Errata to "UTMI+ Low Pin Interface Specification Revision 1.0"

This document lists the fixes, clarifications, and additions required to the ULPI Specification, Revision 1.0. Each errata has a sequence number and a type. An explanation of the types is given in Table 1. Changes to text in the ULPI specification are highlighted in red. Check the http://www.ulpi.org website for the latest errata.

Errata Type	Errata Type Description
Fix	Fixes a problem in the specification. The purpose of a fix is to correct specification behavior that would otherwise cause a failure.
Clarification	Clarifies existing behavior in the specification. The purpose of a clarification is to expand the existing description, providing further information for implementers.
Addition	Adds new behavior to the specification. The purpose of an addition is to detail new and optional functionality.

Table 1 – Errata types

Errata 01 – Incorrect hold time numbers

Type:
Fix.

Issue:
The hold time numbers given in Table 5 were calculated wrongly, and given the wrong polarity.

Resolution:
The correct numbers must be given in Table 5.

Documentation Changes:
Table 5 must be replaced with the table below to update the hold time numbers.

Parameter	Symbol	Min	Max	Units
Output clock				
Setup time (control in, 8-bit data in)	T_{SC}, T_{SD}		6.0	ns
Hold time (control in, 8-bit data in)	T_{HC}, T_{HD}	0.0		ns
Output delay (control out, 8-bit data out)	T_{DC}, T_{DD}		9.0	ns
Setup time (4-bit data in) (optional)	T_{SDD}		3.0	ns
Hold time (4-bit data in) (optional)	T_{HDD}	-0.8		ns
Output delay (4-bit data out) (optional)	T_{DDD}		4.0	ns
Input clock (optional)				
Setup time (control in, 8-bit data in)	T_{SC}, T_{SD}		3.0	ns
Hold time (control in, 8-bit data in)	T_{HC}, T_{HD}	1.5		ns
Output delay (control out, 8-bit data out)	T_{DC}, T_{DD}		6.0	ns
Setup time (4-bit data in)	T_{SDD}		2.5	ns
Hold time (4-bit data in)	T_{HDD}	0.8		ns
Output delay (4-bit data out)	T_{DDD}		3.5	ns

Table 5 – ULPI Interface Timing

Errata 02 – Sending RX CMDs during USB packet transmit and receive

Type:
Clarification.

Issue:
It is not clear what the PHY should do when an RX CMD needs to be sent when the ULPI bus is carrying USB transmit or receive data.

Resolution:
An RX CMD should never abort USB transmit or receive data. Instead, the PHY must flag the RX CMD internally and send an RX CMD when **nxt** is de-asserted or when the receive packet has completed.

Documentation Changes:
For clarification on USB packet receive, the following text should be appended to the end of paragraph 1 in section 3.8.2.4.

"All RX CMD changes during the USB packet receive must be signaled when **nxt** is low. If **nxt** is never low during the packet receive, all RX CMD changes must be replaced with a single RX CMD update that is sent at the end of the USB packet receive, when the ULPI bus is available. The RX CMD update must always convey the current RX CMD values, not a previous or old value."

Similarly, text in the USB packet transmit sections 3.8.2.1 and 3.8.2.2 should also be updated to reflect the text below.

"All RX CMD changes during the USB packet transmit must be replaced with a single RX CMD update that is sent at the end of the USB transmit, when the ULPI bus is available. The RX CMD update must always convey the current RX CMD values, not a previous or old value."

Errata 03 – De-assertion of dir when exiting FS/LS Serial Mode

Type:
Clarification.

Issue:
In section 3.10.3 “Exiting FsLsSerialMode”, the text describing the de-assertion of **dir** when exiting serial mode with the clock running is wrong, and does not match Figure 52. This also applies to exiting Carkit mode, where it is defined that “Entering and exiting Carkit mode is identical to Serial mode”.

Resolution:
The **dir** signal does not need to be de-asserted in the cycle after the link asserts **stp** to exit serial mode, but can be de-asserted 1 or more cycles after the link asserts **stp**.

Documentation Changes:
The second paragraph in section 3.10.3 should be replaced with the text below. Figure 53 should be replaced with the figure below.

“If the clock is running, the Link signals the PHY to exit ***FsLsSerialMode*** by asserting **stp**. The PHY will de-assert **dir** 1 or more cycles after it detects **stp** asserted, as shown in Figure 53. The Link de-asserts **stp** in the cycle following the de-assertion of **dir**. Like Low Power Mode, there is a single cycle of bus turnaround on **data** in the cycle following the de-assertion of **dir**. During the turnaround cycle, the value on **data** is not valid. The PHY stops driving the serial mode signals immediately before the turnaround cycle.”

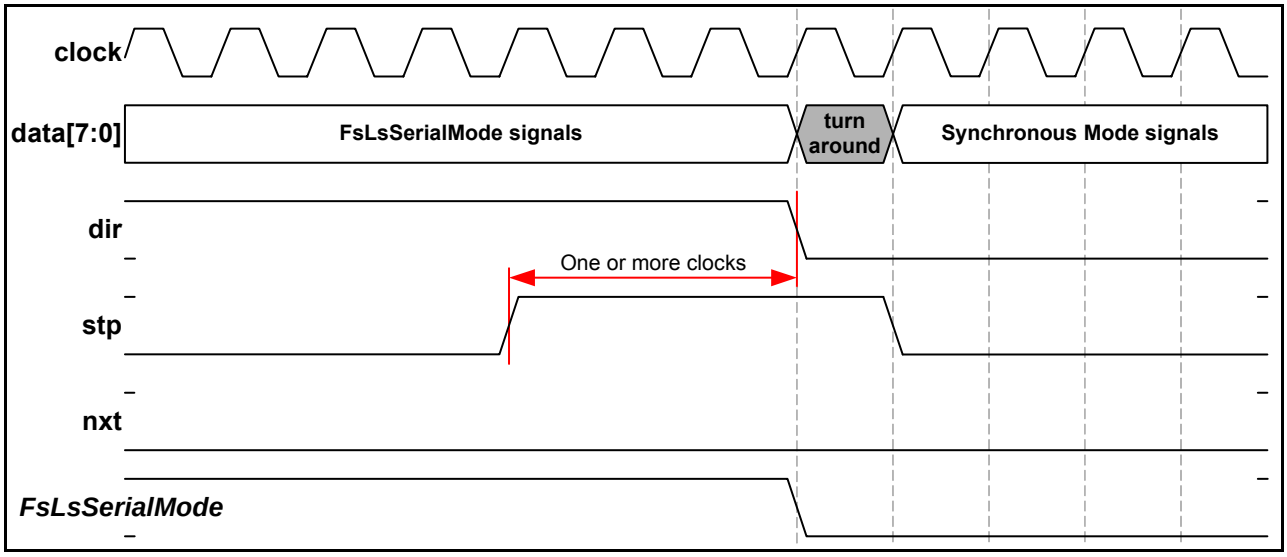


Figure 53 - Interface behaviour when exiting Serial Mode and clock is running

Errata 04 – Control and monitoring of internal charge pump and external VBUS supplies

Type:
Fix, clarification, and addition.

Issue:
Control of the internal VBUS charge pump or external VBUS supply causes an unnecessary burden on link hardware. Also, when the VBUS supply is external to the PHY, it is not clear how the Link is informed when VBUS has an over-current condition.

Resolution:
This issue applies only to controllers with host capability.
For control of the VBUS source, the definition of ***DrvVbus*** and ***DrvVbusExternal*** can be simplified, leading to a more practical link architecture.
For monitoring an external VBUS supply, the external VBUS valid or fault indicator should be routed to an input pin on the PHY, replacing any internal VA_VBUS_VLD comparator. This removes the need for extra pins on the link and keeps all VBUS indicators in-band to the ULPI bus.

Documentation Changes:
This change effects the external pin descriptions, the OTG Operations section and the Immediate Register Set description.

The following paragraphs and table should replace the paragraph titled **DrvVbusExternal** in section 3.3. The optional **VbusValidExternal** pin should also be shown on Figure 5.

DrvVbusExternal and ExternalVbusIndicator

The PHY may optionally control an external VBUS power source via the optional pin **DrvVbusExternal**. For example, the external supply could be a charge pump or 5V power supply controlled using a power switch. The external supply is controlled by the ***DrvVbus*** and the optional ***DrvVbusExternal*** bits in the **OTG Control** register. The polarity of the **DrvVbusExternal** output pin is implementation dependent.

If control of an external VBUS source is provided the PHY may optionally provide for a VBUS power source feed back signal on the optional pin **ExternalVbusIndicator**. If this pin is provided, the use of the pin is defined by the optional control bits in the **OTG Control** and **Interface Control** registers. See Section 3.8.6.3 for further detail.

The following text will be appended to section 3.8.6.

Vbus Power Control (internal and external)

The link turns on VBUS by setting the ***DrvVbus*** bit in the OTG Control Register. If the Vbus supply is external to the PHY, the link sets ***DrvVbus*** and the optional ***DrvVbusExternal*** bit in the **OTG Control** register. The VBUS control settings are detailed in Table XX.

DrvVbus	*DrvExternalVbus*	**Power Source used**
0	X	Internal and external VBUS power sources disabled.
1	0	Internal VBUS charge pump enabled.
1	1	External 5V VBUS supply enabled.

Table XX – OTG Control register power control bits

Section 3.8.6.3 will be replaced with the following text, table, and figure.

Vbus Comparator Thresholds

If the VBUS power supply is external to the PHY, and the external supply provides a signal indicating when VBUS is valid, it is recommended that this signal be an input to the PHY on an optional pin **ExternalVbusIndicator**, and that the state of that pin be reflected to the Link via the VA_VBUS_VLD ≤ VBUS indication in the RX CMD byte. The optional ***UseExternalVbusIndicator*** bit in the **OTG Control** register selects between the internal and external VbusValid indicators.

To support industry standard USB power control devices, the PHY may optionally support two additional bits in the **Interface Control** register, ***IndicatorPassThru*** and ***IndicatorComplement***. These two bits allow the optional **ExternalVbusIndicator** pin to interoperate with either a power valid signal or an over-current fault output from the power control device, and to adapt to either active high or active low signals from the power control device. When a power fault signal is provided on the **ExternalVbusIndicator** pin, the PHY must use a logical combination of the output from the internal VbusValid comparator and the external power fault signal to generate the VA_VBUS_VLD ≤ VBUS indication. Table YY defines the use of the ***UseExternalVbusIndicator***, ***IndicatorPassThru*** and ***IndicatorComplement*** register bits to control the use of the **ExternalVbusIndicator** input pin and the internal VbusValid comparator output to generate the VA_VBUS_VLD ≤ VBUS indication in the RX CMD byte. Table YY also indicates typical applications of each setting. Figure YY provides a graphical representation of the logical combination of the internal and external VbusValid sources, and how the control register bits effect the VA_VBUS_VLD ≤ VBUS indication in the RX CMD byte. The ***UseExternalVbusIndicator***, ***IndicatorPassThru*** and ***IndicatorComplement*** control register bits are individually optional. The PHY may implement any combination of the optional control bits, however, if the control bits are implemented they must provide the function defined in Table YY. If any of the control bits are not implemented it is the responsibility of the PHY to define how the optional **ExternalVbusIndicator** pin effects the state of the VA_VBUS_VLD ≤ VBUS indication in the RX CMD byte.

Typical Application	*UseExternal VbusIndicator*	*Indicator PassThru*	*Indicator Complement*	RxCmd VBUS Valid source
OTG Device	0	don't care	don't care	Internal VA_VBUS_VLD comparator.
	1	1	0	External active high VA_VBUS_VLD signal.
	1	1	1	External active low VA_VBUS_VLD_N signal.
	1	0	1	External active high power fault signal qualified with internal VA_VBUS_VLD comparator.
	1	0	0	External active low power fault signal qualified with internal VA_VBUS_VLD comparator.
Standard Host	1	1	0	External active high power fault signal.
	1	1	1	External active low power fault signal.
Standard Peripheral	0	don't care	don't care	Internal VA_VBUS_VLD comparator.[1]

Table YY – RxCmd VBUS Valid over-current conditions

[1] A standard peripheral should not use Vbus Valid to begin operation. The internal VbusValid may not indicate Vbus is valid on the 5th hub tier, which is allowed to be as low as 4.375V. Therefore the peripheral should use Session Valid.

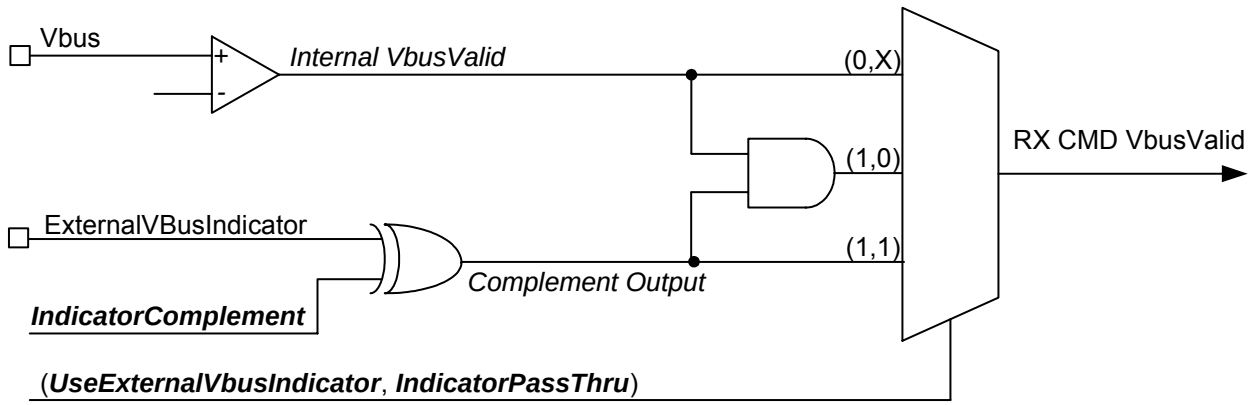


Figure YY – RxCmd $V_{A\_VBUS\_VLD}$ ≤ V_{BUS} indication source

Depending on the application, the link should enable or disable the appropriate Vbus interrupts. Example settings for typical applications are given in Table ZZ.

Application	VbusValid[2]	SessValid	SessEnd
Standard Host	Yes	No	No
Standard Peripheral	No	Yes	No
OTG A-Device	Yes	Yes	No
OTG B-Device	No	Yes	Yes

Table ZZ – Vbus indicators in the RXCMD required for typical applications

[2] The VbusValid indicator in the RXCMD comes from either the internal VbusValid comparator, or the external Vbus indicator input.

A new register bit is defined in the OTG Control register replacing reserved bit 7 as follows:

Field name	Bit	Access	Reset	Description
UseExternal VbusIndicator	7	rd/wr/s/c	0b	Tells the PHY to use an external VBUS over-current indicator. This bit is optional. Refer to 3.8.6.3. 0b: Use the internal OTG comparator (VA_VBUS_VLD) or internal VBUS valid indicator (default). 1b: Use external VBUS valid indicator signal

Two new register bits are defined in the Interface Control register replacing reserved bits 5 and 6 as follows:

Field name	Bit	Access	Reset	Description
IndicatorComplement	5	rd/wr/s/c	0b	Tells the PHY to invert the **ExternalVbusIndicator** input signal, generating the *Complement Output*. Refer to 3.8.6.3 and Figure YY for more details. 0b: PHY will not invert **ExternalVbusIndicator** signal (default). 1b: PHY will invert **ExternalVbusIndicator** signal.
IndicatorPassThru	6	rd/wr/s/c	0b	Controls whether the *Complement Output* is qualified with the *Internal VbusValid* comparator before being used in the Vbus State in the RXCMD. Refer to 3.8.6.3 and Figure YY for more details. 0b: *Complement Output* signal is qualified with the *Internal VbusValid* comparator. 1b: *Complement Output* signal is not qualified with the *Internal VbusValid* comparator.

The ***DrvVbus*** and ***DrvVbusExternal*** are redefined in the OTG Control register as follows.

Field name	Bit	Access	Reset	Description
DrvVbus	5	rd/wr/s/c	0b	Signals the internal charge pump or external supply to drive 5V on Vbus. 0b : do not drive Vbus (default). 1b : drive 5V on Vbus
DrvVbusExternal	6	rd/wr/s/c	0b	Selects between the internal and the external 5V Vbus supply. This bit is optional and does not need to be present if only one Vbus power source is supported. 0b : Drive Vbus using the internal charge pump (default). Support of an internal charge pump is optional. 1b : Drive Vbus using external supply. Support of an external Vbus power source is optional.

The following change will be made to the Vbus State bits in the RXCMD byte definition in 3.8.1.2.

3:2	Vbus State	Encoded Vbus Voltage state				
		Value	**V_{BUS} Voltage**	**SessEnd**	**SessValid**	**VbusValid**[3]
		00	$V_{BUS} < V_{B\_SESS\_END}$	1	0	0
		01	$V_{B\_SESS\_END} \leq V_{BUS} < V_{SESS\_VLD}$	0	0	0
		10	$V_{SESS\_VLD} \leq V_{BUS} < V_{A\_VBUS\_VLD}$	X	1	0
		11	$V_{A\_VBUS\_VLD} \leq V_{BUS}$	X	X	1

[3] The VbusValid indicator in the RXCMD comes from either the internal VbusValid comparator, or the external Vbus indicator input.

Errata 05 – How to send Test_J and Test_K signaling / Updated Table 34

Type:
Clarification.

Issue:
It is not clear how a ULPI PHY can send Test_J and Test_K signaling, as required by USB section 7.1.20.

Also, the contents of Table 34 (Signalling Modes) are difficult to read and are causing confusion for implementers.

Resolution:
For peripherals, Test_J and Test_K signalling is already detailed in UTMI section 5.16.1. The link controller places the PHY into HS mode, and disables bit stuffing and NRZI encoding. The link controller then transmits a constant stream of 1's to generate the Test_J, or a constant stream of 0's to generate the Test_K. The same applies to peripheral, host, and OTG devices.

The contents of Table 34 should be clarified using groupings.

Documentation Changes:
The following changes should be made to Table 34 in section 4.4.

Signalling mode	Register Settings					Resistor Settings				
	XcvrSelect	*TermSelect*	*OpMode*	*DpPulldown*	*DmPulldown*	rpu_dp_en	rpu_dm_en	rpd_dp_en	rpd_dm_en	hsterm_en
General Settings										
Tristate Drivers	XXb	Xb	01b	Xb	Xb	0b	0b	0b	0b	0b
Power-up or Vbus < Vth(SESSEND)	01b	0b	00b	1b	1b	0b	0b	1b	1b	0b
Host Settings										
Host Chirp	00b	0b	10b	1b	1b	0b	0b	1b	1b	1b
Host Hi-Speed	00b	0b	00b	1b	1b	0b	0b	1b	1b	1b
Host Full Speed	X1b	1b	00b	1b	1b	0b	0b	1b	1b	0b
Host HS/FS Suspend	01b	1b	00b	1b	1b	0b	0b	1b	1b	0b
Host HS/FS Resume	01b	1b	10b	1b	1b	0b	0b	1b	1b	0b
Host Low Speed	10b	1b	00b	1b	1b	0b	0b	1b	1b	0b
Host Low Speed Suspend	10b	1b	00b	1b	1b	0b	0b	1b	1b	0b
Host Low Speed Resume	10b	1b	10b	1b	1b	0b	0b	1b	1b	0b
Host Test_J/Test_K	00b	0b	10b	1b	1b	0b	0b	1b	1b	1b
Peripheral Settings										
Peripheral Chirp	00b	1b	10b	0b	0b	1b	0b	0b	0b	0b
Peripheral Hi-Speed	00b	0b	00b	0b	0b	0b	0b	0b	0b	1b

Peripheral Full Speed	01b	1b	00b	0b	0b	1b	0b	0b	0b	0b
Peripheral HS/FS Suspend	01b	1b	00b	0b	0b	1b	0b	0b	0b	0b
Peripheral HS/FS Resume	01b	1b	10b	0b	0b	1b	0b	0b	0b	0b
Peripheral Low Speed	10b	1b	00b	0b	0b	0b	1b	0b	0b	0b
Peripheral Low Speed Suspend	10b	1b	00b	0b	0b	0b	1b	0b	0b	0b
Peripheral Low Speed Resume	10b	1b	10b	0b	0b	0b	1b	0b	0b	0b
Peripheral Test_J/Test_K	00b	0b	10b	0b	0b	0b	0b	0b	0b	1b
OTG device, Peripheral Chirp	00b	1b	10b	0b	1b	1b	0b	0b	1b	0b
OTG device, Peripheral Hi-Speed	00b	0b	00b	0b	1b	0b	0b	0b	1b	1b
OTG device, Peripheral Full Speed	01b	1b	00b	0b	1b	1b	0b	0b	1b	0b
OTG device, Peripheral HS/FS Suspend	01b	1b	00b	0b	1b	1b	0b	0b	1b	0b
OTG device, Peripheral HS/FS Resume	01b	1b	10b	0b	1b	1b	0b	0b	1b	0b
OTG device, Peripheral Test_J/Test_K	00b	0b	10b	0b	1b	0b	0b	0b	1b	1b

Table 34 – Upstream and downstream signaling modes

Errata 06 – SuspendM setting when exiting Low Power Mode

Type:
Clarification.

Issue:
In section 3.9.3, Figure 46 and Figure 47, of the ULPI specification, it is not clear if it is the responsibility of the link or the PHY to set the SuspendM register bit to a 1 after exiting Low Power Mode. Unless a clear responsibility is stated, both link and PHY might assume that the other sets SuspendM to 1b, resulting in a hung bus that is always in Low Power Mode.

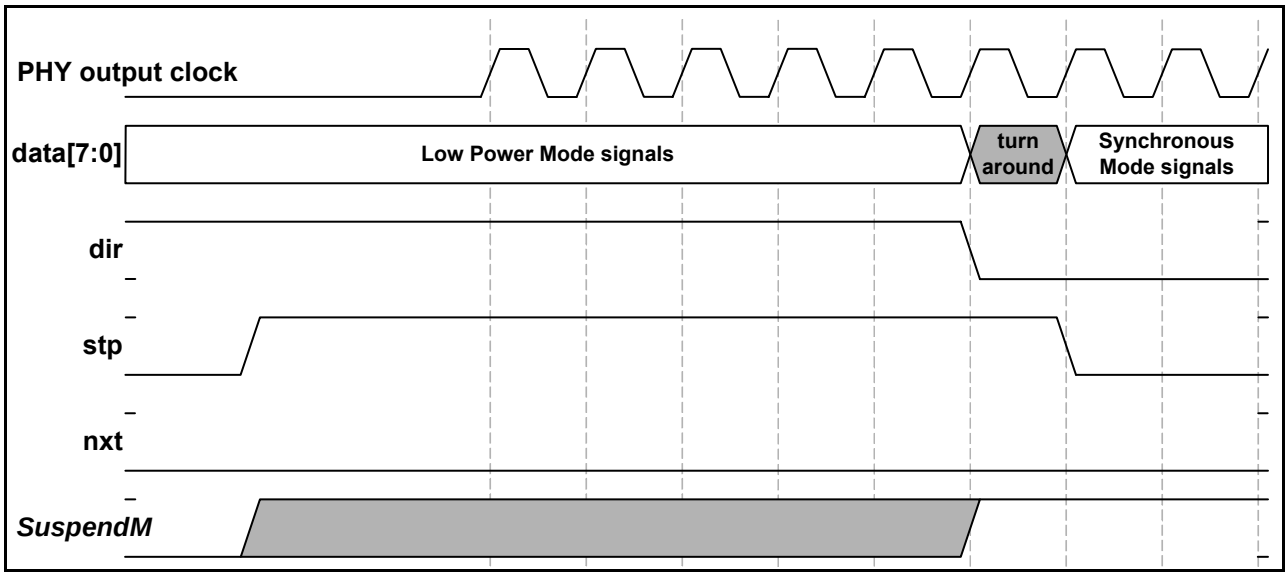
Resolution:
This is clarified in the SuspendM definition in Table 19, which states “The PHY must automatically set this bit to ‘1’ when Low Power Mode is exited.”

To make this even clearer to the reader, a clarification should be added to section 3.9.3.

Documentation Changes:
The following changes will be added to the first paragraph of section 3.9.3.

“As shown in Figure 46 and Figure 47, the Link signals the PHY to exit Low Power Mode by asynchronously asserting **stp**. The PHY immediately starts to wake up its internal circuitry. When the PHY clock meets ULPI timing requirements, the PHY de-asserts **dir**. The PHY must ensure a minimum of 5 cycles of **clock** have been driven prior to de-asserting **dir**. The PHY must also ensure that the ***SuspendM*** register is automatically set to 1b prior to de-asserting **dir**. The Link de-asserts **stp** in the cycle following the de-assertion of **dir**. There is one cycle of data bus turnaround provided after the de-assertion of **dir**, during which the value on **data** is not valid. The PHY stops driving the signals of Table 12 immediately before the turnaround cycle.”

Figure 46 will also be changed to reflect that the ***SuspendM*** register bit must be set to 1b prior to de-asserting **dir**.



Errata 07 – Validity of hostdisconnect in Low Power Mode

Type:
Clarification.

Issue:
It is not clear if hostdisconnect is valid when the PHY is acting as a host and Low Power Mode is enabled.

Resolution:
This is clarified in the UTMI+ specification, but should be indicated in ULPI for clarity. The hostdisconnect status signal must be reset to 0b when the PHY enters Low Power Mode. Since the disconnect detection circuitry is powered down in Low Power Mode, no detection is possible. The link must look at LineState on the data bus for connect and disconnect events.

Documentation Changes:
The hostdisconnect description in Table 24 already states that “Applicable only in host mode”. This text must be expanded to state that:

“Applicable only in host mode. Automatically reset to 0b when Low Power Mode is entered.”

Errata 08 – Enabling detection of the ID pin

Type:
Clarification.

Issue:
It is not clear when the IdGnd status is valid.

Resolution:
This is clarified in the UTMI+ specification, but should also be clarified in ULPI. The IdGnd status is valid 50ms after IdPullup is set to 1b, and the OTG state machines should sample IdGnd only after the 50ms time.

Documentation Changes:
The IdGnd definition in Tables 22, 23, 24, and 25 require the following extra text:

"**IdGnd** is valid 50ms after ***IdPullup*** is set to 1b, otherwise **IdGnd** is undefined and should be ignored."

Errata 09 – Clarifications on HS SOF packets

Type:
Clarification.

Issue:
HS SOF packets are not mentioned in the specification. This can cause confusion and could lead to implementations with different behaviors.

Resolution:
There are two important items to mention. First, the PHY must automatically append a 40-bit EOP when executing a TX CMD with the PID field set to A5. Second, HS SOF packets have a larger TX End Delay.

Documentation Changes:
The following text will be added to section 3.8.2.2:

"For all PID packets, the PHY must automatically prepend a SYNC pattern and append an EOP pattern. In High Speed, when the PID field of the TX CMD is 5h, the PHY must recognize that this is a Start-Of-Frame (SOF) packet and automatically append a long EOP."

A new row will be added to Table 9 to expand the TX End Delay parameters, as shown below.

TX End Delay	2-5	N/A	N/A	Number of clocks between the PHY detecting **stp** on the ULPI bus to completing transmission of EOP on the USB bus. Used for HS packets only. The Link can use TX End Delay to calculate when the packet has completed transmitting on the USB. HS EOP is completed when all 8 consecutive 1's have finished transmitting on the USB bus. FS and LS packets finish many clock cycles after **stp** is asserted. The Link must look for **RX CMD** bytes indicating SE0-to-J transition to determine when the transmission has completed on the USB bus.
	6-9	N/A	N/A	HS SOF packets have a long EOP. The link must wait at least 9 clocks or for an RX CMD indicating squelch (LineState = 00b) before transmitting the next packet.

Errata 10 – Power Control of Interrupt Sources

Type:
Clarification.

Issue:
It is not known which interrupts should be powered down in each mode.

Resolution:
To ensure multi-vendor interoperability, the link must control the power state of the interrupts. Whenever an interrupt rising/falling enable bit is set, the associated circuit must be powered, regardless of the mode. There are two exceptions:

1. IdGnd is gated with IdPullup.
2. Hostdisconnect is never valid during Low Power Mode.

Documentation Changes:
The following text will be added to the first paragraph of Section 3.6, Interrupt Event Notification.

"If an interrupt is enabled, the PHY must power the needed circuitry regardless of which mode the PHY is in. The only exceptions are the ***HostDisconnect*** interrupt which is valid only in Synchronous Mode, and the ***IdGnd*** interrupt which is controlled by ***IdPullup***."

The following change will be made to the first paragraph of Section 3.9, Low Power Mode.

"The Link can optionally place the PHY into Low Power Mode when the USB bus is suspended. The PHY can power down all circuitry except the interface pins and full speed receiver. The bus resistors must also be powered if VBUS is present. Any function must be powered if its corresponding register bit is set, including interrupt sources and the charge pump. If the PLL is powered down, the clock must be stopped without glitches."

The following change will be made to the first paragraph of Section 3.10, Full Speed / Low Speed Serial Mode.

"Full Speed / Low Speed Serial Mode (***FsLsSerialMode***) gives the Link direct access to the FS/LS serial analog transmitter and receiver. Two types of serial mode are defined in ULPI: ***3-pin FsLsSerialMode***, and ***6-pin FsLsSerialMode***. Both modes are optional. The PHY can power down all circuitry except the interface pins, full speed transmitter and receiver. Any function must be powered if its corresponding register bit is set, including interrupt sources and the charge pump."

The following line will be added to the text describing interrupt registers in sections 4.2.5, 4.2.6, 4.2.7, and 4.2.8.

"Interrupt circuitry can be powered down in any mode when both rising and falling edge enables are disabled."

Errata 11 – Enabling Interrupts in Low Power, Serial, and Carkit Modes

Type:
Clarification.

Issue:
It is not known if both rising and falling edges should be enabled when the clock is powered down.

Resolution:
A clarification will be added that the link/software should set both the rising and falling interrupt enables whenever entering a mode where clock is off, such as Low Power Mode and Serial Mode.

Documentation Changes:
The following line will be added to the text describing interrupt registers in sections 3.6, 4.2.5, 4.2.6, 4.2.7, and 4.2.8.

"To ensure interrupts are detectable when **clock** is powered down, the link should enable both rising and falling edges."

Errata 12 – Carkit data during audio

Type:
Addition.

Issue:
The carkit specification was not complete at the time of releasing ULPI 1.0. New additions are required to the ULPI register set to comply with the latest carkit specification. Specifically, carkit has added a data during audio feature.

Resolution:
Four new registers at a total of 6 addresses are required in the reserved area of the ULPI register set to provide the data during audio feature.

Documentation Changes:

The following text will be added to Section 3.11, Carkit.
In cases of conflict between this specification and the Carkit specification, the Carkit specification shall take precedence.

Four new registers are defined in Table 16, replacing 6 of the reserved addresses.

Field name	Size (bits)	Address (6 bits)			
		Rd	Wr	Set	Clr
Immediate Register Set					
Carkit Pulse Control (Optional)	8	22-24h	22h	23h	24h
Transmit Positive Width (Optional)	8	25h	25h	-	-
Transmit Negative Width (Optional)	8	26h	26h	-	-
Receive Polarity Recovery (Optional)	8	27h	27h	-	-

Register details will be inserted after section 4.2.15 as follows:

4.2.16 Carkit Pulse Control
Address: 22h-24h (Read), 22h (Write), 23h (Set), 24h (Clear).

This register is optional. It controls the operation of the carkit data-during-audio function within the PHY. The ***TxPlsEn*** and ***RxPlsEn*** bits are ignored if the ***CarkitMode*** bit in the **Interface Control** register is not set.

Refer to [Ref 7] for more information on carkit.

Field name	Bits	Access	Reset	Description
TxPlsEn	0	rd/wr/s/c	0b	Enables data-during-audio pulse transmit
RxPlsEn	1	rd/wr/s/c	0b	Enables data-during-audio pulse receive
SpkrLeftBiasEn	2	rd/wr/s/c	0b	Enables bias for left speaker.
SpkrRightBiasEn	3	rd/wr/s/c	0b	Enables bias for right speaker.
Reserved	7:4	-	0000b	Reserved.

Table 2 – Carkit Pulse Control

TxPlsEn
When the ***TxPlsEn*** bit is set, and the ***SpkLeftEn*** bit in the **Carkit Control** register is set, then the PHY shall output a positive pulse followed by a negative pulse on the D- line after each rising or falling edge on the data(0) line. When generating such a pulse pair, the PHY shall perform the steps, as defined in the Carkit specification [Ref 7]. The following list of steps provides information on the intent of the Carkit specification.

- tri-state the speaker buffer that drives the D- line
- drive the D- line to a voltage of 3.3V +/- 10%
- wait for the time specified in the **Transmit Positive Width** register
- drive the D- line to ground
- wait for the time specified in the **Transmit Negative Width** register
- stop driving the D- line to ground
- enable the speaker buffer that drives the D- line

RxPlsEn
When the ***RxPlsEn*** bit is set, and the ***MicEn*** bit in the **Carkit Control** register is set, then the PHY shall toggle the data(1) output each time a falling edge is detected on the D+ line that crosses the carkit interrupt threshold of $V_{PH\_DP\_LO}$. When the ***RxPlsEn*** bit is set, the **Receive Polarity Recovery** timer shall be enabled.

4.2.17 Transmit Positive Width
Address: 25h (Read), 25h (Write)

This register is optional. It specifies the width of the positive pulse that is output on the D- line when the ***TxPlsEn*** bit is set. The time is measured in units of 60MHz clock periods. The minimum ***TxPosWdth*** that must be supported is 8. The maximum ***TxPosWdth*** that must be supported is 64.

Refer to [Ref 7] for more information on carkit.

Field name	Bits	Access	Reset	Description
TxPosWdth	7:0	rd/wr	10h	Transmit positive pulse width

Table 3 – Transmit Positive Width

4.2.18 Transmit Negative Width
Address: 26h (Read), 26h (Write)

This register is optional. It specifies the width of the negative pulse that is output on the D- line when the ***TxPlsEn*** bit is set. The time is measured in units of 60MHz clock periods. The minimum ***TxNegWdth*** that must be supported is 8. The maximum ***TxNegWdth*** that must be supported is 64.

Refer to [Ref 7] for more information on carkit.

Field name	Bits	Access	Reset	Description
TxNegWdth	7:0	rd/wr	20h	Transmit negative pulse width

Table 4 – Transmit Negative Width

4.2.19 Receive Polarity Recovery
Address: 27h (Read), 27h (Write)

This register is optional.

When the data-during-audio feature is enabled in the Carkit [Ref 7], then the carkit sends UART data to the phone by converting the non-return-to-zero (NRZ) UART signal into a series of pulses, and transmitting

these pulses to the phone on the D+ line. The PHY in the phone then converts these pulses back into an NRZ UART signal by toggling to the **data**(1) line each time a pulse is received. If the PHY were to erroneously miss a pulse, or detect an extra pulse, then the polarity on the **data**(1) line would be incorrect. To recover from this condition, the PHY automatically resets the polarity of the **data**(1) line to logic high whenever the polarity of the data(1) line has been logic low for the time specified in the **Receive Polarity Recovery** register.

The **Receive Polarity Recovery** is only active if the ***RxPlsEn*** bit in the **Carkit Pulse Control** register is set. The time is measured in units of 0.25ms. The minimum ***RxPolRcvry*** that must be supported is 1. The maximum ***RxPolRcvry*** that must be supported is 255.

Refer to [Ref 7] for more information on carkit.

Field name	Bits	Access	Reset	Description
RxPolRcvry	7:0	rd/wr	02h	Receive polarity recovery time

Table 5 – Receive Polarity Recovery

Errata 13 – Protecting the PHY when the link tri-states stp and data

Type:
Fix and Clarification.

Issue:
It is possible for the link controller to take a longer time to power up than the PHY. When the PHY de-asserts **dir** after power-up and the link is not ready, this could result in unknown input values on **stp** and **data** signals. Unknown values on **data** could be interpreted by the PHY as TXCMD's, resulting in unsolicited USB activity and non-default register settings after power up. Similarly, the link could receive a hardware reset at any time, also resulting in unknown values on **stp** and **data** signals. The ULPI protocol must be fixed to ensure robust link and PHY behavior on power up and during hardware reset.

Resolution:
In general, the onus is placed on the PHY to provide a mechanism that ensures the data bus is not misinterpreted during power up or any time the link receives a hardware reset. The chosen mechanism is not allowed to place any additional constraints on the link controller except for those outlined here. Example solutions are given below.

Documentation Changes:
Additional text is appended to section 3.5 "Power On and Reset":

Further power up and reset requirements are outlined in section 3.12.

A new section is added, 3.13 "Safeguarding PHY Input Signals":

3.13 Safeguarding PHY Input Signals

For reasons including but not limited to hardware reset or slow power up, the link may not be able to correctly drive the ULPI interface. In such cases, when the PHY has **dir** de-asserted, the link is unable to drive **data** to the idle 00h state. The unknown values on the PHY **data** input signals could initiate unsolicited USB activity, register writes, serial or carkit transmissions. For this reason, the PHY must protect its **data** inputs at all times.

To safeguard against false commands on its **data** inputs, the PHY must incorporate a weak pull-up resistor on **stp**. Any time **stp** is unexpectedly high, the PHY assumes the link is unable to drive the interface and must enter a holding state. While in the holding state the PHY must not interpret commands on **data**, and must not assert **dir** unless its internal clocks de-stabilize. The link is also allowed to drive **stp** high at any time, forcing the PHY to stop interpreting commands on **data**. When the PHY is in the holding state, it can optionally enable weak pull-down resistors on **data**, preventing them from floating.

All RXCMD changes that occur while the PHY is in the holding state must be replaced with a single RX CMD update that is sent when the PHY exits the holding state, when the ULPI bus is available. The RX CMD update must always convey the current RXCMD values, not a previous or old value.

If the link can always drive **stp** and **data** to known values, it can disable the protection feature by setting the ***Interface Protect Disable*** bit in the **Interface Control** register to 1b. This potentially reduces power consumption.

When the clock is running, the link should drive **stp** high for at least one clock cycle before ceasing to drive the ULPI interface, forcing the PHY into the holding state to protect its **data** inputs. The pull up in the PHY will hold **stp** high in subsequent cycles. Figure 1 illustrates this scenario. Safe PHY operation cannot be guaranteed for implementations where the link cannot drive **stp** high before ceasing to drive the ULPI interface.

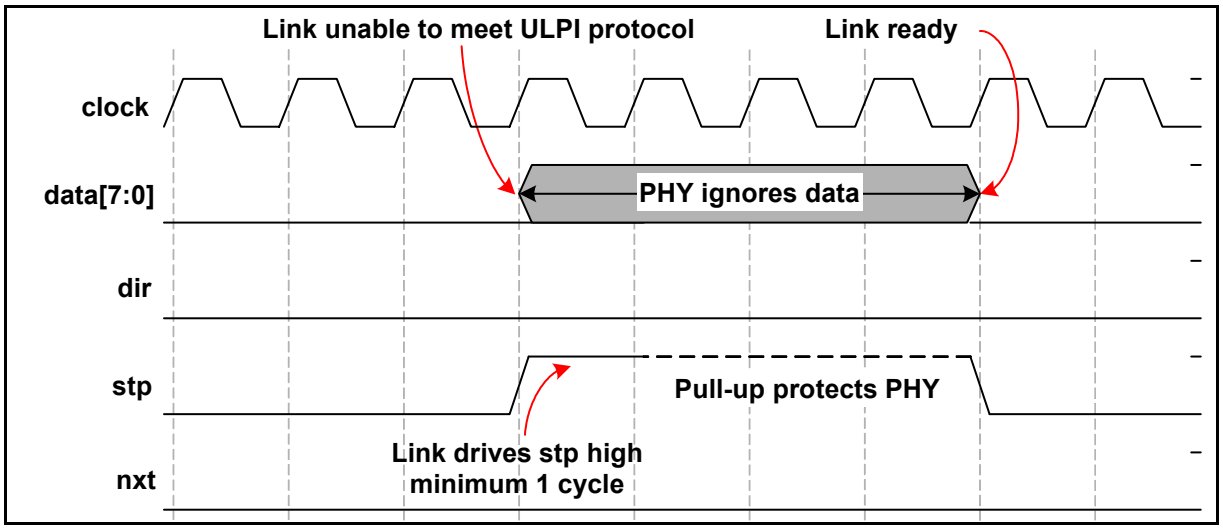


Figure 1 – PHY interface protected when the clock is running

During power up or when the clock is not running, the PHY always asserts **dir**, protecting its **data** inputs. As shown in Figure 2 and Figure 3, if **stp** is high when the PHY de-asserts **dir**, the PHY will immediately enter the holding state and protect its **data** inputs. When the link drives **stp** low, the PHY immediately starts processing its **data** inputs.

Further to Figure 3, if the PHY is in Low Power Mode when the link ceases driving the ULPI interface, the pull up on **stp** will automatically wake up the PHY. If the link does not want the PHY to automatically wake up, it must drive **stp** low.

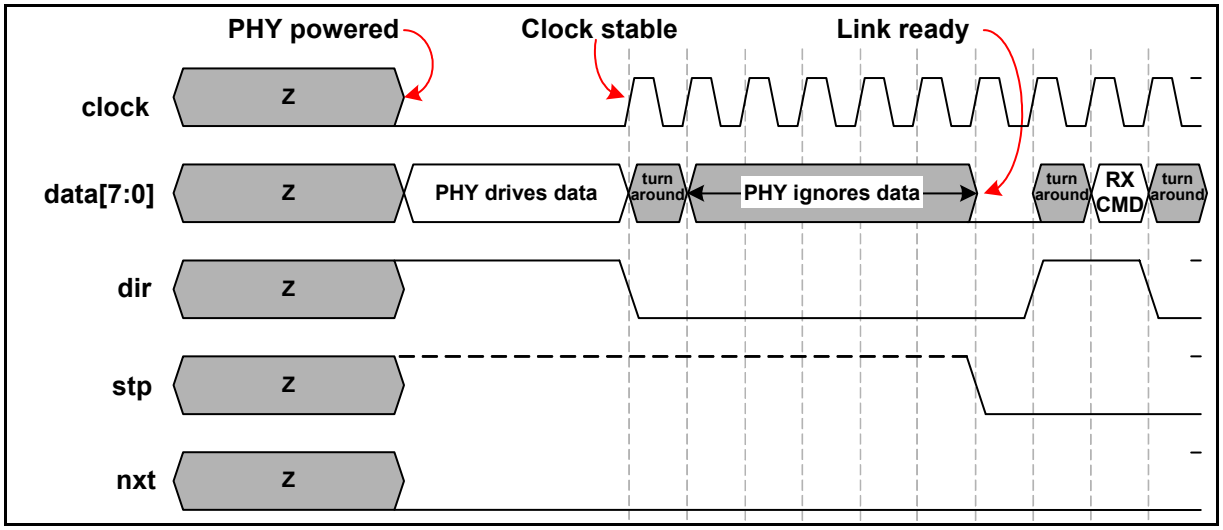


Figure 2 – Power up sequence when PHY powers up before Link. Interface is protected.

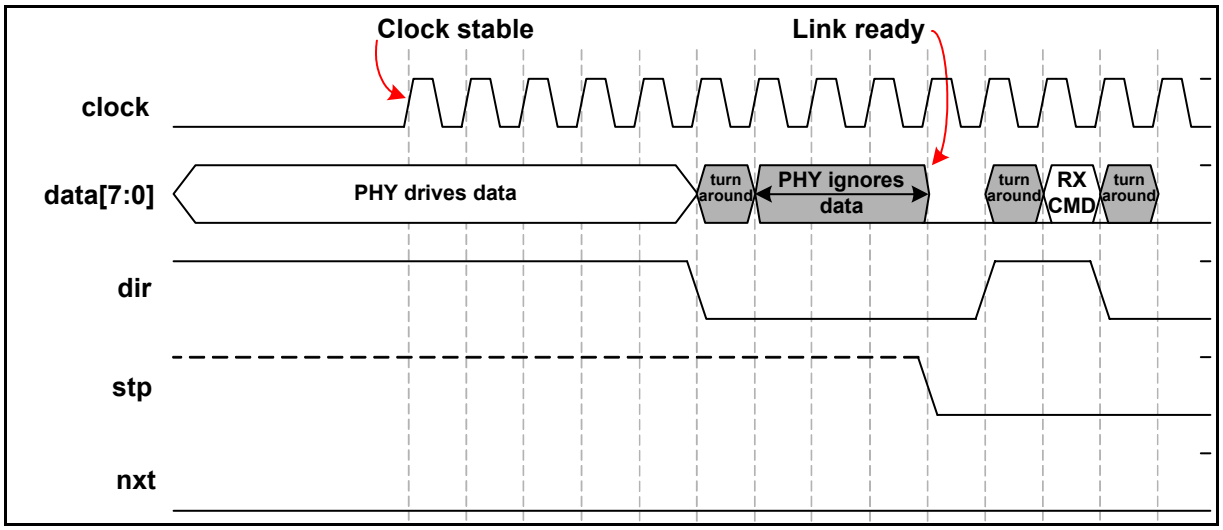


Figure 3 – PHY automatically exits Low Power Mode with interface protected.

If **dir** is high when the link resumes driving the ULPI interface, the link should assume the PHY is in Low Power Mode and drive **stp** high to wake up the PHY, as shown in Figure 4. The link de-asserts **stp** in the cycle after **dir** is de-asserted. The PHY begins processing its **data** inputs in the cycle when **dir** and **stp** are both low. This also applies during power up.

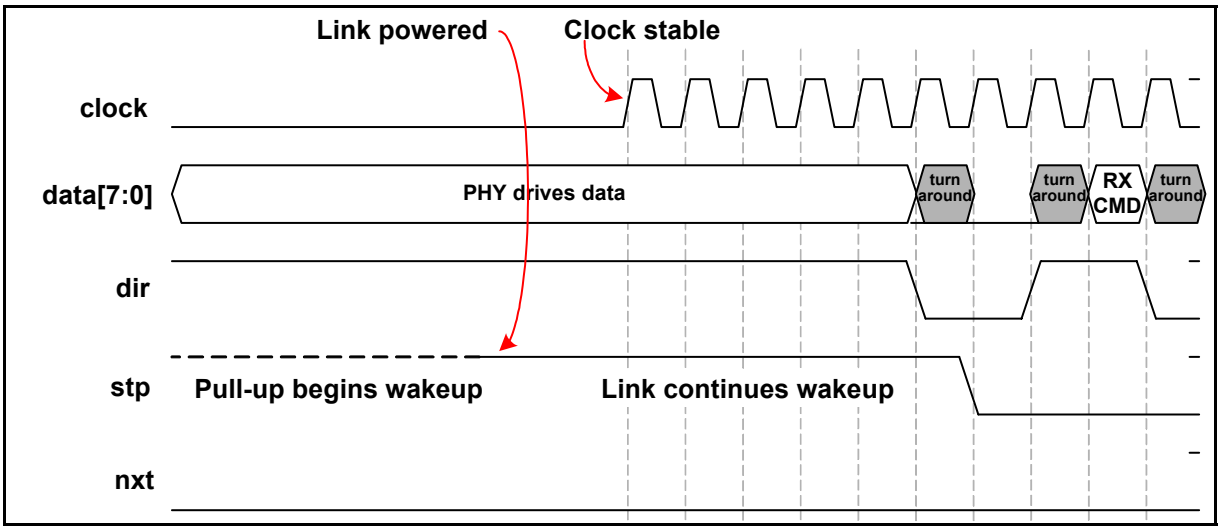


Figure 4 – Link resumes driving ULPI bus and asserts stp because clock is not running

As shown in Figure 5, the link is allowed to drive **stp** low during power up, when **dir** is high. The PHY begins processing its **data** inputs after the turnaround cycle when **dir** is de-asserted.

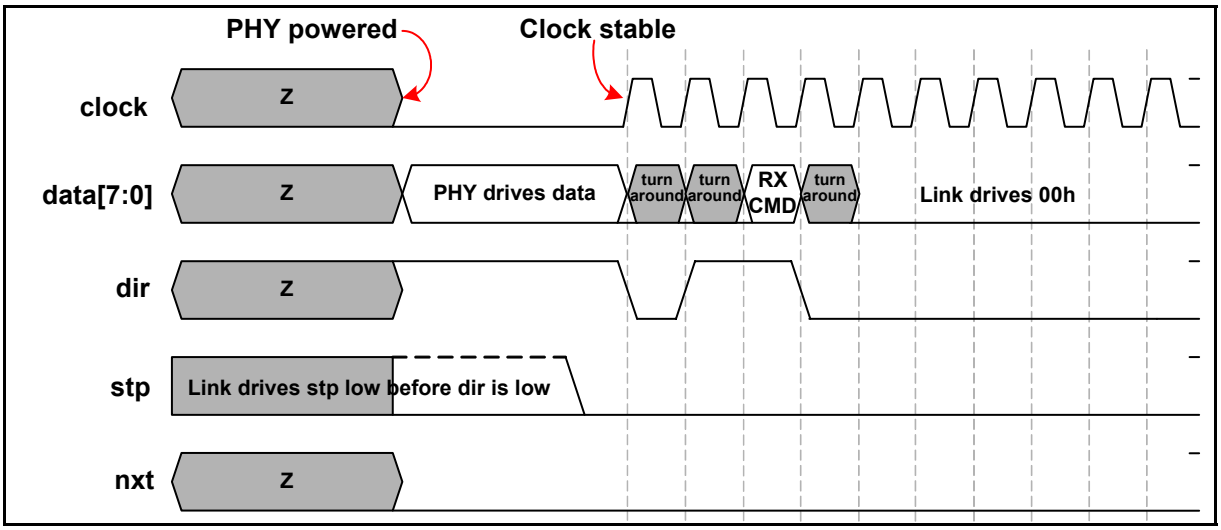


Figure 5 – Power up sequence when link powers up before PHY (ULPI 1.0 compliant links).

A new bit is defined in the Interface Control register:

Field name	Bits	Access	Reset	Description
Interface Protect Disable	7	rd/wr/s/c	0b	Controls circuitry built into the PHY for protecting the ULPI interface when the link tri-states **stp** and **data**. Any pull-ups or pull-downs employed by this feature can be disabled. This bit is not intended to affect the operation of the holding state. Refer to section 3.13 for more details. 0b : Enables the interface protect circuit (default). 1b : Disables the interface protect circuit.

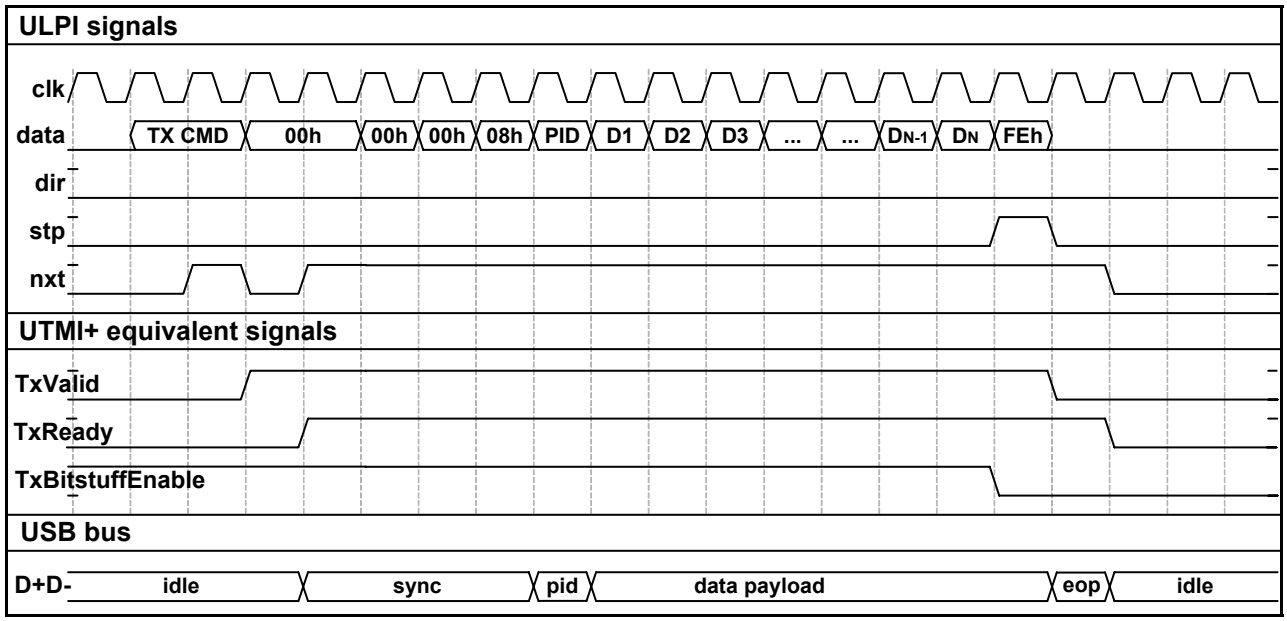
Errata 14 – Corrections for OpMode=11b (No SYNC/EOP generation)

Type:
Clarification.

Issue:
In Figure 44, it is not possible to assert TxValid during the TXCMD. TxValid should be asserted after the TXCMD, to coincide with the first data byte. TxReady should be delayed accordingly.

Resolution:
Figure 44 should be changed to show TxValid asserting after the TXCMD, with TxReady asserting two cycles later.

Documentation Changes:
TxValid and TxReady will be changed, with the figure below replacing Figure 44.



Errata 15 – Increased RXCMD delay when LineState indicates SE0

Type:
Clarification.

Issue:
As detailed in UTMI+ section 2.1.1.2, filtering on LineState can cause an RXCMD to be delayed by extra cycles when LineState indicates SE0.

Resolution:
The RXCMD Delay parameter of Table 9 must be supplemented with an extra row, showing new values when LineState indicates SE0.

Documentation Changes:
An extra row is added to Table 9, as shown below.

Parameter Name	HS PHY Delay	FS PHY Delay	LS PHY Delay	Definition
RX CMD Delay	2-4	2-4	2-4	Number of clocks after a change in the internal USB bus state is detected to an **RX CMD** byte being sent over the ULPI bus. Applies to all changes except SE0.
	2-4	4-6	16-18	Number of clocks between the USB bus indicating SE0 to an **RX CMD** byte being sent over the ULPI bus. Delay is increased due to filtering.

Errata 16 – Clock and SuspendM behavior during AutoResume

Type:
Clarification.

Issue:
In Figure 43, when the PHY exits Low Power Mode and is automatically transmitting resume signalling, the diagram incorrectly shows that the link must set SuspendM to 1b. This is not necessary because the PHY must automatically set ***SuspendM*** to 1b whenever it exits Low Power Mode. The diagram also fails to show that the PHY waits a minimum of 5 cycles before de-asserting **dir** when exiting Low Power Mode.
The text and Figure 43 also incorrectly reference Tprep as the clock wakeup time. The clock wakeup time is defined by Tstart_host.

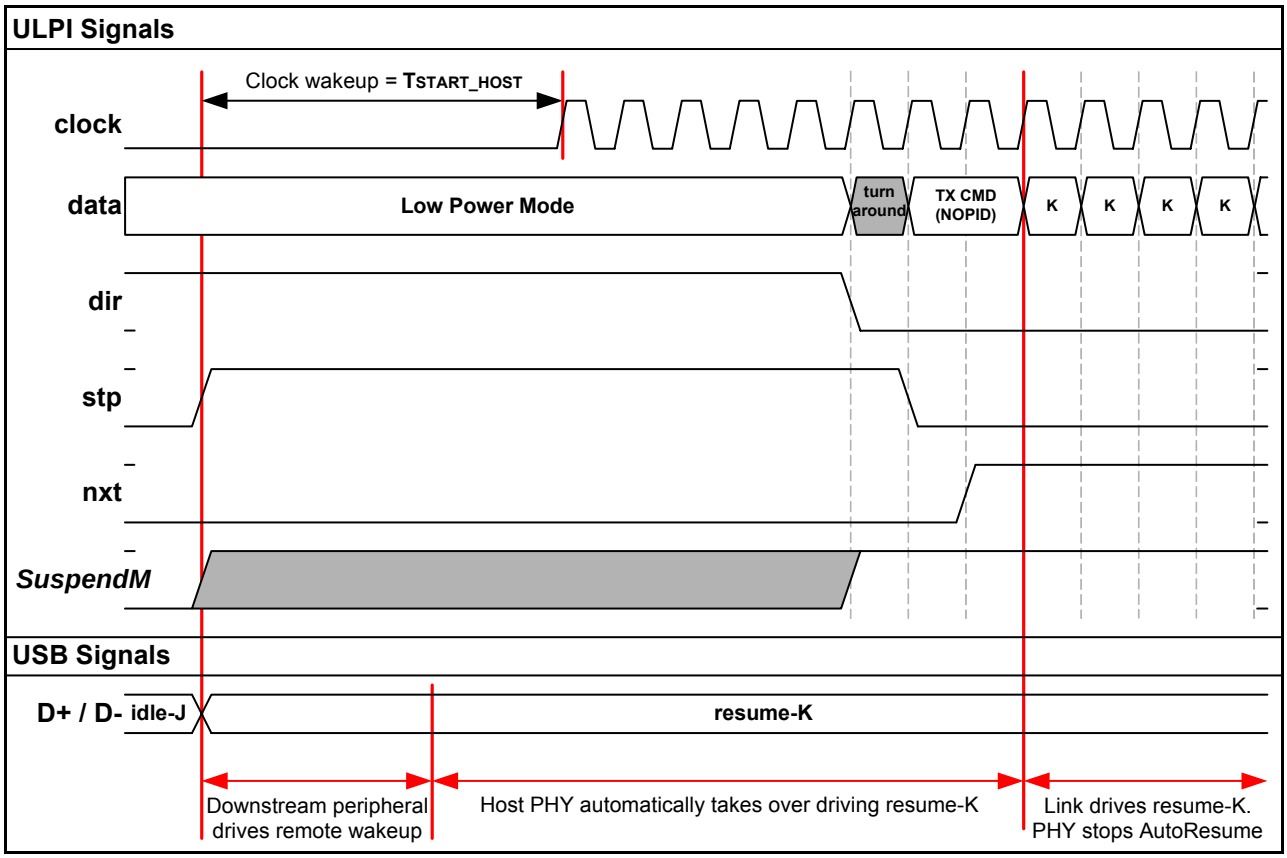
Resolution:
Figure 43 must be updated to show that it is not necessary for the link controller to set ***SuspendM*** to 1b, and that the PHY must set ***SuspendM*** to 1b automatically. If a link controller is currently designed to set SuspendM to 1b when exiting Low Power Mode, there should be no adverse affect.

Figure 43 must also be corrected to show that the PHY must wait at least 5 cycles before de-asserting **dir**.

References to Tprep must be changed to Tstart_host.

Documentation Changes:
The text in section 3.8.5.4.4 and Figure 43 are updated as shown below.

As shown in Figure 43, the PHY must internally drive the resume-K until the clock is restored and it receives a **TXCMD** of the NOPID type. When the clock is restored, the Link takes over driving the resume-K by sending a **TXCMD** of the NOPID type. The PHY must ensure there are no glitches during the resume sequence when transitioning between automatic resume and the resume-K driven by the link in the NOPID command. The PHY must also ensure the ***SuspendM*** register bit is automatically set to 1b before exiting Low Power Mode. Implementation details are left to the PHY vendor, however the PHY vendor must specify the clock wake-up time, Tstart_host, as defined in Table 4.


ULPI Signals
Clock wakeup = TSTART_HOST
clock
data
Low Power Mode
turn around
TX CMD (NOPID)
K
K
K
K
dir
stp
nxt
SuspendM
USB Signals
D+ / D-
idle-J
resume-K
Downstream peripheral drives remote wakeup
Host PHY automatically takes over driving resume-K
Link drives resume-K. PHY stops AutoResume

Errata 17 – OpMode typo in 3.8.5.1

Type:
Clarification.

Issue:
In section 3.8.5.1, item 3 “Peripheral Responds, the text indicates that Chirp Mode setting is 01b. This is wrong, it should be 10b.

Resolution:
Change the text in section 3.8.5.1.

Documentation Changes:
The text in section 3.8.5.1 is updated as shown below.

3. **Peripheral Responds** – After detecting SE0 for no less than 2.5us, if the peripheral is Hi-Speed capable, the peripheral Link sets ***XcvrSelect*** to 00b (HS) and ***OpMode*** to 10b (chirp), and follows this immediately with a **TX CMD** (NOPID), transmitting a chirp K for no less than 1ms. and the chirp K must end it no more than 7ms after reset time T0. If the peripheral is in Low Power Mode, it must wake up its clock within 5.6ms, leaving 200us for the Link to start transmitting the chirp K, and 1.2ms for the chirp K to complete (worst case with 10% slow clock).

Errata 18 – Host PHY must first detect peripheral resume-K before transmitting automatic Resume-K

Type:
Clarification.

Issue:
The PHY must first detect a resume-K from the downstream peripheral before automatically transmitting the host resume-K. The text on AutoResume does not clearly specify this detection, possibly resulting in PHY implementations that always transmit resume signaling when the PHY exits Low Power Mode. This in turn would result in an unintentional resuming of the entire USB tree attached to that particular host port.

Resolution:
The text in section 3.8.5.4.4 must be changed to clarify that the PHY is responsible for detecting the peripheral resume-K before automatically transmitting the host resume-K.

Documentation Changes:
Text in the second paragraph of section 3.8.5.4.4 will be updated as follows:

If the PHY is in host mode, the clock is powered down, and the remote wake-up signaling is detected by the PHY, the Link must wake up the clock and take over driving of the resume-K signaling.

Errata 19 – Section 3.9.3 “Exiting Low Power Mode” makes incorrect reference to 3.8.5.4.4 “Autoresume”

Type:
Clarification.

Issue:
The reference to 3.8.5.4.4 in section 3.9.3 is incorrect. The reference was intended to point to Tprep timing during AutoResume. However, all references to Tprep in 3.8.5.4.4 have been replaced by Tstart_host, as outlined in Errata 16.

Resolution:
The reference to 3.8.5.4.4 in section 3.9.3 should be removed.

Documentation Changes:
Text in the second paragraph of section 3.9.3 will be updated as follows:

“As shown in Figure 47, when the Link provides an input clock, the PHY must synchronize its internal clock within Tprep. Note that Tprep is implementation dependent. Refer to Table 4 ~~and section 3.8.5.4.4~~ for more information on Tprep.”