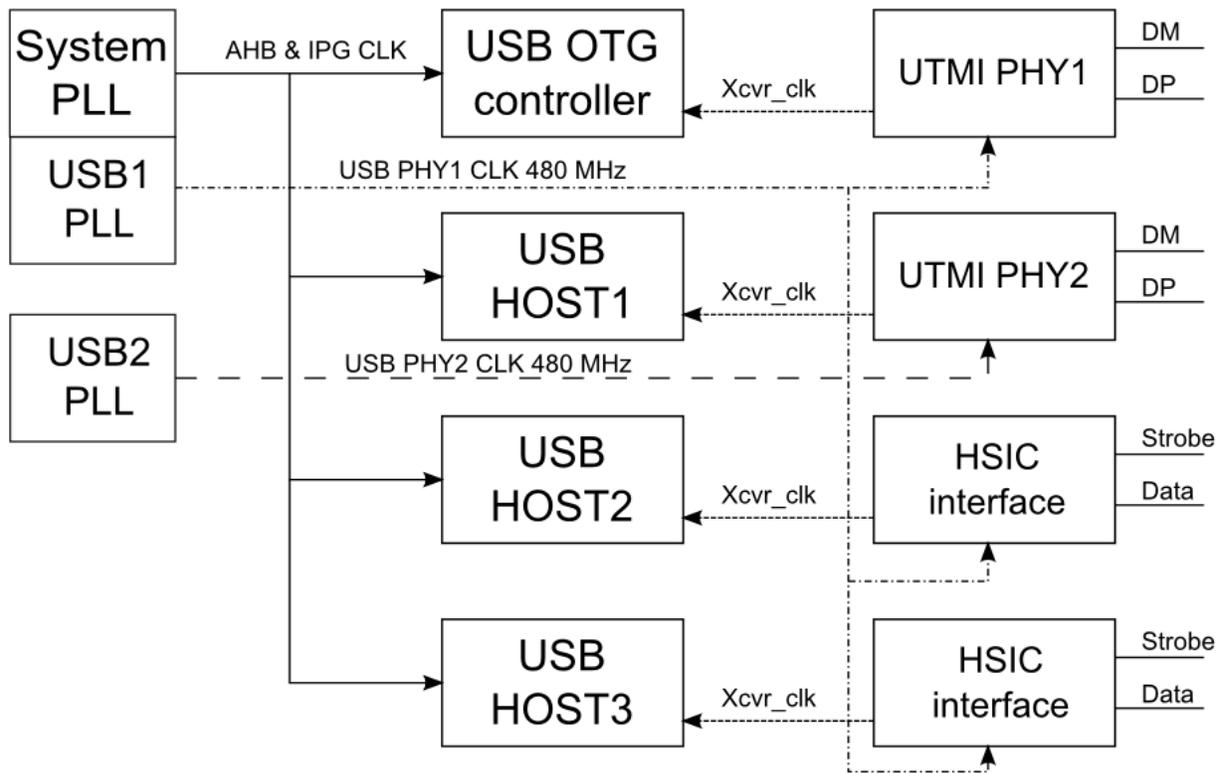


Figure 1. i.MX 6Dual/6Quad and i.MX 6DualLite USB Complex

All USB controller cores use the same AHB/IPG clock. This clock is enabled in the CCM register CCM_CCGR6, bits 1:0 (usboh3_clk_enable).

USB1 PLL provides the 480 MHz clock for the UTMI transceiver on the OTG controller, as well as for the HSIC interfaces on the HOST2 and HOST3 controllers. USB2 PLL is identical to USB1 PLL and provides the 480 MHz clock for the UTMI PHY2 transceiver.

Note that the USB specification requires the signal rate to be 480 MHz +/- 500 ppm. Therefore, these clocks must be 480 MHz.

2.2 i.MX 6SoloLite and i.MX6SoloX

The USB subsystem on these processors features two OTG controller cores with UTMI transceivers and 1 Host-only controller with HSIC interface for connections to on-board peripherals. The Dual OTG controller makes this processor suitable for applications requiring 2 USB device ports.

The PLLs, USB controller and Transceiver functions are identical to the ones on i.MX 6Dual/6Quad. The only difference is their instantiation.

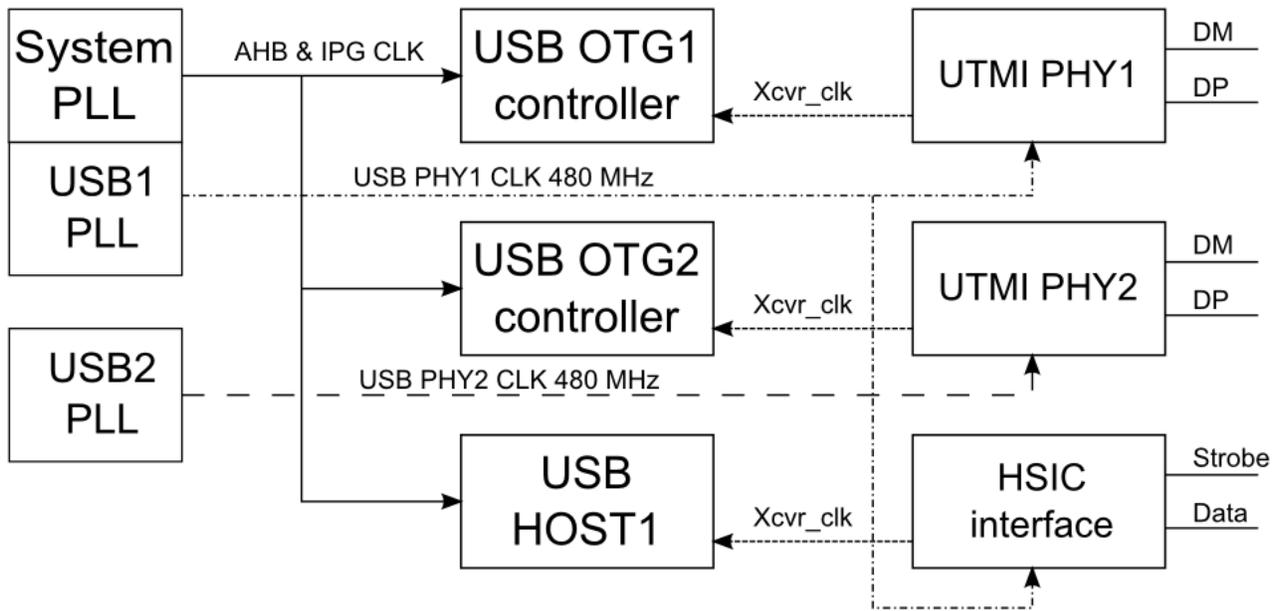


Figure 2. i.MX 6SoloLite and i.MX6SoloX USB Complex

The clocks are identical to i.MX 6Dual/6Quad . There is a single AHB/IPG clock, sourced from the System PLL, for all 3 USB controllers. One 480 MHz PLL, common for the UTMI PHY on the OTG1 controller and the Host1 HSIC interface and one dedicated 480 MHz PLL for the OTG2 controller.

3 Memory map reference

The memory map for USB related registers is quite extensive and spread over several modules.

The tables below provide the base address of the module and a reference to the registers details section in the reference manual.

Table 1. i.MX 6Dual/6Quad and i.MX 6DualLite

Instance	Base Address	Description	Reference Manual Chapter
USBC_UOG1	0x02184000	USB OTG controller with UTMI Transceiver	USB Core Memory Map
USBC_UH1	0x02184200	USB Host-only controller with UTMI transceiver	USB Core Memory Map
USBC_UH2	0x02184400	USB Host-only controller with HSIC interface	USB Core Memory Map
USBC_UH3	0x02184600	USB Host-only controller with HSIC interface	USB Core Memory Map
USBC_NON_CORE	0x02184800	Integration specific registers	USB Non-Core Memory Map
USBPHY1	0x020C9000	UTMI PHY connected to OTG controller	USB PHY Memory Map
USBPHY2	0x020CA000	UTMI PHY connected to H1 controller	USB PHY Memory Map
USB_ANALOG	0x020C81A0	Charger detector and VBUS detection	USB Analog Memory Map

Table 2. i.MX6 SoloLite and i.MX6 SoloX

Instance	Base Address	Description	Reference Manual Chapter
USBC_UOG1	0x02184000	USB OTG controller with UTMI Transceiver	USB Core Memory Map
USBC_UOG2	0x02184200	USB OTG controller with UTMI transceiver	USB Core Memory Map
USBC_UH1	0x02184400	USB Host-only controller with HSIC interface	USB Core Memory Map
USBC_NON_CORE	0x02184800	Integration specific registers	USB Non-Core Memory Map
USBPHY1	0x020C9000	UTMI PHY connected to OTG1 controller	USB PHY Memory Map
USBPHY2	0x020CA000	UTMI PHY connected to OTG2 controller	USB PHY Memory Map
USB_ANALOG	0x020C81A0	Charger detector and VBUS detection	USB Analog Memory Map

4 Configuring the system for USB operation

4.1 AHB/IPG clock

The AHB/IPG clocks, derived from the system PLL will be running by the time the USB controller is configured. All that needs to be done is to enable the clock in the CCM module by setting bits 1, 0 in the CCM_CCGR6 register. The 4 possible settings allow to automatically start/stop the clock when the CPU enters a new power mode. Setting 11b will keep the clock enabled until the CPU enters stop mode. For more details, refer to the CCM section in the Reference Manual.

4.2 USB n PLL

i.MX6 processors contain 2 PLLs that are intended for use with the USB modules and produce a low-jitter 480 MHz clock. Although the PLLs are configurable for other frequencies, they must be producing 480 MHz when the connected USB module is used. The USB PLLs are configured through the registers CCM_ANALOG_PLL_USB1, for USB1 PLL, and CCM_ANALOG_PLL_USB2 for USB2 PLL.

The PLL must be initialized in the following order:

1. Enable PLL.
2. Set POWER bit to power-up the PLL.
3. Wait for PLL_LOCK bit to assert.
4. Clear BYPASS bit. This will switch the PLL output from Bypass clock to PLL output clock.
5. Set EN_USB_CLKS. This enables the PLL's 9-phase clock output for the PHY.

NOTE

The PLL's ENABLE and POWER bits should remain set during system suspend or USB low-power suspend. The USB controller will place the PLL in low-power mode when it enters low-power suspend.

4.3 UTM_n PHY

Similar to the i.MX5 USB transceivers, the UTMI PHY on i.MX6 processors is a UTMI+ level 3 PHY. This means that it supports all the functionality needed for Host, Peripheral and OTG operation.

The UTMI PHY on i.MX6 family is, however, significantly different from the PHY on i.MX5 processors. The configuration requirements are discussed in the following sections.

4.3.1 Power and clocks

The PHY provides several options for power control. Most of the functions can be enabled/disabled separately. Note that all registers operations require that the PHY clock is enabled. Therefore, the clock gate must be cleared before any other programming can be performed.

The recommended sequence is:

- Set USBPHY_CTRL.SFTRST—Put the PHY in reset.
- Clear USBPHY_CTRL.CLKGATE—Enables the clock in the PHY
- Clear USBPHY_CTRL.SFTRST—Release the PHY from reset.

At this point, the PHY is in low-power mode and ready for further configuration.

- Clear USBPHY_PWD register—Release PHY from low-power mode

The PHY can now be used for communication.

4.3.2 USB PHY low power mode

UTMI transceivers can be placed in low-power mode when the USB bus is suspended. In this mode, all drivers and receivers in the PHY can be turned off. The full-speed single-ended receivers are needed to detect wakeup conditions on the bus and cannot be turned off.

The i.MX6 UTMI transceiver has multiple power control bits in the USBPHY_PWD register that allow for turning-off different sections of the transceiver. This granularity is only useful for test purposes. For normal USB operation, all power domains must be enabled (bits cleared).

Note that setting the power-down bits in the USBPHY_PWD register does not stop the PHY clock or PLL.

4.3.2.1 Entering low power mode

Apart from setting the PHCD bit in the USB controller's PORTSC register after the port or the bus is suspended, on i.MX6 processors the PHY's PWD bits must also be set and the clock gated immediately after setting the PHCD bit.

The procedure is slightly different between host and device mode.

For host mode operation:

- Disable the HS disconnect detector in the USBPHY_x_CTRL register.
- Set the SUSP bit in the PORTSC register.
- Wait for SUSP bit to assert (bit assertion will be delayed until the end of the current transaction).

- Set the PHCD bit in the PORTSC register to place the PHY in low-power mode.
- Set PWD bits in USBPHY_PWD register.
- Set CLKGATE in USBPHY_CTRL register to stop the PHY clock.
- Set WKDC bit in PORTSC 300 μ S after the bus becomes idle (J-state) to avoid a false disconnect wakeup.

For device mode:

- Wait for suspend interrupt from the USB controller.
- Set the PHCD bit in the PORTSC register.
- Set PWD bits in USBPHY_PWD register.
- Set CLKGATE in USBPHY_CTRL register to stop the PHY clock.

4.3.2.2 Exiting low power mode

Since software disables the clocks and forces power-down mode to enter low-power mode, software must also re-enable the clocks and clear the Power Down bits for the PHY to wake-up.

To do so, the wakeup interrupt must be enabled after entering low-power mode. Upon detection of the wakeup interrupt, software must perform the following actions to resume from low-power suspend.

- Clear the WIE bit in the corresponding USB_USB_x_CTRL register to clear the interrupt request — PHCD, CLKGATE and PWD bits will be cleared automatically by the wakeup event

To exit low power mode under CPU control (no USB wakeup event)

- Clear PHCD bits in the PORTSC register — CLKGATE and PWD will auto clear
- Set FPR in PORTSC to force resume
- Wait for FPR to clear, then re-enable HS disconnect detector

4.3.3 Operational settings

In general, it is expected that the PHY operates as a UTMI+ level 3 PHY. In this mode, the PHY will have all capabilities enabled. The USB controller will automatically select the appropriate PHY setting based on its operational mode.

4.3.3.1 General controls—USBPHY_CTRL register

The following settings are essential for basic USB operation.

ENUTMILEVEL3 and ENUTMILEVEL2 Bits

To enable full USB support on the PHY, bits ENUTMILEVEL3 and ENUTMILEVEL2, in the USBPHY_CTRL register, must be set. UTMI+ Level 2 adds support for directly connected low-speed devices. This is needed when the controller operates in host mode for operation with low-speed devices such as USB mice. UTMI+ Level 3 adds support for directly connected full-speed hubs that need to support low-speed devices.

The details of UTMI+ can be found in the UTMI+ specification. However, this specification is only available to member companies. A whitepaper “*UTMI+ White Paper*,” with similar information is available from the usb.org web site.

ENHOSTDISCONDETECT bit

The Host-Disconnect function informs the controller that a disconnect condition was detected on an active high-speed port. This function is only used when the controller operates in high-speed host mode. The detection is done by means of the high-speed disconnection envelope detector which measures the differential level on DM/DP during the last 8 bits of the 40-bit SOF EOP.

This bit must be set when the controller is used in host mode.

NOTE

This bit must be cleared before placing the USB bus in SUPEND mode. Once the USB controller has resumed the bus, this bit must be set.

4.3.3.2 Transmitter settings—USBPHY_TX register

The configuration options in the Transmitter allow for changing the HS driver current and the termination resistance of the high-speed transceiver. In general, the default values should not be changed.

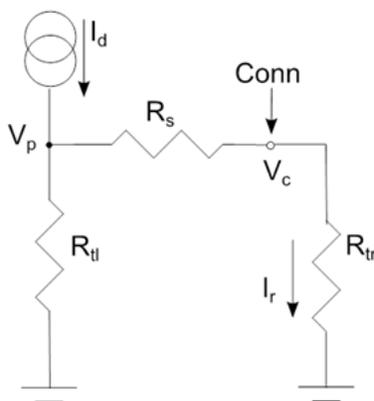
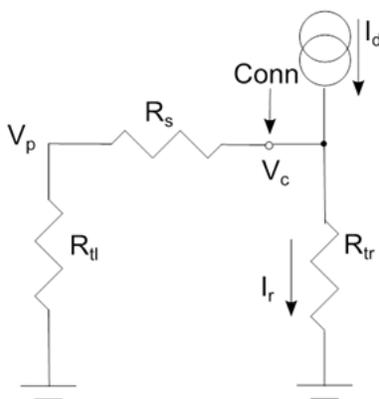
Termination resistors: TXCAL45

Bit fields TXCAL45DP and TXCAL45DM allow for changing the resistance of the high-speed termination. This can be used to compensate for losses caused by external resistive components in the signal path, for example ESD protection and chokes. However this should be used carefully and verified with an eye diagram measurement. Increasing the termination resistor value will increase the DM/DP signals level at the connector; however, it will also affect the differential impedance of the signal path and contribute to signal distortion due to reflections. It is therefore not recommended to change the HS termination resistors.

The following example illustrates how the system responds to a change in termination resistance. [Figure 3](#) shows the equivalent schematic of a USB driver with local and remote termination resistors and a series resistor representing external components in the data path. [Figure 4](#) shows the same circuit but now with the remote side driving and the local side receiving.

The high-speed driver is a current source that drives 17.78 mA (I_d) through the termination resistors of the local and the remote transceivers. In the ideal case, the driver current will be evenly split over the local and remote termination resistors and this will result in a 400 mV signal level.

When external components are placed in the signal path, the driver current is no longer evenly split between receiver and transmitter as less current will flow through the receiver branch.


Figure 3. HS termination when Driving

Figure 4. HS termination when receiving

In the above schematics, the symbols represent:

- V_p —Voltage at the transceiver pin.
- V_c —Voltage at the USB connector.
- R_{tl} —Local termination resistor, nominal 45Ω .
- R_{tr} —Remote termination resistor, nominal 45Ω
- R_s —Combined series resistor
- I_d —Transmitter driver current, 17.78 mA

The voltage at the USB connector is given by:

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

The voltage at the device pin is:

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

With $R_s = 0$ and $R_{tl} = R_{tr}$, both V_p and V_c become $I_d * R_t/2$, giving a balanced operation.

Example: 8 Ω serial resistance in the data path

This example illustrates the DC effects of serial resistance in the data path and compensation with adjusting the HS termination resistors.

With default settings, the voltage levels at the device pin and USB connector become:

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

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

From a DC perspective, this is still within spec (400 mV +/- 10%), however the margin for other signal artifacts is significantly reduced.

When attempting to compensate for the voltage drop over R_s by trimming the termination resistor then, the system will respond as follows:

To have 400 mV at the USB connector, there must be $\frac{1}{2} I_d$ flowing through R_{tr} . To achieve this, R_{tl} must be equal to $R_{tr} + R_s$ such that each branch (Figure 3) receives 50% of the total current.

Applied to this example, R_{tl} becomes $45 + 8 = 53 \Omega$ which gives the following voltages:

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

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

The voltage at the USB connector is now nicely 400 mV, but notice that the level at the driver pin becomes 471V. This reduces the margin to the Disconnect Detector trigger point with 71 mV.

The disconnect detector is a host-side function that measures the differential level during transmission of the SOF. The trigger level to signal a disconnection is between 525 mV and 625 mV.

Therefore, the higher the signal level becomes on the driver's I/O pin, the less noise margin there is for the disconnect detector.

When receiving from the remote device (Figure 4), and the same termination settings, the following results are obtained:

$$V_P = I_d * R_{tr} * R_{tl} / (R_{tl} + R_{tr} + R_s) = 17.78 * 45 * 53 / (53 + 45 + 8) = 400 \text{ mV}$$

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

There are 400 mV at the device pin during reception which is exactly right. The remote side's transmit level is 460 mV, again reducing the margin to the host-disconnect level.

More of a concern is the effect of this trimming on the differential impedance.

The differential impedance must be 90 Ω throughout the signal path to avoid signal distortion. In our example, however, the impedance at the USB connector is 61 Ω single-ended (53 Ω tuned R_{tl} plus 8 Ω serial resistance) or 122 Ω differential. This discontinuity in impedance at the USB connector has a reflection coefficient of -0.15 which means that 15% of the transmitted energy will be reflected. This will have a negative impact on the eye diagram.

High-speed driver current: D_CAL

This field allows for trimming the current reference for the high-speed driver. Reducing the resistance will increase the driver current and hence the amplitude of the transmitted signal will increase.

Trimming the signal amplitude can be used to compensate for losses on in the signal path. From a DC point of view, the effect will be the same as trimming the termination resistors, but the impact on the dynamic behavior is significantly less.

The same example is used as before to analyze the effect:

Without compensation the voltage levels are:

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

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

At the USB connector (V_c), 400 mV are needed so that the voltage must increase by a factor of $400 / 367 = 1.09$ or 9%.

When compensated, the actual driver current will be $17.78 * 1.09 = 19.38$ mA. The voltages are in that case:

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

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

This is identical to what was found for trimming the termination resistors.

When receiving, there will be no compensation, so the voltages are:

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

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

The receiver still sees 367 mV which is well above the 150 mV minimum level for the receiver. Note that this does affect the trigger levels of the transmission envelope detector as seen at the connector. As an example, 125 mV at the I/O pins becomes 147 mV at the connector. This may lead to receiver sensitivity violations during compliance testing. In that case, the trigger level of the transmission envelope detector must be adjusted.

The benefit of trimming the driver current instead of the termination resistors is the differential impedance. As the termination resistors are still 45 Ω , the differential impedance seen at the USB connector is now $(45 + 8) * 2 = 106 \Omega$, giving a reflection factor of -0.08 when driving. Therefore, only 8% of the energy will be reflected and hence there will be less distortion in the eye diagram.

4.3.3.3 Receiver settings—USBPHY_RX register

The HS receiver in the i.MX6 USB transceiver has adjustable parameters for the transmission envelope detector and the disconnect detector.

Transmission envelope detector: ENVADJ

The transmission envelop detector determines when a valid signal is present. It turns the receiver off (squench) when the signal becomes too low for reliable reception and forces and idle condition in the PHY. The USB specification defines that the squench circuit may activate at 150 mV and must be active at 100 mV and below.

The default trip level for the envelop detector is 125 mV. It can be adjusted 25 mV down and 2 steps of 12.5 mV up. This level should normally not be changed.

Host disconnect detector: DISCONADJ

The Host Disconnect detector is similar to transmission envelope detector in that it limits the range of valid signal levels. When a high-speed device disconnects from a host while operating in high-speed mode, the 45 Ω termination to GND will be missing on the device side and hence all the driver current (17.78 mA) will flow through the host side 45 Ω termination. Therefore, the signal level will become 800 mV during transmission. The disconnect detector compares the signal level to a predefined trip point and if the signal level exceeds the trip level, it will signal a disconnection to the host controller.

The range for the disconnect trip level is 525 mV to 625 mV per USB 2.0 specification. The i.MX6 transceiver's default trip point is at 575 mV. It can be adjusted in steps of 6.25 mV between 568 mV and 587 mV.

4.3.4 Battery charger detection

The UTMI PHY on i.MX6 processors has support for battery charger detection as defined in the USB Charging specification.

The charger detector must be disabled for normal USB operation. The resistors in the charger detector will (when enabled) create an imbalance in the termination resistance and will affect the signal quality.

The charger detection function is controlled by bits in the USB_ANALOG_USBn_CHRG_DETECT and USB_ANALOG_USBn_CHRG_DETECT_STAT registers.

For a description of the detection process, see the USB-PHY Chapter in each of the i.MX6 family reference manuals.

4.3.5 VBUS detection

VBUS detection is primarily used in battery powered devices to detect a connection to a host.

The VBUS detection circuit on i.MX6 allows devices to sense the presence of VBUS and hence the connection to a host or charger. With this capability, the USB controller's clocks can be stopped and transceiver power can be turned off, allowing for greater power savings.

The VBUS detection can be used to detect a disconnection from the host, the device driver software waits for the controller to issue a suspend interrupt. This interrupt is generated 3 ms after the USB bus becomes idle. Then it checks for VBUS_VALID. If VBUS is valid, then the host suspended the bus and the cable is still connected. If VBUS is not valid, then the cable was disconnected. Note that VBUS may remain above the VBUS Valid threshold for much longer than 3 ms when the VBUS capacitors are not being discharged. System software may therefore enable the DISCHARGE_VBUS function to speed-up the discharge process.

4.3.6 HSIC interface

USB controllers USB_UH2 and USB_UH3 on i.MX 6Dual/6Quad and i.MX 6DualLite and USB_UH1 on i.MX 6SoloLite and i.MX 6SoloX are intended for connections to on-board peripherals with HSIC interfaces. The controllers are by default configured for operation with the HSIC interface.

HSIC interface ports cannot connect directly to regular USB ports because HSIC signaling uses DATA and STROBE signals where as regular USB uses differential data with clock recovery. Regular USB devices can be connected to controllers with HSIC ports through a hub with HSIC upstream port.

For more information on HSIC, please refer to the specification document “High-Speed Inter-Chip USB Electrical Specification” which can be found in the zip archive of the USB 2.0 specification.

4.3.7 IOMUX configuration on i.MX 6Dual/6Quad and i.MX 6DualLite

The HSIC ports do not have dedicated I/O pins. The interface is based on low-voltage CMOS, similar to LPDDR2 DRAM I/O. To use the HSIC ports, the IOMUX as well as the I/O pads must be programmed. Table 3 below shows the I/O settings for i.MX 6Dual/6Quad and i.MX 6DualLite HSIC operation.

Table 3. i.MX 6Dual/6Quad and i.MX 6DualLite HSIC I/O Settings

HSIC signal	IOMUX register	Mode for HSIC	Register	Address
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 Pad Settings				
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 IOMUX configuration for i.MX 6SoloLite

The settings on i.MX 6SoloLite are slightly different because of the differences in I/O interface. Table 4 shows the register settings for HSIC operation of the port.

Table 4. i.MX 6SoloLite HSIC I/O Settings

HSIC signal	IOMUX register	Mode for HSIC	Register	Address
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 Pad settings				
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 IOMUX configuration for i.MX 6SoloX

Table 5 shows the register settings for HSIC operation of the port.

Table 5. i.MX 6SoloX HSIC I/O settings

HSIC signal	IOMUX register	Mode for HSIC	Register	Address
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 pad settings	—	—	—
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 Revision History

Table 5 provides a revision history for this application note.

Table 6. Document Revision History

Rev. Number	Date	Substantive Change(s)
Rev. 0	10/2012	Initial release.
Rev. 1	02/2015	Added i.MX 6SoloX references.

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Information in this document is provided solely to enable system and software implementers to use Freescale products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document.

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: freescale.com/SalesTermsandConditions.

Freescale and the Freescale logo are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. All other product or service names are the property of their respective owners. ARM is the registered trademark of ARM Limited. ARM Cortex™-A9 is the trademark of ARM Limited.

© Freescale Semiconductor, Inc., 2015. All rights reserved.

Document Number: AN4589
Rev. 1
03/2015

