UTMI+ Low Pin Interface (ULPI) Specification

Revision 1.1~~1.0~~

October 20~~February 2~~, 2004

Revision History

Revision	Issue Date	Comment
0.9	November 12, 2003	Pre-release.
1.0rc1	January 3, 2004	Introduce PHY interface "modes". Update interface timings. Clarify 4-bit data clocking. Clarify sending of RX CMD's and interrupts. Introduce AutoResume feature. Route int pin to data(3) during 6-pin Serial Mode. Explain VBUS thresholds. Add T&MT diagram and updated text. Add new section to explain how PHY is aborted by Link. Various clarifications.
1.0rc2	January 13, 2004	Add block diagram. Tighten interface timing. Modify suspend protocol to more closely resemble UTMI. Add SPKR_L and SPKR_MIC to signal list and T&MT connector. Various clarifications.
1.0rc3	January 19, 2004	Specify that PHY must send RX CMD after Reset. Link + PHY clock startup time of no more than 5.6ms for a peripheral is now mandatory. PHY output delay reduced from 10ns to 9ns. Added link decision time numbers for low speed. Various Clarifications.
1.0	February 2, 2004	1.0rc3 adopted as 1.0 release.
1.1rc1	September 1, 2004	Various clarifications and fixes to hold time numbers, sending RXCMDs, FsLsSerialMode, Vbus control and monitoring, Test_J and Tesk_K signalling, Low Power Mode, Hostdisconnect, ID detection, HS SOF packets, interrupts, Carkit Mode, interface protection, No SYNC/EOP mode, linestate filtering, and AutoResume.
1.1rc2	October 4, 2004	Re-arranged text in section 3.8.7.3. Updated contributors list.
1.1	October 20, 2004	1.1rc2 adopted as 1.1 release.

Promoters

ARC International Inc.
Conexant Systems, Inc.
Mentor Graphics Corporation
Philips
SMSC
TransDimension, Inc.

Contributors

Bart Vertenten	Philips
Batuhan Okur	Philips
Bill Anderson	Motorola
Bill McInerney	TransDimension
Brian Booker	Cypress
Chris Belanger	ARC
Chris Kolb	ARC
Chris Schell	Philips
Chung Wing Yan	Philips
Dave Sroka	Philips
David Wang	Philips
David Wooten	TransDimension
Eric Kawamoto	SMSC
Farran Mackay	Philips
Frank Frazier	Conexant
Fred Roberts	Synopsys
Hassan Farooq	Conexant
Hyun Lee	TransDimension
Ian Parr	Mentor
Jay Standiford	TransDimension
Jerome Tjia	Philips
Mark Saunders	Mentor
Mohamed Benromdhane	Conexant
Morgan Monks	SMSC
Nabil Takla	ISI
Peter Tengstrand	ARC
Ramanand Mandayam	Conexant
Rob Douglas	Mentor
Saleem Mohamed	Synopsys
Shaun Reemeyer	Philips (Author)
Simon Nguyen	Cypress
Subramanyam Sankaran	Philips
Sue Vining	Texas Instruments
Terry Remple	Qualcomm
Timothy Chen	Conexant
Vincent Chang	Conexant

Questions should be emailed to lpcwg@boardrooms.org.

Table of Contents

Figures

Tables

1. Introduction

1.1 General

This specification defines a generic PHY interface in Chapter 2.

In Chapter 3, the generic interface is applied to the UTMI+ protocol, reducing the pin count for discrete USB transceiver implementations supporting On-The-Go, host, and peripheral application spaces.

1.2 Naming Convention

Emphasis is placed on normal descriptive text using underlined Arial font, e.g. must.
Signal names are represented using the lowercase bold Arial font, e.g. **clk**.
Registers are represented using initial caps, bold Arial font, e.g. **OTG Control**.
Register bits are represented using initial caps, bold italic Arial font, e.g. ***USB Interrupt Enable Falling***.

1.3 Acronyms and Terms

A-device	Device with a Standard-A or Mini-A plug inserted into its receptacle
B-device	Device with a Standard-B or Mini-B plug inserted into its receptacle
DRD	Dual-Role Device
FPGA	Field Programmable Gate Array
FS	Full Speed
HNP	Host Negotiation Protocol
HS	Hi-Speed
Link	ASIC, SIE, or FPGA that connects to an ULPI transceiver
LPI	Low Pin Interface
LS	Low Speed
OTG	On-The-Go
PHY	Physical Layer (Transceiver)
PLL	Phase Locked Loop
SE0	Single Ended Zero
SIE	Serial Interface Engine
SRP	Session Request Protocol
T&MT	Transceiver and Macrocell Tester
ULPI	UTMI+ Low Pin Interface
USB	Universal Serial Bus
USB-IF	USB Implementers Forum
UTMI	USB 2.0 Transceiver Macrocell Interace
UUT	Unit Under Test

1.4 References

[Ref 1] Universal Serial Bus Specification, Revision 2.0
[Ref 2] On-The-Go Supplement to the USB 2.0 Specification, Revision 1.0a
[Ref 3] USB 2.0 Transceiver Macrocell Interface (UTMI) Specification, v1.05
[Ref 4] UTMI+ Specification, Revision 1.0
[Ref 5] CEA-2011, OTG Transceiver Specification
[Ref 6] CEA-936A, Mini-USB Analog Carkit Interface Specification
[Ref 7] USB 2.0 Transceiver and Macrocell Tester (T&MT) Interface Specification, Version 1.2

2. Generic Low Pin Interface

2.1 General

This section describes a generic low pin interface (LPI) between a Link and a PHY. Interface signals are defined and the basic communication protocol is described. The generic interface can be used as a common starting point for defining multiple application-specific interfaces.

Chapter 3 defines the UTMI+ Low Pin Interface (ULPI), which is based on the generic interface described here. For ULPI implementations, the definitions in chapter 3 over-ride anything defined in chapter 2.

2.2 Signals

The LPI transceiver interface signals are described in Table 1~~Table 1~~. The interface described here is generic, and can be used to transport many different data types. Depending on the application, the data stream can be used to transmit and receive packets, access a register set, generate interrupts, and even redefine the interface itself. All interface signals are synchronous when **clock** is toggling, and asynchronous when **clock** is not toggling. Data stream definition is application-specific and should be explicitly defined for each application space for inter-operability.

Control signals **dir**, **stp**, and **nxt** are specified with the assumption that the PHY is the master of the data bus. If required, an implementation can define the Link as the master. If the Link is the master of the interface, the control signal direction and protocol must be reversed.

Signal	Direction	Description
PHY Interface		
clock	I/O	Interface clock. Both directions are allowed. All interface signals are synchronous to **clock**.
data	I/O	Bi-directional data bus, driven low by the Link during idle. Bus ownership is determined by **dir**. The Link and PHY initiate data transfers by driving a non-zero pattern onto the data bus. LPI defines interface timing for single-edge data transfers with respect to rising edge of **clock**. An implementation may optionally define double-edge data transfers with respect to both rising and falling edges of **clock**.
dir	OUT	Direction. Controls the direction of the **data** bus. When the PHY has data to transfer to the Link, it drives **dir** high to take ownership of the bus. When the PHY has no data to transfer it drives **dir** low and monitors the bus for Link activity. The PHY pulls **dir** high whenever the interface cannot accept data from the Link. For example, when the internal PHY PLL is not stable.
stp	IN	Stop. The Link asserts this signal for 1 clock cycle to stop the data stream currently on the bus. If the Link is sending data to the PHY, **stp** indicates the last byte of data was on the bus in the previous cycle. If the PHY is sending data to the Link, **stp** forces the PHY to end its transfer, de-assert **dir** and relinquish control of the the data bus to the Link.
nxt	OUT	Next. The PHY asserts this signal to throttle the data. When the Link is sending data to the PHY, **nxt** indicates when the current byte has been accepted by the PHY. The Link places the next byte on the data bus in the following clock cycle. When the PHY is sending data to the Link, **nxt** indicates when a new byte is available for the Link to consume.

Table 1 – LPI generic interface signals

2.3 Protocol

2.3.1 Bus Ownership

The PHY is the master of the LPI bi-directional data bus. Ownership of the data bus is determined by the **dir** signal from the PHY, as shown in Figure 1~~Figure 1~~. When **dir** is low, the Link can drive data on the bus. When **dir** is high, the PHY can drive data on the bus. A change in **dir** causes a turnaround cycle on the bus during which, neither Link nor PHY can drive the bus. Data during the turnaround cycle is undefined and must be ignored by both Link and PHY.

The **dir** signal can be used to directly control the data output buffers of both PHY and Link.

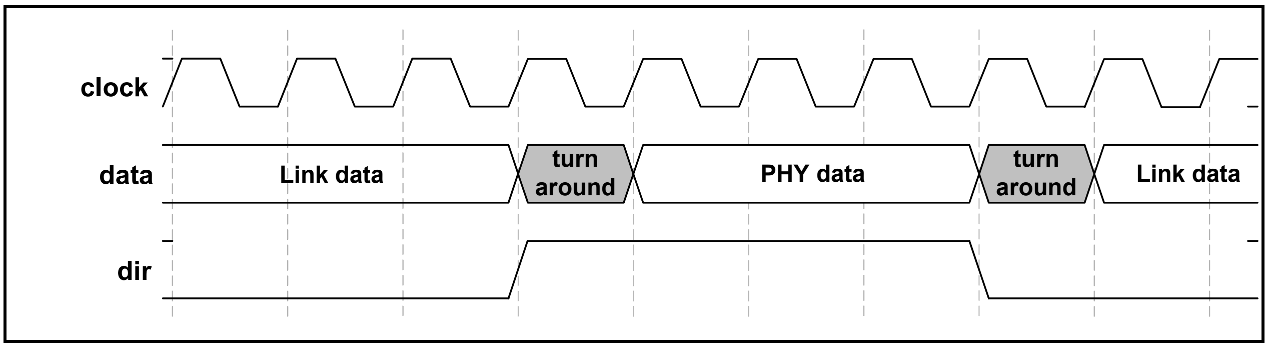

Figure 1 – LPI generic data bus ownership

2.3.2 Transferring Data

As shown in the first half of Figure 2~~Figure 2~~, the Link continuously drives the data bus to 00h during idle. The Link transmits data to the PHY by driving a non-zero value on the data bus. To signal the end of data transmission, the Link asserts **stp** in the cycle following the last data byte.

In the second half of Figure 2~~Figure 2~~, the Link receives data when the PHY asserts **dir**. The PHY asserts **dir** only when it has data to send to the Link, and keeps **dir** low at all other times. The PHY drives data to the Link after the turnaround cycle.

The **nxt** signal can be used by the PHY to throttle the data during transmit and receive. During transmit, **nxt** may be asserted in the same cycle that the Link asserts **stp**.

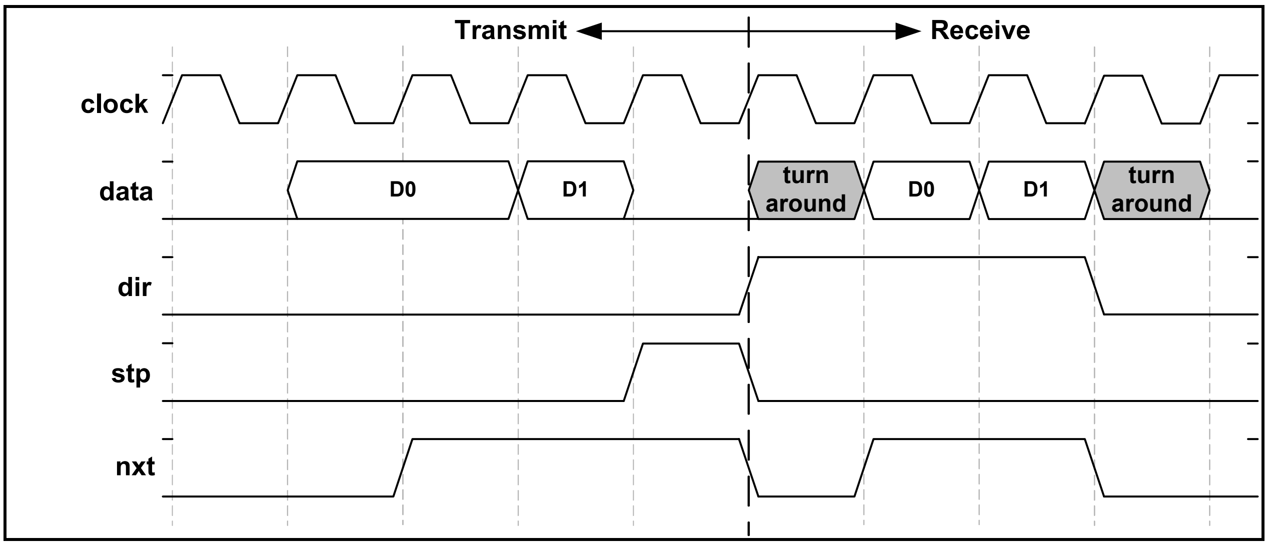

Figure 2 – LPI generic data transmit followed by data receive

2.3.3 Aborting Data

The PHY can assert **dir** to interrupt any data being transmitted by the Link. If the Link needs to interrupt data being received from the PHY, it asserts **stp** for one clock cycle, as shown in Figure 3~~Figure 3~~. This causes the PHY to unconditionally[1] de-assert **dir** and accept a complete data transmit from the Link. The PHY may re-assert **dir** again only when the data transmit from the Link has completed.

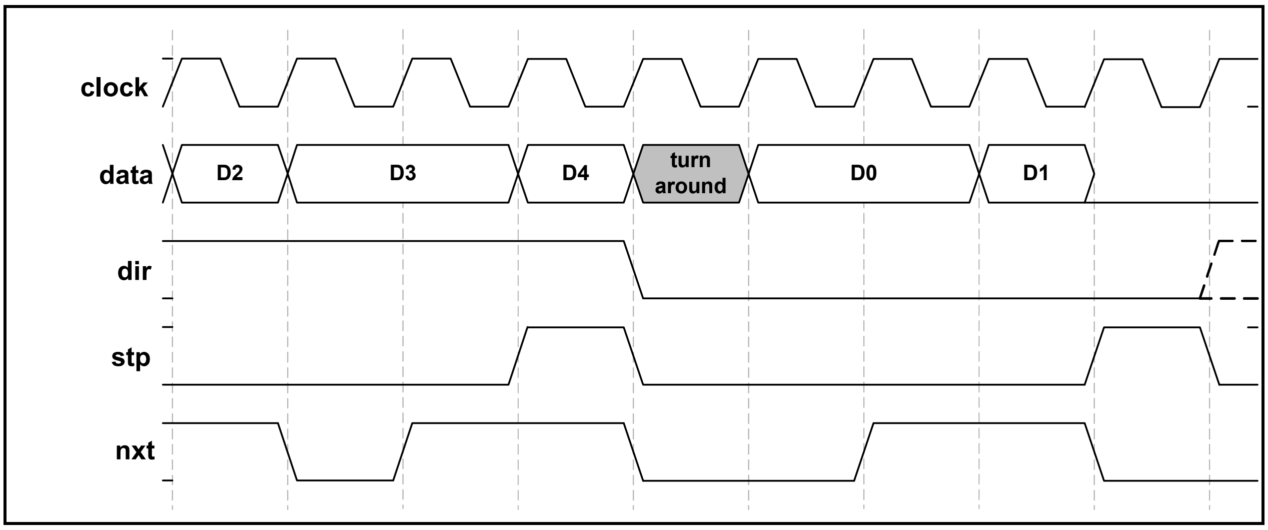

Figure 3 – Link asserts stp to halt receive data

[1] The PHY will not de-assert **dir** if the ULPI interface is not usable. For example, if the internal PLL is not stable.

3. UTMI+ Low Pin Interface

3.1 General

This section describes how any UTMI+ core can be wrapped to convert it to the smaller LPI interface. The generic interface described in chapter 2 is used as a starting point. This section always over-rides anything stated in chapter 2. While this specification details support of UTMI+ Level 3, PHY implementers may choose to support any of the Levels defined in UTMI+.

ULPI defines a PHY to Link interface of 8 or 12 signals that allows a lower pin count option for connecting to an external transceiver that may be based on the UTMI+ specification. The pin count reduction is achieved by having relatively static UTMI+ signals be accessed through registers and by providing a bi-directional data bus that carries USB data and provides a means of accessing register data on the ULPI transceiver.

This specification relies on concepts and terminology that are defined in the UTMI+ specification [Ref 4]. Specifically, if a ULPI PHY design is based on an internal UTMI+ core, then that core must implement the following UTMI+ features.

- **Linestate** must accurately reflect **D+**/**D-** to within 2-3 clocks. It is up to individual Link designers to use **Linestate** to time bus events.
- Filtering to prevent spurious SE0/SE1 states appearing on **Linestate** due to skew between **D+** and **D-**. Filtering of 14 clock cycles is required in Low Speed, and 2 clock cycles in Full Speed and Hi-Speed modes.
- The PHY must internally block the USB receive path during transmit. The receive path can be unblocked when the internal Squelch (HS) or SE0-to-J (FS/LS) is seen.
- **TxReady** must be used for all types of data transmitted, including Chirp.
- Due to noise on the USB, it is possible that **RxActive** asserts and then de-asserts without any valid data being received, and **RxValid** will not assert. The Link should operate normally with these data-less **RxActive** assertions.

As shown in Figure 4~~Figure 4~~, a PHY or Link based on this specification can be implemented as an almost transparent wrapper around existing UTMI+ IP cores, preserving the original UTMI+ packet timing, while reducing pin count and leaving all functionality intact. This should not be taken to imply that other implementations are not possible.

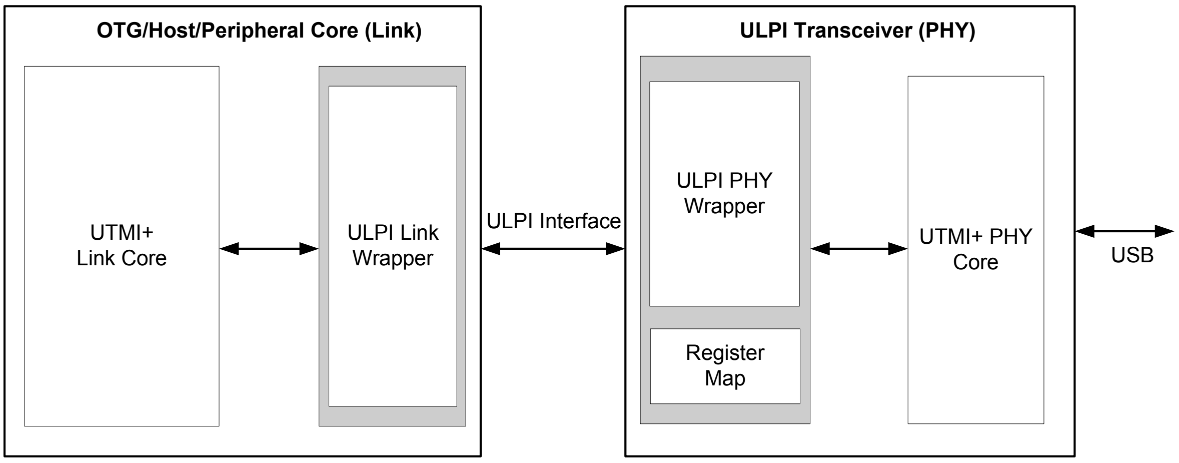

Figure 4 – Creating a ULPI system using wrappers

3.2 Signals

Table 2 describes the ULPI interface on the PHY. The PHY is always the master of the ULPI bus. USB and Miscellaneous signals may vary with each implementation and are given only as a guide to PHY designers.

Signal	Direction	Description
PHY Interface		
clock	I/O	Interface clock. The PHY must be capable of providing a 60MHz output clock. Support for an input 60MHz clock is optional. If the PHY supports both clock directions, it must not use the ULPI control and data signals for setting the clock direction.
data	I/O	Data bus. Driven to 00h by the Link when the ULPI bus is idle. Two bus widths are allowed: • 8-bit data timed on rising edge of **clock**. • (Optional) 4-bit data timed on rising and falling edges of **clock**.
dir	OUT	Controls the direction of the **data** bus[2]. The PHY pulls **dir** high whenever the interface cannot accept data from the Link. For example, when the internal PLL is not stable. This applies whether Link or PHY is the clock source.
stp	IN	The Link must assert **stp** to signal the end of a USB transmit packet or a register write operation, and optionally to stop any receive. The **stp** signal must be asserted in the cycle after the last data byte is presented on the bus.
nxt	OUT	The PHY asserts **nxt** to throttle all data types, except register read data and the **RX CMD**. Identical to **RxValid** during USB receive, and **TxReady** during USB transmit. The PHY also asserts **nxt** and **dir** simultaneously to indicate USB receive activity (**RxActive**), if **dir** was previously low. The PHY is not allowed to assert **nxt** during the first cycle of the **TX CMD** driven by the Link.
USB Interface		
D+	I/O	**D+** pin of the USB cable. Required.
D-	I/O	**D-** pin of the USB cable. Required.
ID	IN	**ID** pin of the USB cable. Required for OTG-capable PHY's.
VBUS	I/O	**VBUS** pin of the USB cable. Required for OTG-capable PHY's. Required for driving VBUS and the VBUS comparators.
Miscellaneous		
XI	IN	Crystal input pin. Vendors should specify supported crystal frequencies.
XO	OUT	Crystal output pin.
C+	I/O	Positive terminal of charge pump capacitor.
C-	I/O	Negative terminal of charge pump capacitor.
SPKR_L	IN	Optional Carkit left/mono speaker input signal.
SPKR_MIC	I/O	Optional Carkit right speaker input or microphone output signal.
RBIAS	I/O	Bias current resistor.

Table 2 – PHY interface signals

[2] UTMI+ wrapper developers should note that data bus control has been reversed from UTMI to ensure that USB data reception is not interrupted by the Link.

3.3 Block Diagram

An example block diagram of a ULPI PHY is shown in Figure 5~~Figure 5~~. This example is based on an internal UTMI+ Level 3 core [Ref 4], which can interface to peripheral, host, and On-The-Go Link cores. A description of each major block is given below.

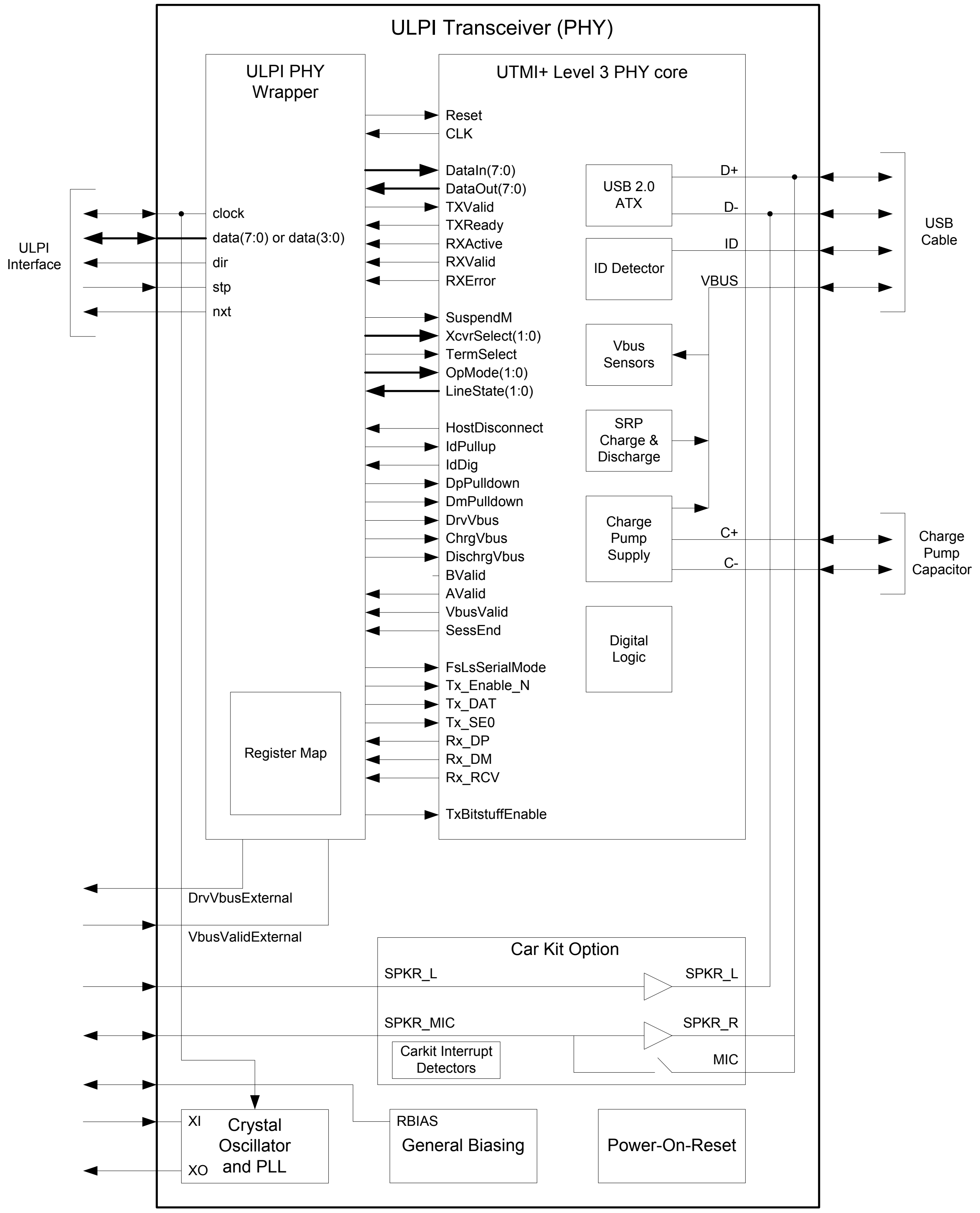

Figure 5 – Block diagram of ULPI PHY

UTMI+ Level 3 PHY core

The ULPI PHY may contain a core that is compliant to any UTMI+ level [Ref 4]. Signals for 16-bit data buses are not supported in ULPI. While Figure 5~~Figure 5~~ shows the typical blocks for a Level 3 UTMI+ core, the PHY vendor must specify the intended UTMI+ level, and provide the functionality necessary for compliance to that level.

ULPI PHY Wrapper

The ULPI PHY wrapper of Figure 5~~Figure 5~~ reduces the UTMI+ interface to the Low Pin Interface described in this document. All signals shown on the UTMI+ Level 3 PHY core are reduced to the ULPI interface signals **clock**, **data**, **dir**, **stp**, and **nxt**. The Register Map stores the relatively static signals of the UTMI+ interface.

Crystal Oscillator and PLL

When a crystal is attached to the PHY, the internal clock(s) and the external 60MHz interface clock are generated from the internal PLL. When no crystal is attached, the PHY may optionally generate the internal clock(s) from an input 60MHz clock provided by the Link.

General Biasing

Internal analog circuits require an accurate bias current. This is typically generated using an external, accurate reference resistor.

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~~ control of the~~ external supply is ~~vendor-specific but must be derived from the ~~controlled by the ***DrvVbus*** and the optional ***DrvVbusExternal*** bits in the **OTG Control** register~~, which is set by the Link~~. 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.

Power-On-Reset

A power-on-reset circuit must be provided in the PHY. When power is first applied to the PHY, the power-on-reset will reset all circuitry and leave the ULPI interface in a usable state.

Carkit Option

The PHY may optionally support Carkit Mode [Ref 6]. While in Carkit Mode, the PHY routes speaker and microphone signals between the Link and the USB cable. In carkit mono mode, **SPKR_L** inputs a mono speaker signal and **SPKR_MIC** outputs the microphone signal, **MIC**. In carkit stereo mode, **SPKR_L** inputs the left speaker signal, and **SPKR_MIC** inputs the right speaker signal, **SPKR_R**.

3.4 Modes

The ULPI interface can operate in one of five independent modes listed in Table 3~~Table 3~~. The interface is in Synchronous Mode by default. Other modes are enabled by bits in the **Function Control** and **Interface Control** registers. In Synchronous Mode, the data bus carries commands and data. In other modes, the data pins are redefined with different functionality. Synchronous Mode and Low Power Mode are mandatory.

Mode Name	Mode Description
Synchronous Mode	This is the normal mode of operation. The clock is running and is stable with the characteristics defined in section 3.6. The ULPI interface carries commands and data that are synchronous to **clock**.
Low Power Mode	The PHY is powered down with the clock stopped. The PHY keeps **dir** asserted, and the data bus is redefined to carry ***LineState*** and interrupts. See section 3.9 for more information.
6-pin FS/LS Serial Mode (optional)	The data bus is redefined to 6-pin serial mode, including 6 pins to transmit and receive serial USB data, and 1 pin to signal interrupt events. The clock can be enabled or disabled. This mode is valid only for implementations with an 8-bit data bus. See section 3.10 for more information.
3-pin FS/LS Serial Mode (optional)	The data bus is redefined to 3-pin serial mode, including 3 pins to transmit and receive serial USB data, and 1 pin to signal interrupt events. The clock can be enabled or disabled. See section 3.10 for more information.
Carkit Mode (optional)	The data bus is redefined to Carkit mode [Ref 6], including 2 pins for serial UART data, and 1 pin to signal interrupt events. The clock may optionally be stopped. See section 3.11 for more information.

Table 3 – Mode summary

3.5 Power On and Reset

The ULPI PHY provides an internal power-on reset circuit that resets all logic, including the UTMI+ core, ULPI interface and registers, on power-up.

After power-up, and when the clock starts toggling, the Link must reset the PHY by writing to the ***Reset*** bit in the **Function Control** register. When this bit is set, the transceiver will assert **dir** and reset the digital core. When the reset completes, the PHY de-asserts **dir** and automatically clears the ***Reset*** bit in the **Function Control** register. After de-asserting **dir**, the PHY must immediately re-assert **dir** and send an **RX CMD** update to the Link.

During reset, the data bus is driven by the PHY, however the data is undefined and must not be interpreted by the Link. The Link must not attempt to access the PHY until the reset has completed and **dir** is de-asserted. The ULPI interface and registers are not affected by ***Reset***, unless otherwise stated in the register definitions of chapter 4.

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

3.6 Interrupt Event Notification

In any mode, the PHY is able to detect interrupt events, and notify the link that an interrupt event has occurred. Interrupt event notifications are enabled by the **USB Interrupt Enable Rising**, **USB Interrupt Enable Falling**, and the **Carkit Interrupt Enable** registers. 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***. To ensure interrupts are detectable when **clock** is powered down, the link should enable both rising and falling edges.

If the link is in Synchronous Mode, an interrupt event causes the PHY to send an **RX CMD** byte to the Link.If the link is not in Synchronous Mode, an interrupt event causes the PHY to assert the **int** pin. When the Link detects that **int** is asserted, it wakes up the clock (if powered down), and then reads the **USB Interrupt Latch** and **Carkit Interrupt Latch** registers to determine the source of the interrupt.

3.7 Timing

3.7.1 Clock

Clock timing is summarized in Table 4~~Table 4~~. Timing for 8-bit peripherals are taken from UTMI [Ref 3]. New timing has been added for the optional 4-bit interface. ULPI also introduces several new parameters that must be filled in by the PHY vendor. If the PHY supports the optional input clock, then the Link clock must meet the requirements of Table 4~~Table 4~~ and any vendor-specific requirements.

Parameter		Symbol	Min	Nominal	Max	Units
Frequency (first transition)	8-bit ±10%	FSTART_8BIT	54	60	66	MHz
	4-bit ±5%[1]	FSTART_4BIT	57	60	63	
Frequency (steady state) ±500ppm		FSTEADY	59.97	60	60.03	MHz
Duty Cycle (first transition)	8-bit ±10%	DSTART_8BIT	40	50	60	%
	4-bit ±5%[1]	DSTART_4BIT	45	50	55	
Duty Cycle (steady state) ±500ppm		DSTEADY	49.975	50	50.025	%
Time to reach steady state frequency and duty cycle after first transition		TSTEADY			1.4	ms
Clock startup time after de-assertion of ***SuspendM***	Peripheral [2]	TSTART_DEV			5.6	ms
	Host [3]	TSTART_HOST				
PHY preparation time after first transition of input clock [4]		TPREP				us
Jitter		TJITTER				ps
Rise and fall time		TRISE/TFALL				ns

Table 4 – Clock timing parameters

[1] 4-bit clock frequency and duty cycle have been reduced by 5% to provide a usable falling edge of clock.

[2] Peripheral clock startup time is calculated to meet chirp timing, and is taken from the UTMI specification [Ref 3] section 5.22.2.3.

[3] Host clock startup time must be filled in by the PHY vendor. A startup time of less than 1ms is recommended to meet resume timing. If the PHY is used as a host and the interface clock cannot be designed to be usable within 1ms, it must internally transmit resume signalling automatically. This is illustrated in section 3.8.5.4.4.

[4] If the PHY supports the optional input clock feature, the PHY vendor must state the preparation time. The PHY must synchronize its PLL and be ready to accept a transmit command from the Link by the specified time. This is illustrated in Figure 48~~Figure 47~~. The Link clock startup time added to the PHY preparation time, must meet the 5.6ms startup time for a peripheral (UTMI specification [Ref 3], section 5.22.2.3). A total startup time of 1ms is recommended for a host.

3.7.1.1 Output Clock

The PHY must be capable of sourcing an output clock, and meet the requirements listed in Table 4~~Table 4~~.

Output clock timing for an 8-bit peripheral follows the UTMI specification [Ref 3]. The steady state frequency (F_{STEADY}) provides ±500ppm accuracy as required by USB Hi-Speed data. The startup time (T_{START_DEV}) allows a peripheral to wake up its clock and finish transmitting a Chirp-K within the maximum allowed time of 7ms (T_{START_DEV} + T_{STEADY}).

For a host with an 8-bit data bus, ULPI defines a new output clock startup time (T_{START_HOST}). When a USB host detects remote wake-up signalling, it must wake up its clock and start transmitting resume-K within the maximum allowed time of 1ms. The PHY vendor must specify this value.

For the optional 4-bit data interface, both rising and falling edges are used to clock data. To provide a usable falling edge, ULPI reduces frequency (F_{START_4BIT}) and duty cycle (D_{START_4BIT}) requirements by 5%.

In all cases, the output clock must be stopped when the PHY is suspended or when **clock** is not "usable".

3.7.1.2 Input Clock (optional)

The PHY may optionally support a 60MHz input clock from the Link, removing the need for a crystal. The PHY must drive its internal PLL from the 60MHz input clock.

PHY vendors are responsible for specifying the required input clock timing. The PHY vendor must state the required tolerances on frequency, duty cycle, jitter, rise and fall times, as shown in Table 4~~Table 4~~. The internal PHY clock preparation time, T_{PREP}, must also be specified. If necessary, the requirements should include the input capacitance, and the current sinking and sourcing capability.

The Link can choose to disable or shut down its PLL during Low Power Mode, and should not activate its clock output until its PLL is stable. An unstable Link clock may cause the PHY PLL to take longer to stabilize.

3.7.1.2.1 Input Clock Jitter

For the timing measurement planes shown in Figure 6~~Figure 6~~, the USB Specification mandates a 7.5% jitter budget on the Hi-Speed transmit eye diagram, measured at Connector A (TP2). A jitter budget of 5% is recommended at the PHY pins (TP1). The ULPI specification does not specify the clock jitter requirement at the Link output clock pin (TP0), however Link and PHY designers should minimize their clock jitter such that the total of Link plus PHY jitter meets the recommended 5% jitter budget.

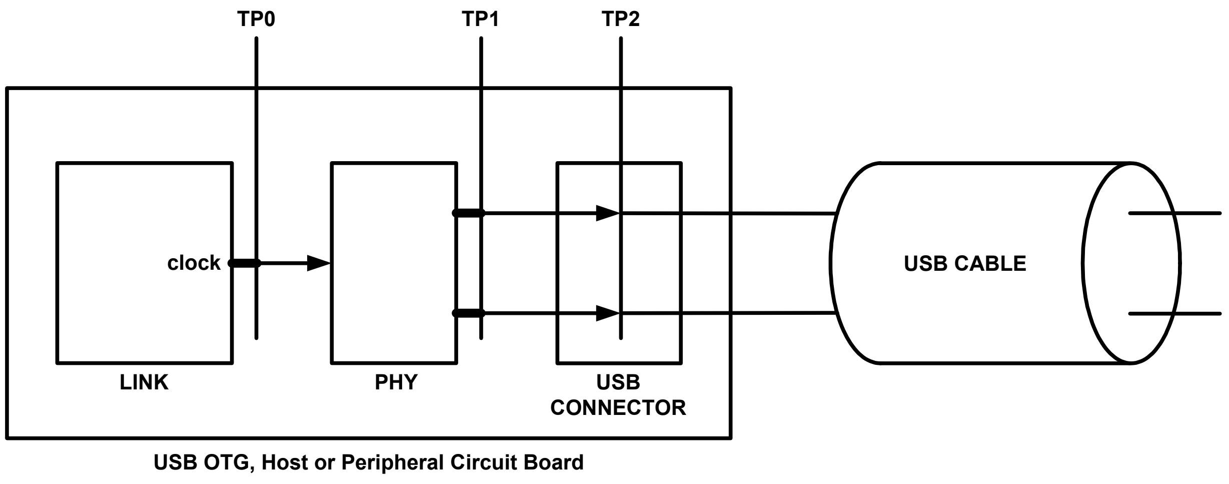

Figure 6 – Jitter measurement planes

3.7.2 Control and Data

Control and data timing requirements are given in Table 5~~Table 5~~ and illustrated in Figure 7~~Figure 7~~. These timings apply to Synchronous Mode only. All timings are measured with respect to the clock as seen at the PHY **clock** pin. Control signals and 8-bit data are always clocked on the rising edge of **clock**, while the optional double-edge 4-bit data signals are clocked on rising and falling edges.

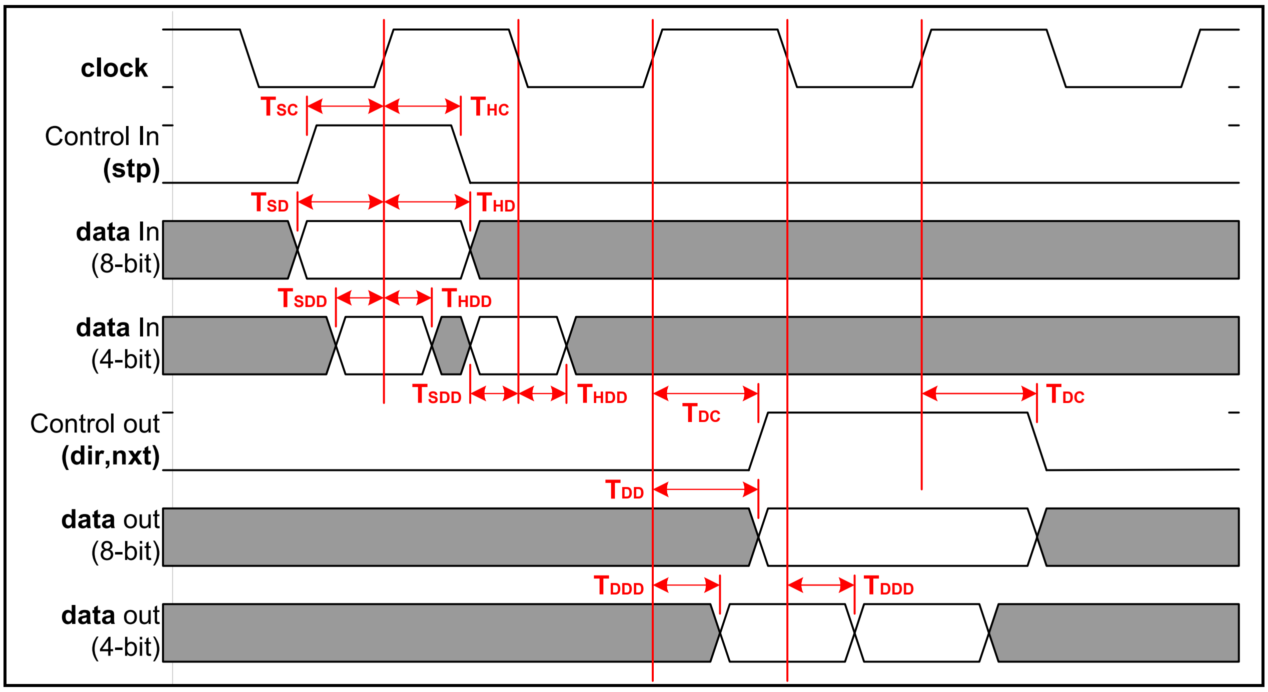

Figure 7 – ULPI timing diagram

Parameter	Symbol	Min	Max	Units
Output clock				
Setup time (control in, 8-bit data in)	TSC, TSD		6.0	ns
Hold time (control in, 8-bit data in)	THC, THD	0.0		ns
Output delay (control out, 8-bit data out)	TDC, TDD		9.0	ns
Setup time (4-bit data in) (optional)	TSDD		3.0	ns
Hold time (4-bit data in) (optional)	THDD	-0.8~~0.0~~		ns
Output delay (4-bit data out) (optional)	TDDD		4.0	ns
Input clock (optional)				
Setup time (control in, 8-bit data in)	TSC, TSD		3.0	ns
Hold time (control in, 8-bit data in)	THC, THD	1.5~~-1.5~~		ns
Output delay (control out, 8-bit data out)	TDC, TDD		6.0	ns
Setup time (4-bit data in)	TSDD		2.5	ns
Hold time (4-bit data in)	THDD	0.8~~-1.0~~		ns
Output delay (4-bit data out)	TDDD		3.5	ns

Table 5 – ULPI interface timing

3.7.2.1 4-bit Data Clocking (optional)

The PHY may optionally support a 4-bit data bus instead of an 8-bit data bus. The original 8-bit data or command is split into two 4-bit “nibbles”. The least significant nibble, **data**(3:0), is transferred first on the rising edge of **clock**. The most significant nibble, **data**(7:4), is transferred second on the falling edge of **clock**. Transferring an odd number of 4-bit nibbles is not allowed. The 4-bit data must be clocked as shown in Figure 8~~Figure 8~~. This ensures that a complete byte of data arrives at the same rising edge as 8-bit designs.

The Link and PHY must meet tighter timing as specified in Table 5~~Table 5~~ with respect to both rising and falling edges. Both output and input clock directions are allowed.

The control signals **dir**, **stp**, and **nxt** are clocked only on the rising edge of the 60MHz interface clock because data transfers are always byte-aligned.

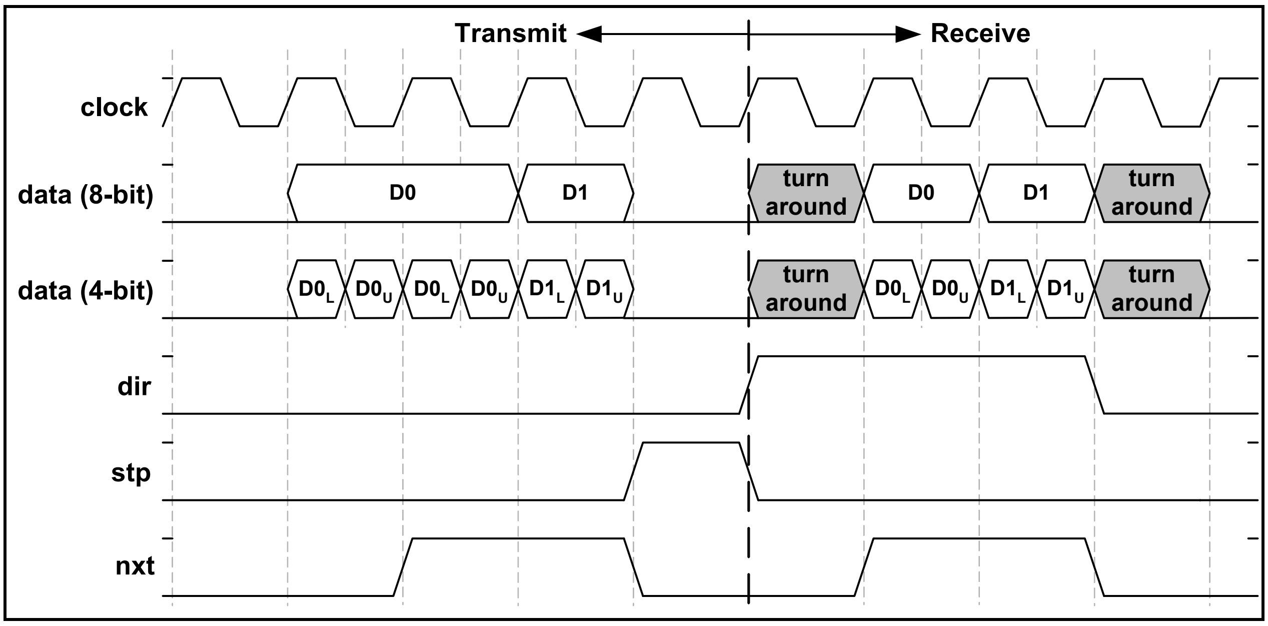

Figure 8 – Clocking of 4-bit data interface compared to 8-bit interface

3.8 Synchronous Mode

This section describes the Synchronous Mode of the UTMI+ Low Pin Interface protocol. The rules of Chapter 2 always apply unless otherwise stated in this section. ULPI also defines a register map in the PHY that is accessed by the Link.

3.8.1 ULPI Command Bytes

ULPI modifies the original UTMI data stream so that it can fit more data types. Redundancy in the PID byte during transmit is overloaded with ULPI transmit commands (**TX CMD**). Unused data bytes in the receive stream are overloaded with receive commands (**RX CMD**). ULPI defines a **Transmit Command** byte that is sent by the Link and a **Receive Command** byte that is sent by the PHY.

3.8.1.1 Transmit Command Byte (TX CMD)

The Link initiates transfers to the PHY by sending the **Transmit Command** byte of Table 6~~Table 6~~ The **TX CMD** byte consists of a 2-bit command code and a 6-bit payload.

Byte Name	Command Code data(7:6)	Command Payload data(5:0)	Command Description
Special	00b	000000b (NOOP)	No operation. 00h is the idle value of the data bus. The Link drives NOOP by default.
		XXXXXXb (RSVD)	Reserved command space. Values other than those above will give undefined behavior.
Transmit	01b	000000b (NOPID)	Transmit USB data that does not have a PID, such as chirp and resume signalling. The PHY starts transmitting on the USB beginning with the next data byte.
		00XXXXb (PID)	Transmit USB packet. **data**(3:0) indicates USB packet identifier PID(3:0).
		XXXXXXb (RSVD)	Reserved Command space. Values other than those above will give undefined behaviour.
RegWrite	10b	101111b (EXTW)	Extended reqister write command. 8-bit address available in the next cycle.
		XXXXXXb (REGW)	Register write command with 6-bit immediate address.
RegRead	11b	101111b (EXTR)	Extended register read command. 8-bit address available in the next cycle.
		XXXXXXb (REGR)	Register read command with 6-bit immediate address.

Table 6 – Transmit Command (TX CMD) byte format

3.8.1.2 Receive Command Byte (RX CMD)

The **Receive Command** byte of Table 7~~Table 7~~ is sent by the PHY to update the Link with LineState, USB receive, disconnect and OTG status information. VBUS thresholds are described in 3.8.7.3~~3.8.6.3~~.

data	Name	Description and Value				
1:0	LineState	UTMI+ LineState signals. **data**(0) = **LineState**(0) **data**(1) = **LineState**(1)				
3:2	Vbus State	Encoded Vbus Voltage state				
		Value	**VBUS Voltage**	**SessEnd**	**SessValid**	**VbusValid**[3]
		00	VBUS < VB_SESS_END	1	0	0
		01	VB_SESS_END ≤ VBUS < VSESS_VLD	0	0	0
		10	VSESS_VLD ≤ VBUS < VA_VBUS_VLD	X~~0~~	1	0
		11	VA_VBUS_VLD ≤ VBUS	X~~0~~	X~~1~~	1
5:4	RxEvent	Encoded UTMI event signals				
		Value	**RxActive**	**RxError**	**HostDisconnect**[4]	
		00	0	0	0	
		01	1	0	0	
		11	1	1	0	
		10	X	X	1	
6	ID	Set to the value of **IdGnd** (UTMI+ **IdDig)**. Valid 50ms after ***IdPullup*** is set to 1b.				
7	alt_int	Asserted when a non-USB interrupt occurs. This bit must be set when an unmasked event occurs on any bit in the **Carkit Interrupt Latch** register. The Link must read the **Carkit Interrupt Latch** register to determine the source of the interrupt.				

Table 7 – Receive Command (RX CMD) byte format

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

[4] **Hostdisconnect** state must be indicated to the Link only in host mode (***DpPulldown*** and ***DmPulldown*** both set to 1b). When in peripheral mode, **Hostdisconnect** must be ignored and must not mask events on **RxActive** or **RxError**.

3.8.1.3 When to send an RX CMD

An **RX CMD** is sent only in Synchronous Mode, and conveys two types of information to the Link. The first is USB receive information. The second is interrupt events. All information is encoded into the single **RX CMD** byte of Table 7~~Table 7~~.

USB receive information includes **LineState, RxActive** and **RxError.** The sending of **RX CMD**'s for USB receive data is covered in 3.8.2.4. After a USB transmit, the PHY must send an **RX CMD** with **LineState** indicating EOP to the Link. For Hi-Speed, an EOP is the !squelch to squelch transition on **LineState**. For Full Speed and Low Speed, an EOP is the transition from SE0 to J on **LineState**. This is illustrated in section 3.8.2.2.

Interrupt events include **Hostdisconnect**, **Vbus**, **IdGnd**, and alternative sources such as Carkit interrupts. An **RX CMD** is sent to the Link whenever these events are detected, and the corresponding **USB Interrupt Enable Rising** or **USB Interrupt Enable Falling** registers are set.

Figure 9~~Figure 9~~ illustrates how the PHY sends **RX CMD** information to the Link. The first packet shows a single **RX CMD**. If back-to-back changes are detected, the PHY keeps **dir** asserted and drives back-to-back **RX CMD**'s, as shown in the second packet of Figure 9~~Figure 9~~. The Link must be able to accept any number of continuous, back-to-back **RX CMD**'s.

An **RX CMD** has lower priority than USB receive and transmit data, but has higher priority than Register Read and Write commands. If the ULPI bus is busy with USB data, the **RX CMD** will be queued in the PHY and sent when the ULPI bus is available. When sent to the Link, a queued **RX CMD** must always convey the current **RX CMD** values, not a previous or old value. The PHY must also send **RX CMD**'s to the Link when **nxt** is de-asserted during USB receive packets.

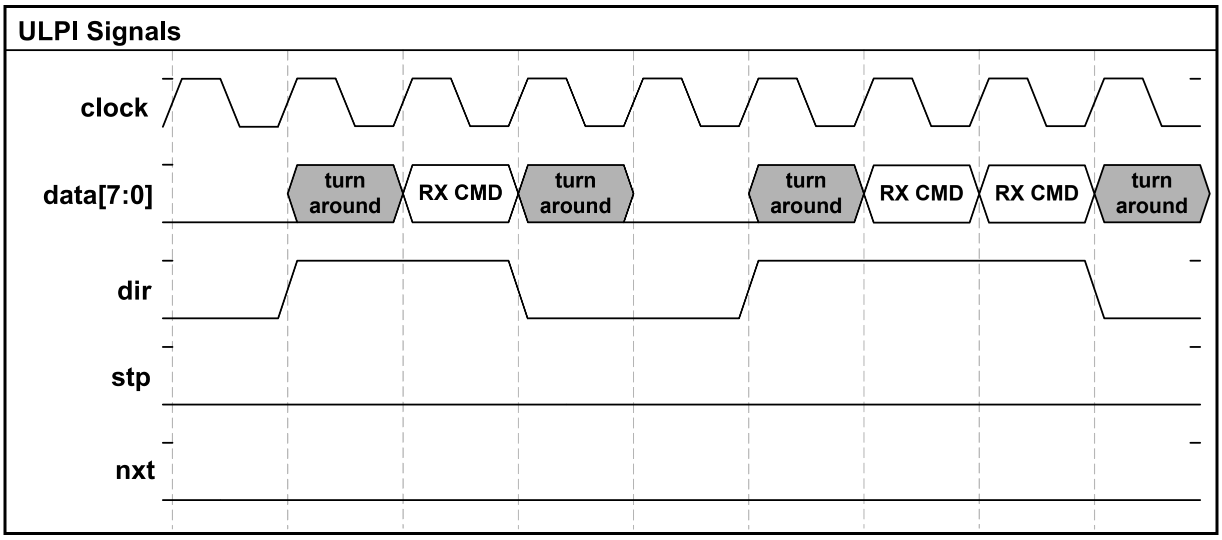

Figure 9 – Sending of RX CMD

3.8.2 USB Packets

This section describes how USB packets are transmitted and received over the ULPI bus. Limits on PHY and Link processing delays are given so that USB inter-packet delays can be met.

3.8.2.1 USB Data Transmit (NOPID)

As shown in Figure 10~~Figure 10~~, to transmit data on the USB that does not contain a Packet Identifier (PID), the Link sends a **TX CMD** byte of the **NOPID** type. The PHY must assert **nxt** to throttle the data coming from the Link. The PHY must not assert **nxt** in the first cycle of the **TX CMD**, and must de-assert **nxt** when it detects **stp** high. Since this command does not contain PID data, the PHY must wait for the next data byte before beginning transmission on the USB. When the last byte has been consumed by the PHY, the Link asserts **stp** for 1 cycle, and drives **data** to 00h if no transmit errors occurred. The Link must not assert **stp** before the first byte has been consumed by the PHY.

This command must be used for chirp and resume signalling, which are detailed in sections 3.8.5.1 and 3.8.5.3, respectively. For such signalling, the ***OpMode*** bits in the **Function Control** register must first be set to 10b. For chirp and resume signalling, the PHY does not need to insert stuff bits and so must hold **nxt** high until it sees the **stp** pulse.

While the Link is driving the ULPI bus with a **NOPID** packet, the PHY should not assert **dir** unless it wants to abort the packet. 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.~~**RX CMD** updates will be queued in the PHY and sent when the ULPI bus is available. When sent to the Link, a queued **RX CMD** must always convey the current **RX CMD** values, not a previous or old value.~~

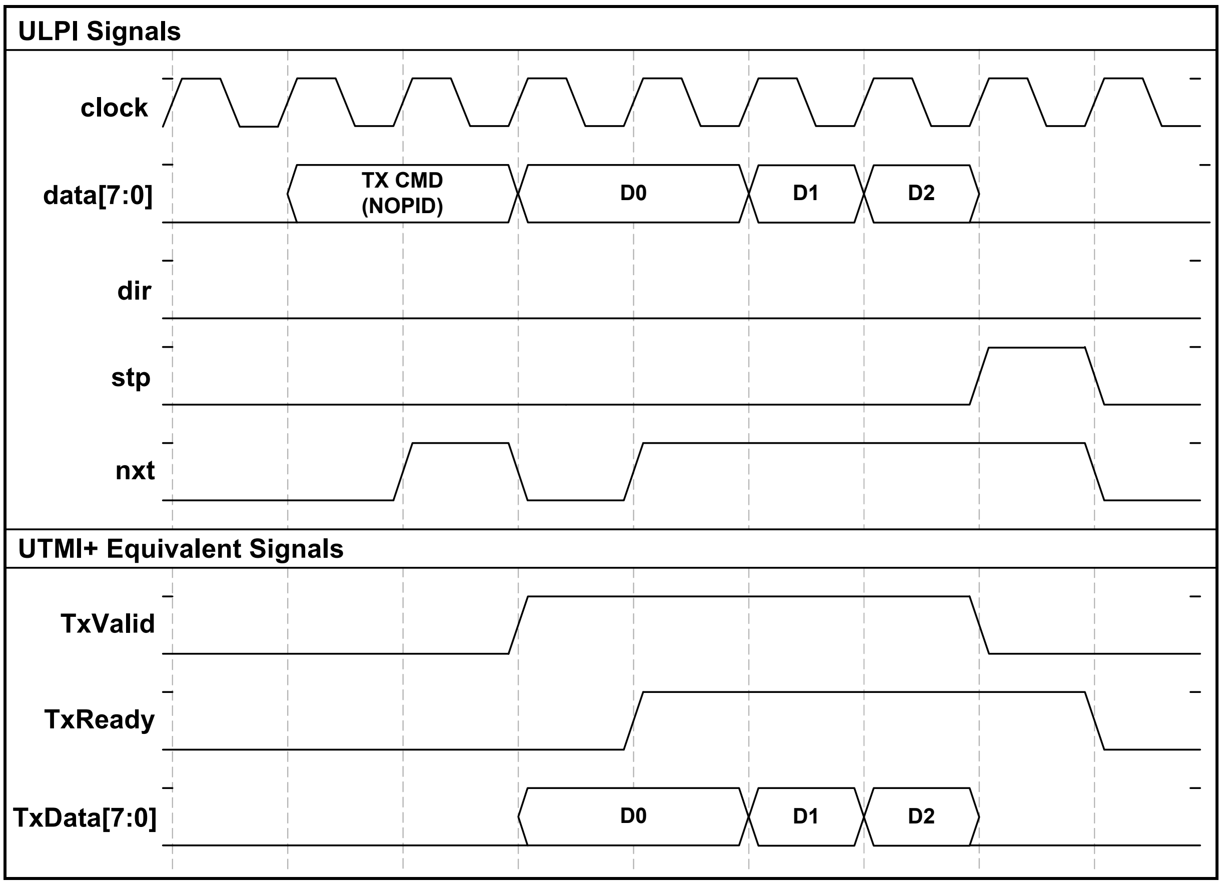

Figure 10 – USB data transmit (NOPID)

3.8.2.2 USB Packet Transmit (PID)

As shown in Figure 11~~Figure 11~~, to transmit a USB packet, the Link first drives a **TX CMD** byte. As in Table 6~~Table 6~~, the Link sets the Command Code to 01b (Transmit), and places the USB Packet Identifier (PID) on **data**(3:0). The PHY throttles the data using **nxt** such that the Link provides the next byte in the cycle after **nxt** is detected as high. This is identical to UTMI's **DataIn** and **TxReady**. When the last byte has been consumed by the PHY, the Link asserts **stp** for 1 cycle, and drives **data** to 00h if no transmit errors occurred. The Link must not assert **stp** before the first byte has been consumed by the PHY.

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.

After asserting **stp**, the Link cannot transmit another packet until the first packet has completed on the USB. The Link must follow the packet timings given in section 3.8.2.6, however if an **RX CMD** indicating end-of-packet (EOP) is received before the given timings, the Link can begin transmitting the next packet, as shown in Figure 12~~Figure 12~~. The PHY must always send the **RX CMD** indicating EOP.

While the Link is driving the ULPI bus with a **PID** packet, the PHY should not assert **dir** unless it wants to abort the packet. 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.~~**RX CMD** updates will be queued in the PHY and sent when the ULPI bus is available. When sent to the Link, a queued **RX CMD** must always convey the current **RX CMD** values, not a previous or old value.~~

As clarified in UTMI+ [Ref 4], the PHY must internally block the USB receive path during transmit.

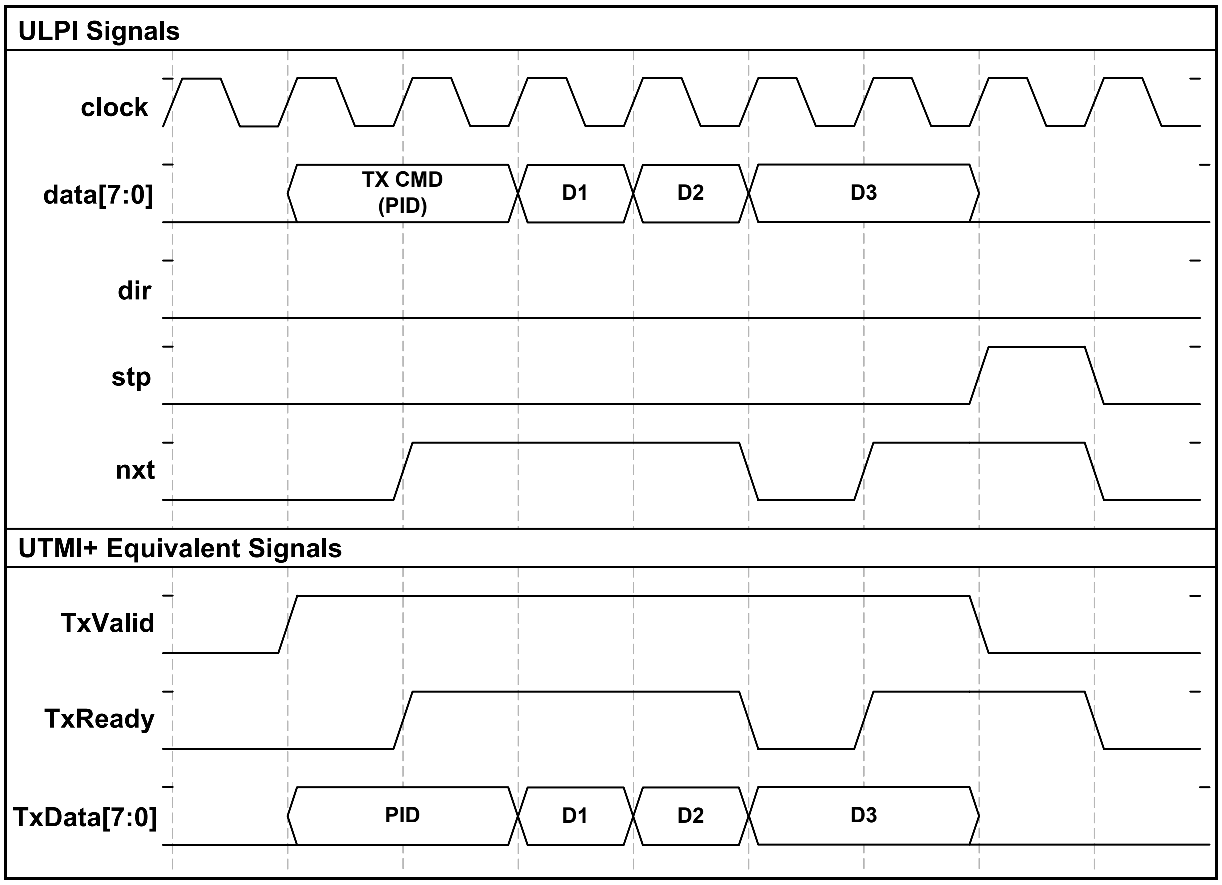

Figure 11 – USB data transmit (PID)

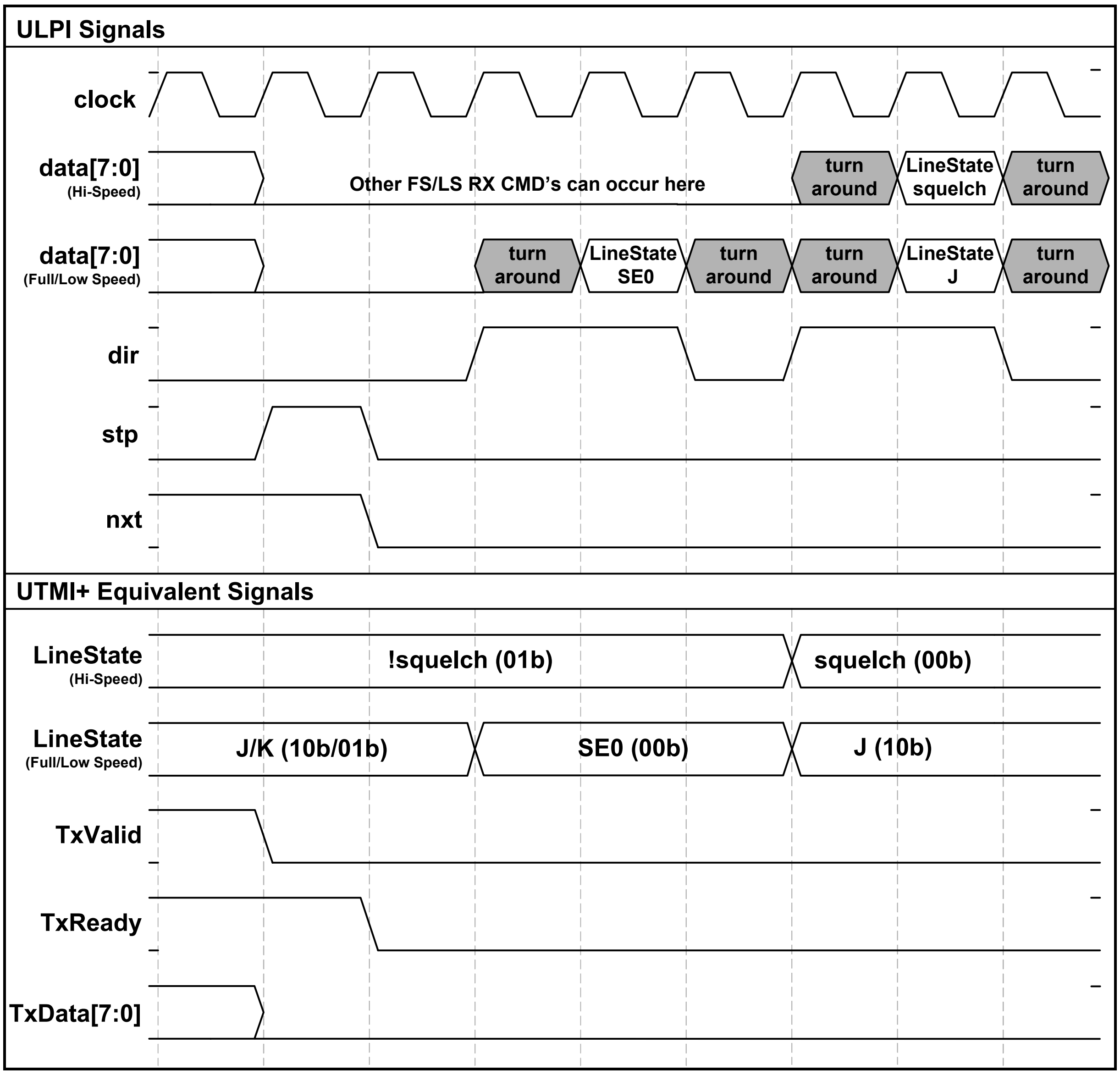

Figure 12 – PHY drives an RX CMD to indicate EOP (FS/LS LineState timing not to scale)

3.8.2.3 USB Transmit Error

If the Link experiences a buffer under-run during the transmission of a Full Speed or Low Speed USB packet, it must insert a bit-stuff error into the packet before the packet ends. To force a transmit error, the Link drives FFh onto **data** at the end of the packet, in the same cycle it asserts **stp**. This is shown as the last, grey-colored data byte in Figure 11~~Figure 11~~. The Link must drive no more than one byte to FFh, and the PHY will automatically generate the Full Speed transmit error by sending a minimum of 8 consecutive 1's on the USB bus. The Link must wait for an **RX CMD** indicating the SE0-to-J transition before transmitting another packet. The Link must not assert **stp** before the first byte has been consumed by the PHY.

This sequence replaces UTMI's method of changing **OpMode** from 00b to 10b during the last byte of transmission.

To force a Hi-Speed transmit error, the Link must transmit an inverted CRC at the end of the packet.

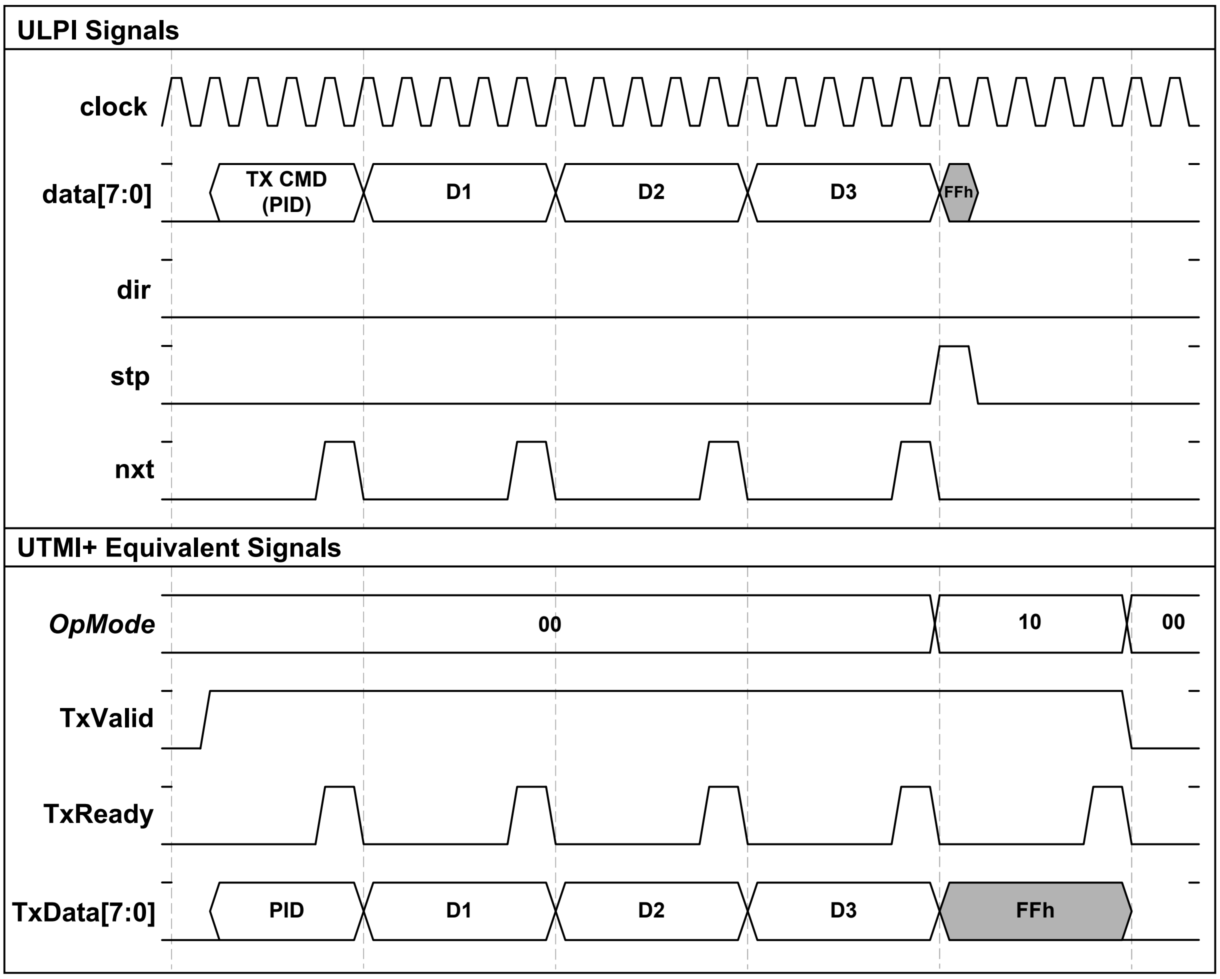

Figure 13 – Forcing a full/low speed USB transmit error (timing not to scale)

3.8.2.4 USB Packet Receive

As shown in Figure 14~~Figure 14~~, when the PHY is receiving USB data, it first gains ownership of the data bus by asserting **dir**. The PHY asserts **dir** in the same cycle that UTMI's **RxActive** asserts. As shown in Figure 14~~Figure 14~~, if **dir** was previously low, the PHY will assert both **dir** and **nxt** so that the Link knows immediately that this is a USB receive packet. If **dir** was previously high, as seen in Figure 15~~Figure 15~~, the PHY will de-assert **nxt** and drive an **RX CMD** with the RxEvent field set to 01b (RxActive state). The PHY can start driving **data** in the following cycle, or will output **RX CMD's** until USB data is available. Valid USB packet data is presented to the Link by asserting **nxt** and placing a byte onto the bus. When **nxt** is low, the PHY drives the **RX CMD** byte. All **RX CMD** changes during the USB packet receive must be signalled 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.

While the data stream has been modified from UTMI's **DataOut**, **nxt** is identical to **RxValid**. The Link considers a packet to be completed when the **RX CMD** byte shows **RxActive** is set to 0b, or **dir** is de-asserted, whichever occurs first.

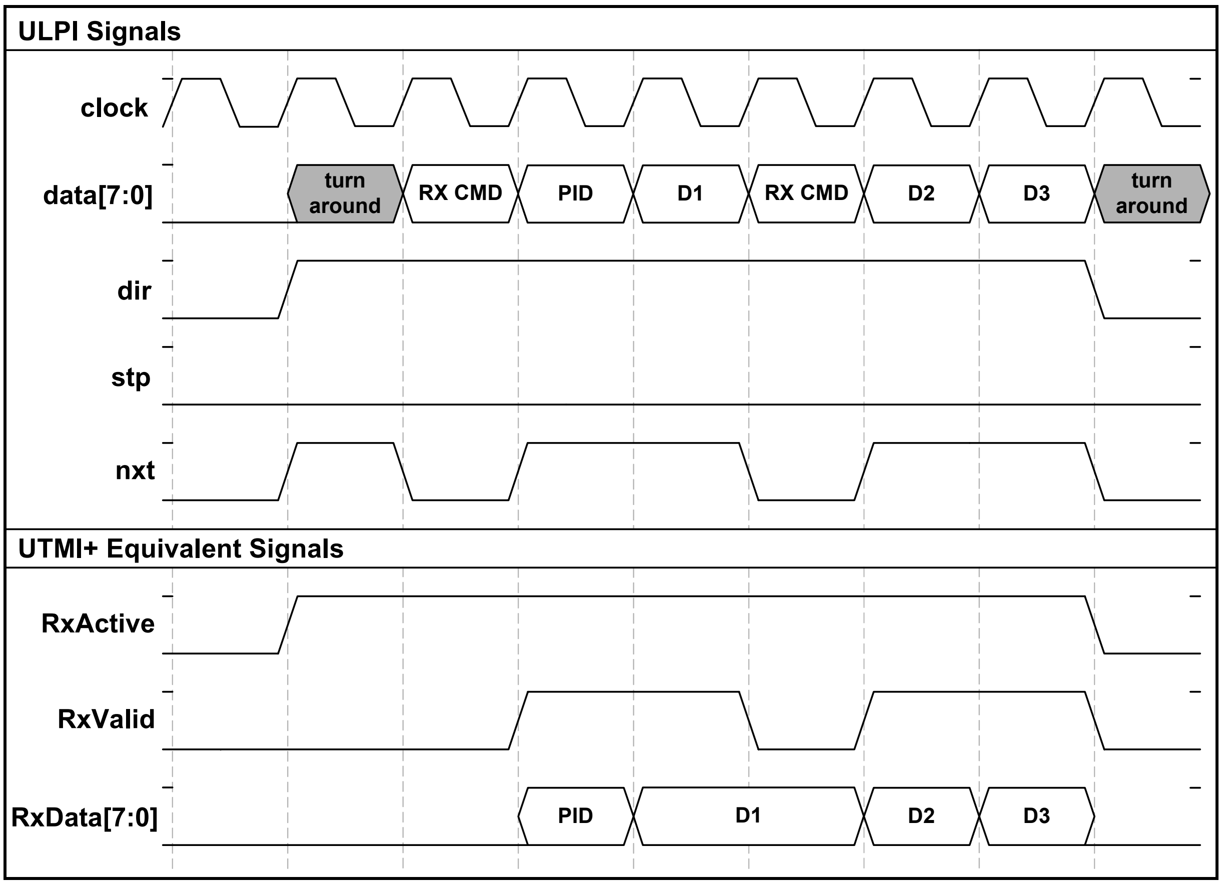

Figure 14 – USB receive while dir was previously low

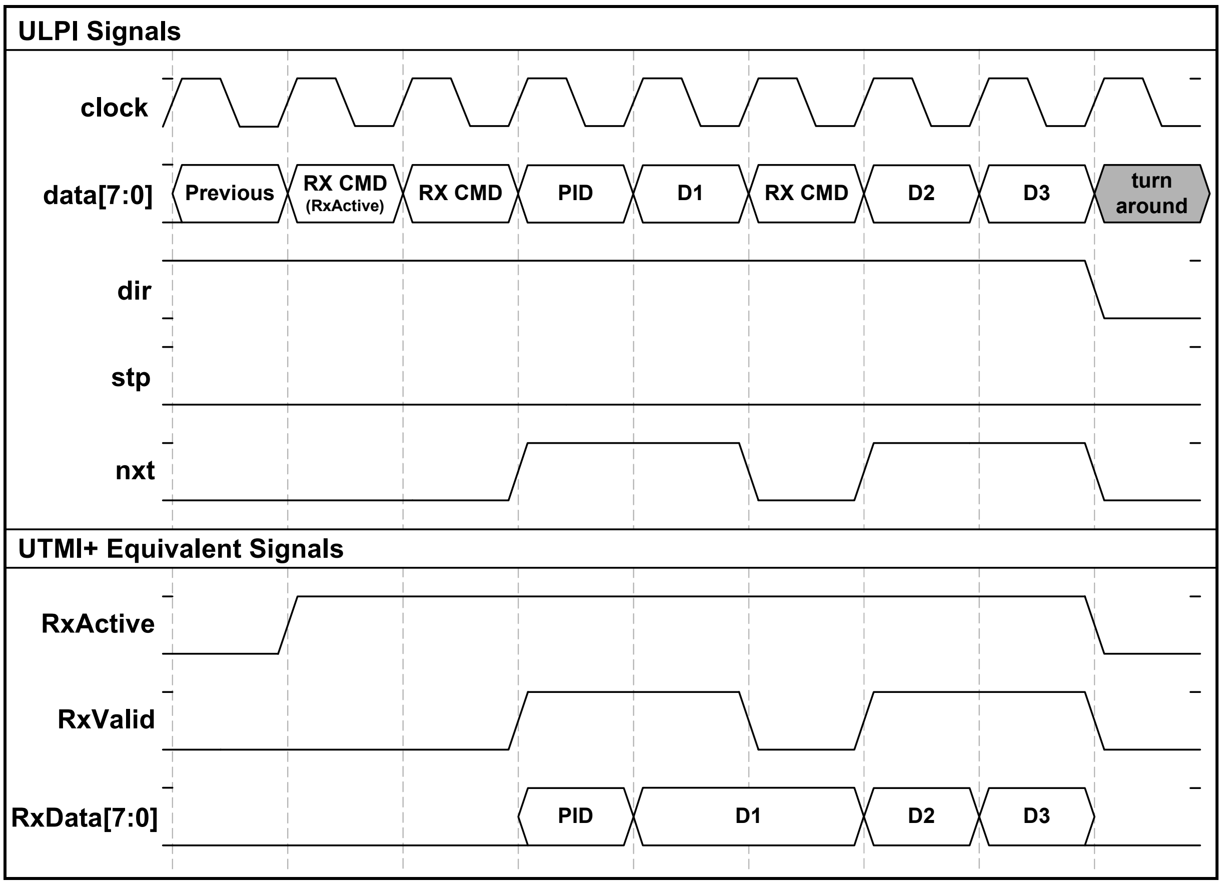

Figure 15 – USB receive while dir was previously high

3.8.2.5 USB Receive Error

If the PHY detects a USB packet receive error, it ignores any currently received data byte, de-asserts **nxt** and drives **RX CMD** bytes with the RxEvent field set to 11b (RxError state). The PHY de-asserts **dir** when **RxActive** is de-asserted. The RxError state is identical to UTMI's **RxError** signal.

All PHY implementations must set **RxError** when a Full Speed bit-stuff error is detected. The PHY can optionally set **RxError** for a Hi-Speed EOP that is not aligned to a byte boundary, for data overflow, data underflow, or another valid receive error condition. The PHY does not assert **RxError** for a single Full Speed dribble bit prior to the EOP[5].

Figure 16~~Figure 16~~ shows the behaviour of the ULPI interface when a bit-stuff error is detected mid-packet. Figure 17~~Figure 17~~ shows the behaviour when a bit-stuff error occurs in the final CRC byte.

The Link must ignore packets with a receive error.

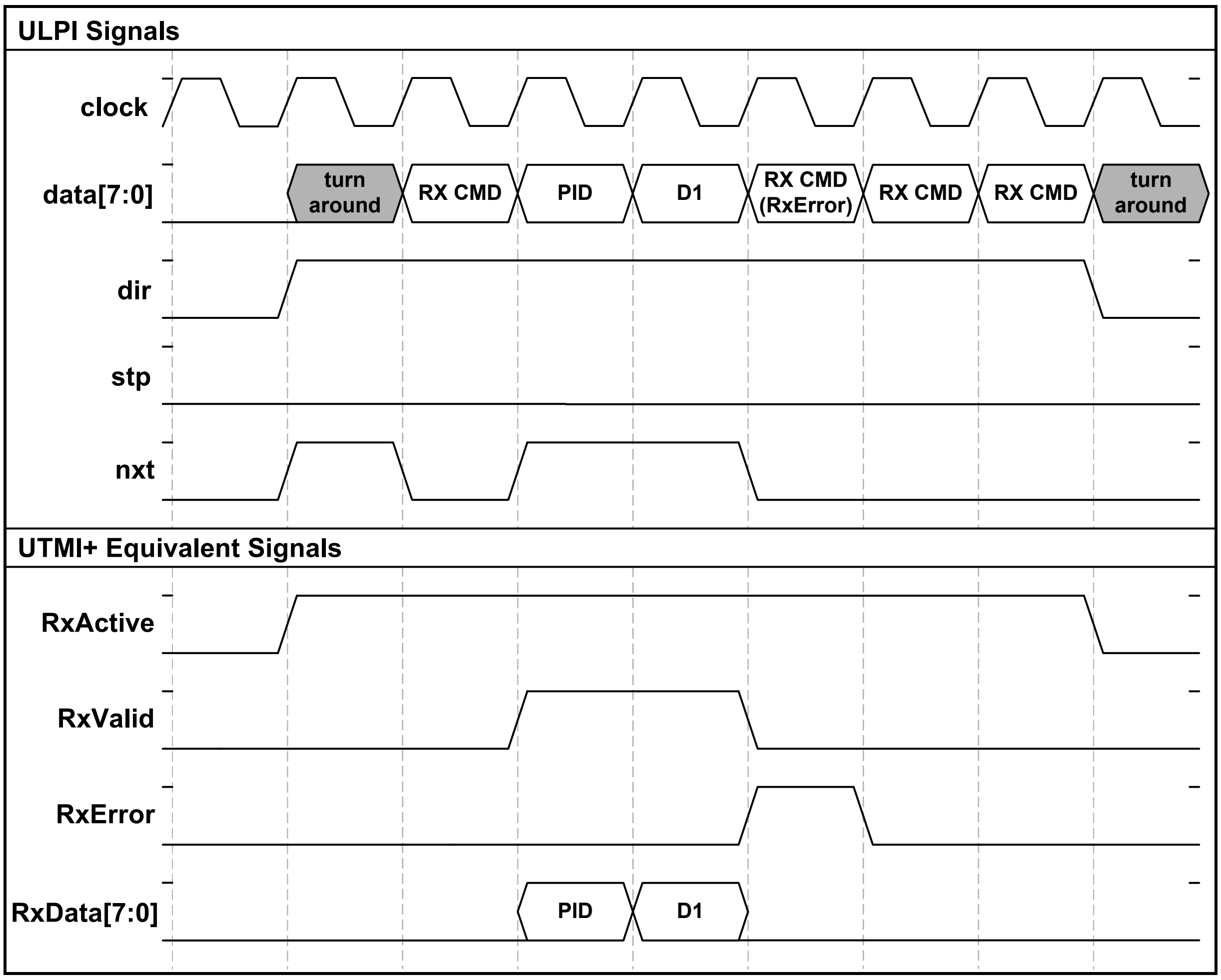

Figure 16 – USB receive error detected mid-packet

[5] For an explanation of dribble bit see USB Specification [Ref 1] 7.1.9.1.

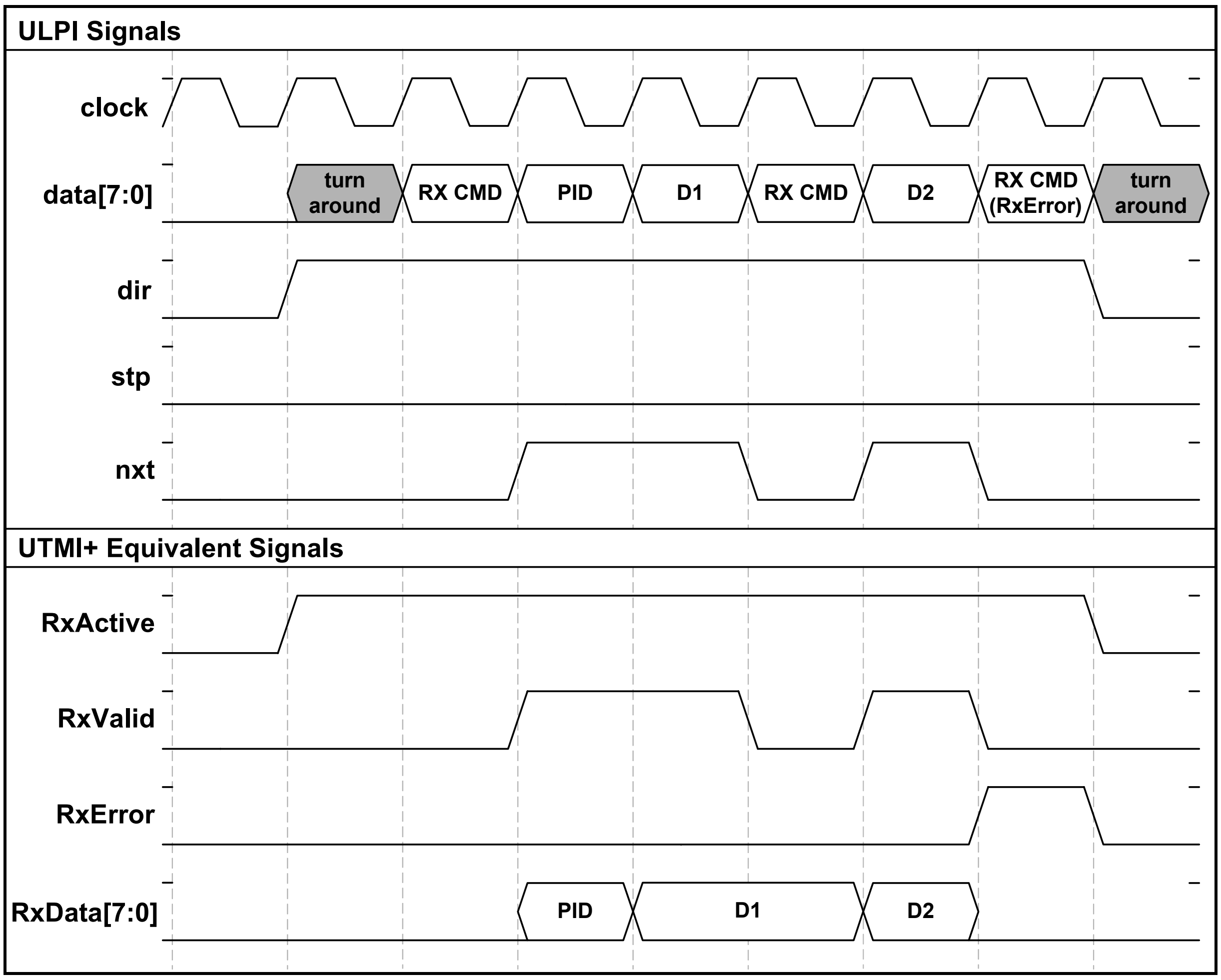

Figure 17 – USB receive error during the last byte

3.8.2.6 USB Packet Timing

The USB specification [Ref 1] defines inter-packet timing. The ULPI specification relates these timings back to the ULPI interface by specifying a budget of clock cycles available to the PHY and Link. The PHY must process data within the given timing range. Similarly, the Link must respond within the given decision time, or timeout if a response is not received within the given timing range. The following sections detail all required ULPI inter-packet timing, and are derived from both the USB specification [Ref 1] and UTMI PHY processing and synchronization delays[6].

3.8.2.6.1 USB Inter-packet delay and Packet timeout

Table 8~~Table 8~~ is provided as background information. Full Speed inter-packet delays are measured from the SE0-to-J transition of the first packet, to the start of the SYNC pattern on the second packet. Hi-Speed inter-packet delays are measured from the time the bus enters the idle state at the end of the first packet, to the time the bus leaves the idle state at the start of the second packet. There are 8 HS bit times in one clock cycle, 5 clock cycles in one FS bit time, and 40 clock cycles in one LS bit time, as shown in Figure 18~~Figure 18~~.

Packet Sequence	HS bit times		FS/LS bit times		Definition
	Min	Max	Min	Max	
Transmit-Transmit (host only)	88	192	2	6.5	Two consecutive host packets must have an inter-packet delay within this range.
Receive-Receive (peripheral only)	32	-	2	-	Peripherals must be capable of receiving back-to-back packets with this timing.
Receive-Transmit (host or peripheral)	8	192	2	6.5	Host or peripheral receives a packet and must transmit a response within the given timing range.
Transmit-Receive (host or peripheral)	736	816	16	18	Host or peripheral transmits a packet and will timeout within this range if a response is not received[7].

Table 8 – USB specification inter-packet timings

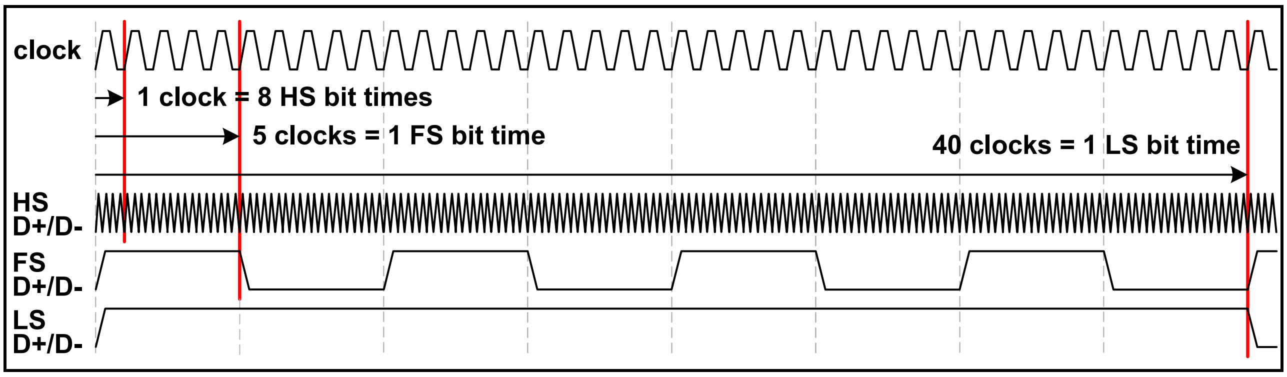

Figure 18 – USB HS, FS, and LS bit lengths with respect to clock

[6] PHY timings are taken from the UTMI Specification [Ref 3]. Implementers can refer to UTMI for architectural pointers.
[7] Timing includes a maximally connected USB system consisting of a return trip through 6 USB cables, 5 external hubs, and the downstream host or peripheral. Refer to USB specification 7.1.19 for more information.

3.8.2.6.2 PHY Pipeline Delays

Any PHY using the ULPI interface must comply to the timings given in Table 9~~Table 9~~. USB bus events are measured relative to transitions on **D+** and **D-** lines. ULPI interface timings are measured relative to the **clock** edge on which the transition is detected (e.g. **clock** edge on which **stp** is detected by the PHY).

Parameter Name	HS PHY Delay	FS PHY Delay	LS PHY Delay	Definition
RX CMD Delay[8]	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.
TX Start Delay	1-2	1-10	1-10	Number of clocks between the PHY detecting a **TX CMD** on the ULPI bus to transmitting the first K of the SYNC pattern on the USB bus.
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.
RX Start Delay	3-8	N/A	N/A	Number of clocks after first K of SYNC pattern is seen on the USB bus to the simultaneous assertion of **dir** and **nxt**, or an **RX CMD** indicating **RxActive**. Used for HS packets only. For FS and LS packets the Link must look for **RX CMD** bytes indicating J-to-K transition.
RX End Delay	3-8	17-18	122-123	Number of clocks after EOP occurs on the USB bus to the PHY de-asserting **dir** or indicating **RxActive** low in an **RX CMD** byte. HS EOP is measured when all 8 consecutive 1's have finished transmitting on the USB bus. FS and LS EOP occurs when SE0 starts on the USB bus. For FS and LS, the Link uses **LineState** and not **RX End Delay** to time USB packets.

Table 9 – PHY pipeline delays

[8] Same as UTMI's LineState delay of 2-3 clocks, but with 1 extra clock cycle added to the maximum due to ULPI bus turnaround during an RX CMD.

3.8.2.6.3 Link Decision Time

Table 10~~Table 10~~ shows the time allotted to the ULPI Link for the various packet sequences. When the Link transmits the second packet, it must meet the inter-packet delay. When the Link is expecting a response to a transmitted packet, it must time the inter-packet delay to detect a timeout. For back-to-back receive packets, the Link must be capable of receiving them with the timing given.

Packet Sequence	HS Link Delay	FS Link Delay	LS Link Delay	Definition
Transmit-Transmit (host only)	15-24	7-18[9]	77-247	Number of clocks a host link must wait before driving the **TX CMD** for the second packet. In HS, the link starts counting from the assertion of **stp** for the first packet. In FS, the link starts counting from the **RX CMD** indicating **LineState** has transitioned from SE0-to-J for the first packet. The timings given ensure inter-packet delays of 2-6.5 bit times.
Receive-Transmit (host or peripheral)	1-14	7-18[9~~8~~]	77-247	Number of clocks the link must wait before driving the **TX CMD** for the transmit packet. In HS, the link starts counting from the end of the receive packet (de-assertion of **dir** or an **RX CMD** indicating **RxActive** is low). In FS/LS, the link starts counting from the **RX CMD** indicating **LineState** has transitioned from SE0-to-J for the receive packet. The timings given ensure inter-packet delays of 2-6.5 bit times.
Receive-Receive (peripheral only)	1	1	1	Minimum number of clocks between consecutive receive packets. The link must be capable of receiving both packets.
Transmit-Receive (host or peripheral)	92[10]	80	718	Host or peripheral transmits a packet and will timeout after this amount of clock cycles if a response is not received[7~~6~~]. Any subsequent transmission can occur after this time.

Table 10 – Link decision times

[9] Same as UTMI's FS SIE Decision Time of 7-19 clocks, but with 1 clock cycle subtracted from the maximum due to the ULPI PHY requiring a bus turnaround during the RX CMD.
[10] The UTMI specification [Ref 3] Hi-Speed transmit to receive timing is incorrect and should be ignored.

3.8.2.6.4 Inter-packet Timing Diagrams

Figure 19~~Figure 19~~ and Figure 20~~Figure 20~~ illustrate the PHY pipeline delays and the Link decision time for the various packet sequences.

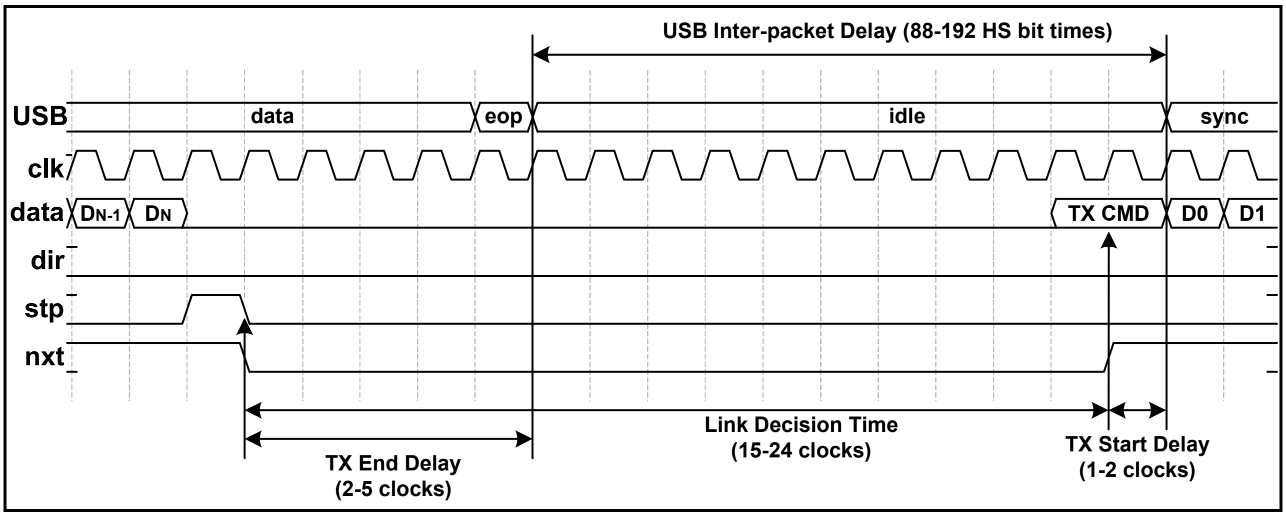

Figure 19 – HS transmit-to-transmit packet timing

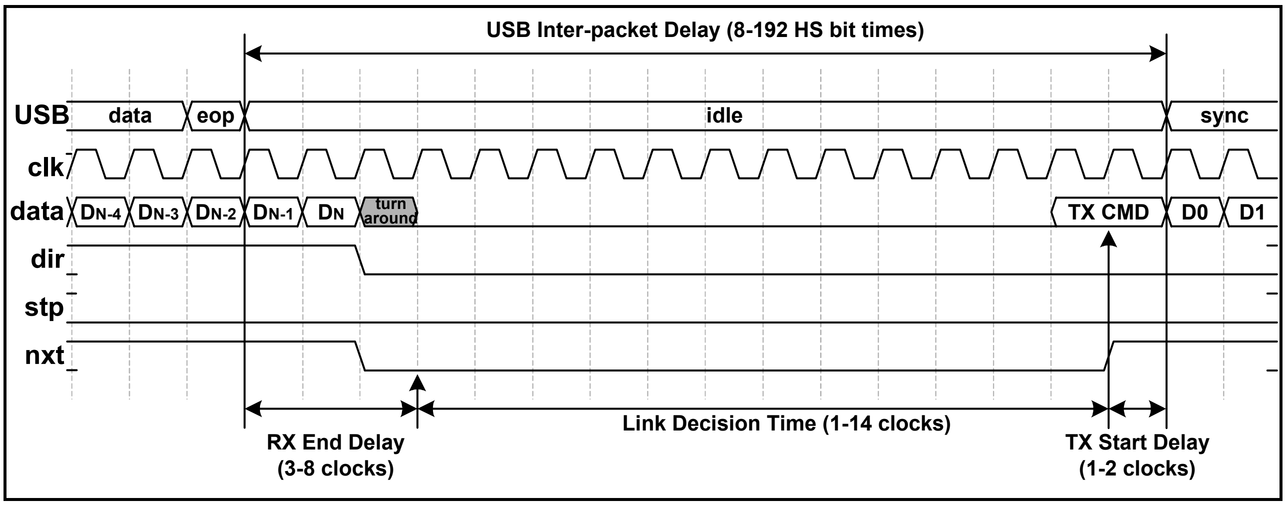

Figure 20 – HS receive-to-transmit packet timing.

3.8.3 Register Operations

ULPI provides a set of registers in the PHY that are accessed by the Link to control PHY functionality. The ULPI register set is detailed in chapter 4. Where possible, registers follow the input signals defined in the UTMI+ specification [Ref 4]. The Link can read or write register bytes, set and clear register bits as needed using the **TX CMD** byte of Table 6~~Table 6~~.

3.8.3.1 Immediate Register Read and Write

Immediate registers are accessed by first sending the **TX CMD** byte as either a RegWrite or a RegRead command with the required register address.

For a register write, as shown in Figure 21~~Figure 21~~, the Link sends a register write command and waits for **nxt** to assert. In the cycle after **nxt** asserts, the Link sends the register write data and waits for **nxt** to assert again. When **nxt** asserts the second time, the Link asserts **stp** in the following cycle to complete the operation. The PHY must detect **stp** assertion before it can accept another transmit command. If the PHY aborts the RegWrite by asserting **dir**, the Link must retry the RegWrite when the bus is idle.

For a register read, as shown in Figure 22~~Figure 22~~, the Link sends a register read command and waits for **nxt** to assert. In the cycle after **nxt** asserts, the PHY asserts **dir** to gain control of the data bus. In the cycle after **dir** asserts, the PHY must return the register read data. The PHY does not assert **nxt** when **dir** is asserted during the register read operation, even during the cycle that register read data is being returned. This allows a USB receive to always over-ride the register read during any cycle, and is explained in 3.8.3.2. If the PHY aborts the RegRead by asserting **dir** earlier than shown in Figure 22~~Figure 22~~, the Link must retry the RegRead when the bus is idle.

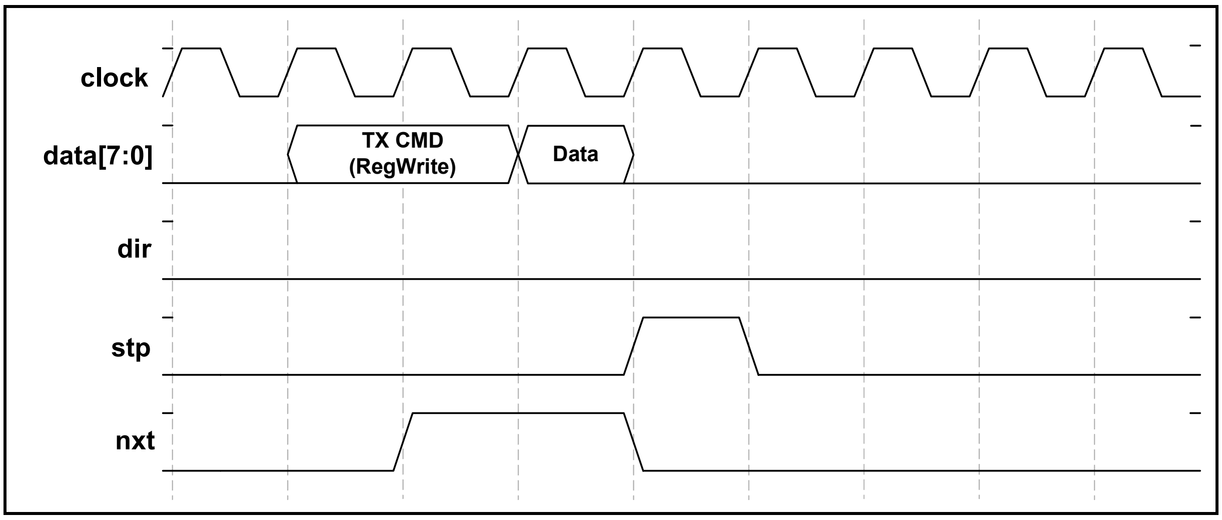

Figure 21 – Register write

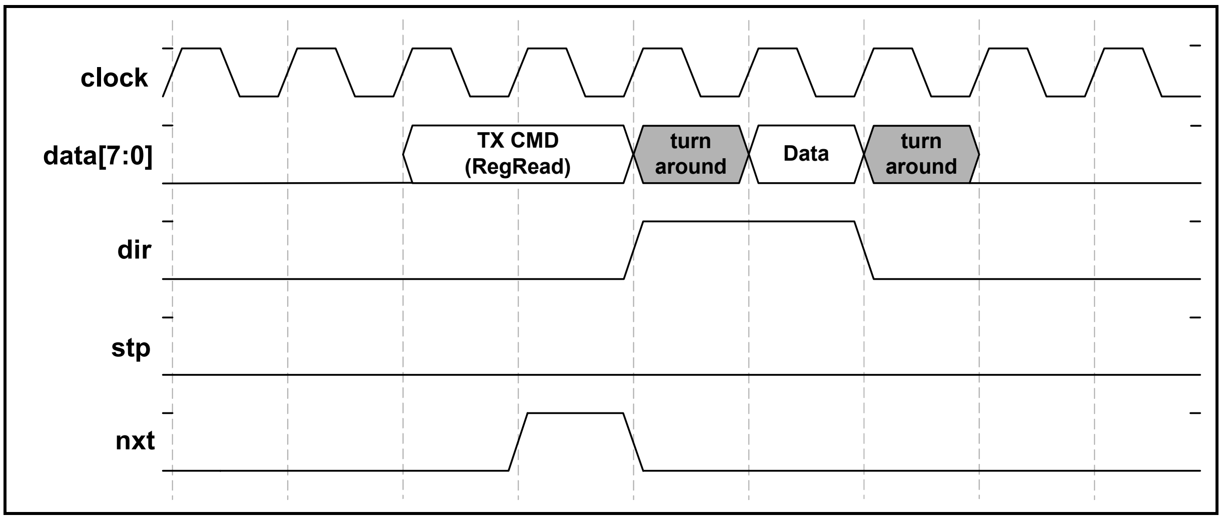

Figure 22 – Register read

3.8.3.2 Immediate Register Read and Write Aborted by USB Receive

A register read is the only instance where ULPI does not use **nxt** to throttle data. The **nxt** signal is asserted only during USB receive so that the Link can always distinguish a USB receive from other data transfers.

Figure 23~~Figure 23~~ shows a register read or write being aborted by a USB receive during the initial **Transmit Command** byte. Figure 24~~Figure 24~~ shows a register read being aborted by a USB receive in the same cycle that register read data was to be returned to the Link. In both cases, the PHY asserts both **dir** and **nxt** to indicate **RxActive**.

Register read and write operations are also aborted by the PHY sending an **RX CMD**, except during the cycle that register read data is being returned to the Link.

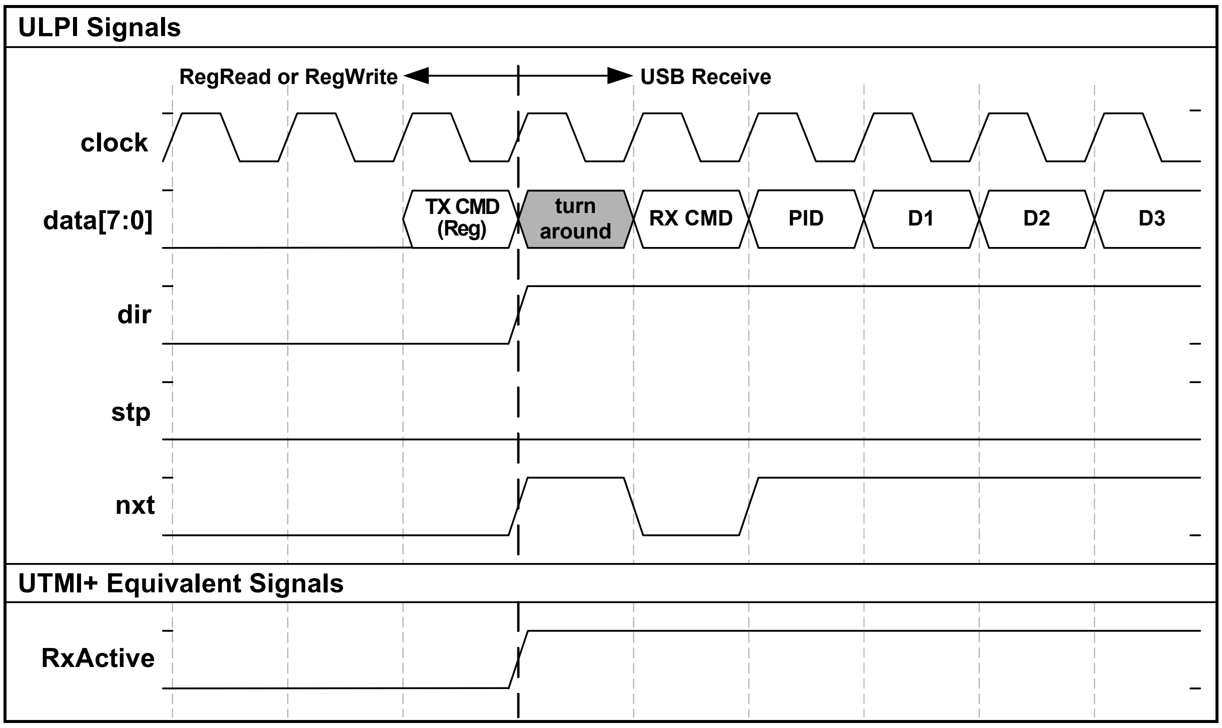

Figure 23 – Register read or write aborted by USB receive during TX CMD byte

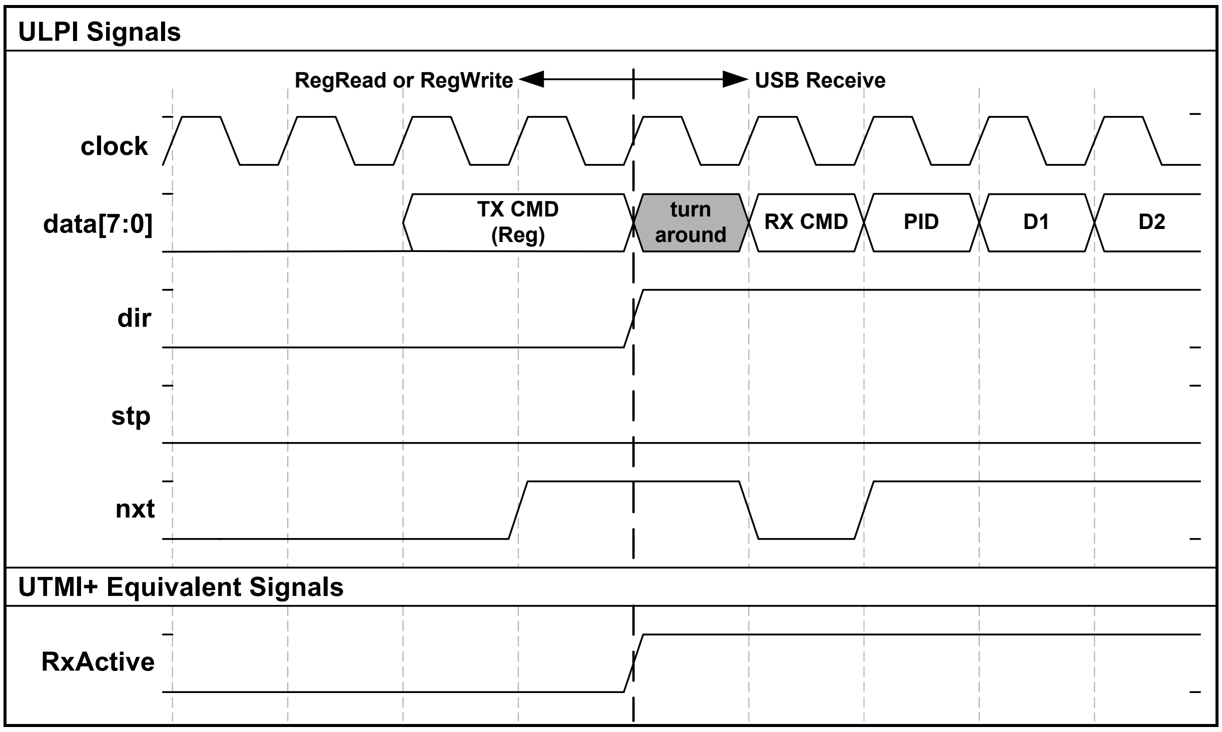

Figure 24 – Register read turnaround cycle or Register write data cycle aborted by USB receive

3.8.3.3 Back-to-back Immediate Register Read/Write and USB Receive

Figure 25~~Figure 25~~ shows a USB receive occurring in the same cycle that register read data is returned to the Link. The PHY must first return the register read data, not an **RX CMD** byte indicating **RxActive**. The PHY indicates **RxActive** in the next cycle, placing the USB Receive data back-to-back with the register read.

Figure 26~~Figure 26~~ shows a USB receive occurring in the cycle immediately after a register read completes. The PHY places the USB receive data back-to-back with the register read.

The Link must be able to accept back-to-back packets where **dir** does not de-assert between packets.

As shown in Figure 27~~Figure 27~~, if **dir** asserts during the same cycle that **stp** is asserted at the end of a register write operation, the PHY considers the register write to have completed successfully. The PHY does not de-assert **dir** because of **stp**.

Figure 28~~Figure 28~~ shows a USB receive starting in the cycle after the register read data is returned to the Link. Note the two cycles of bus turnaround when dir de-asserts for a single cycle.

In all cases, the PHY asserts **dir** and **nxt** to indicate **RxActive**.

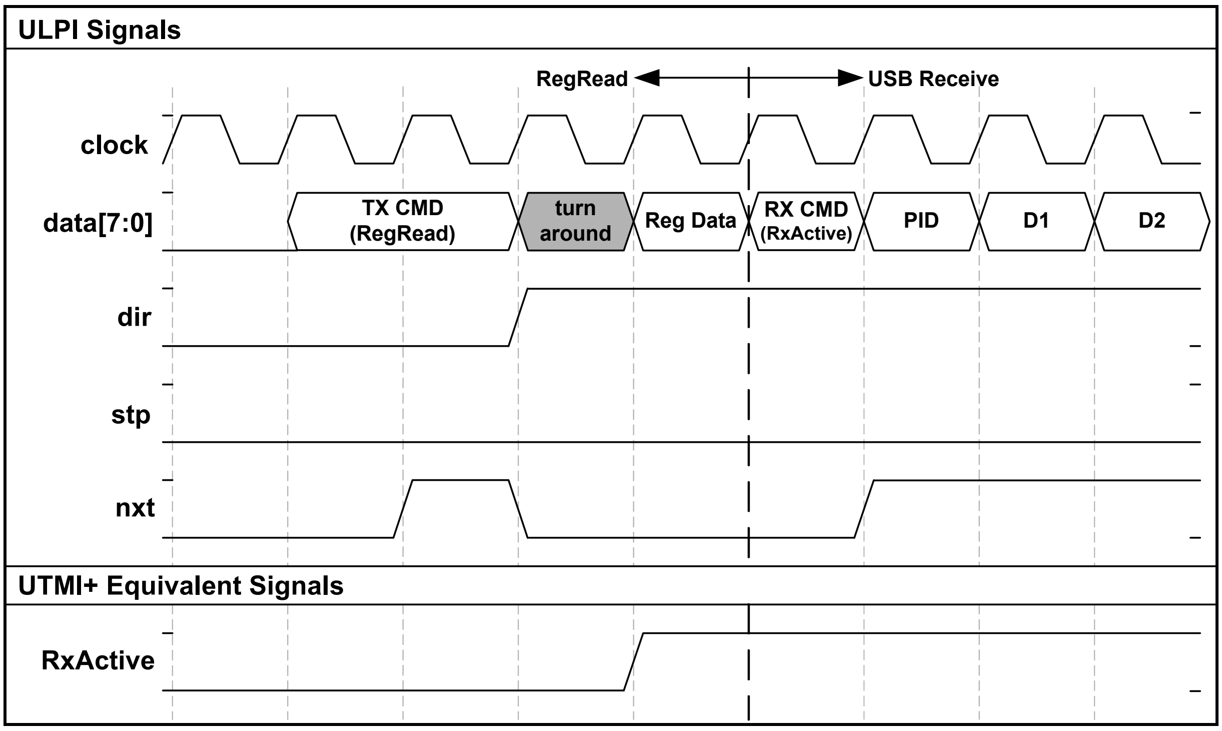

Figure 25 – USB receive in same cycle as register read data. USB receive is delayed.

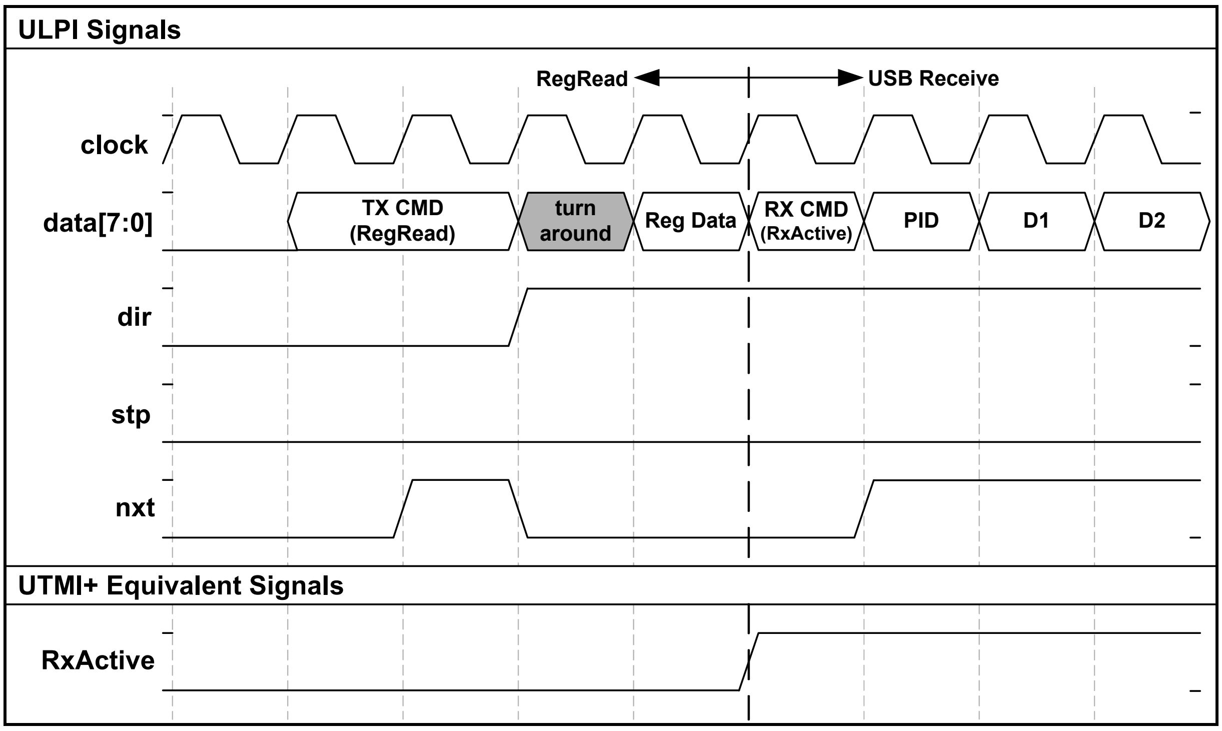

Figure 26 – Register read followed immediately by a USB receive

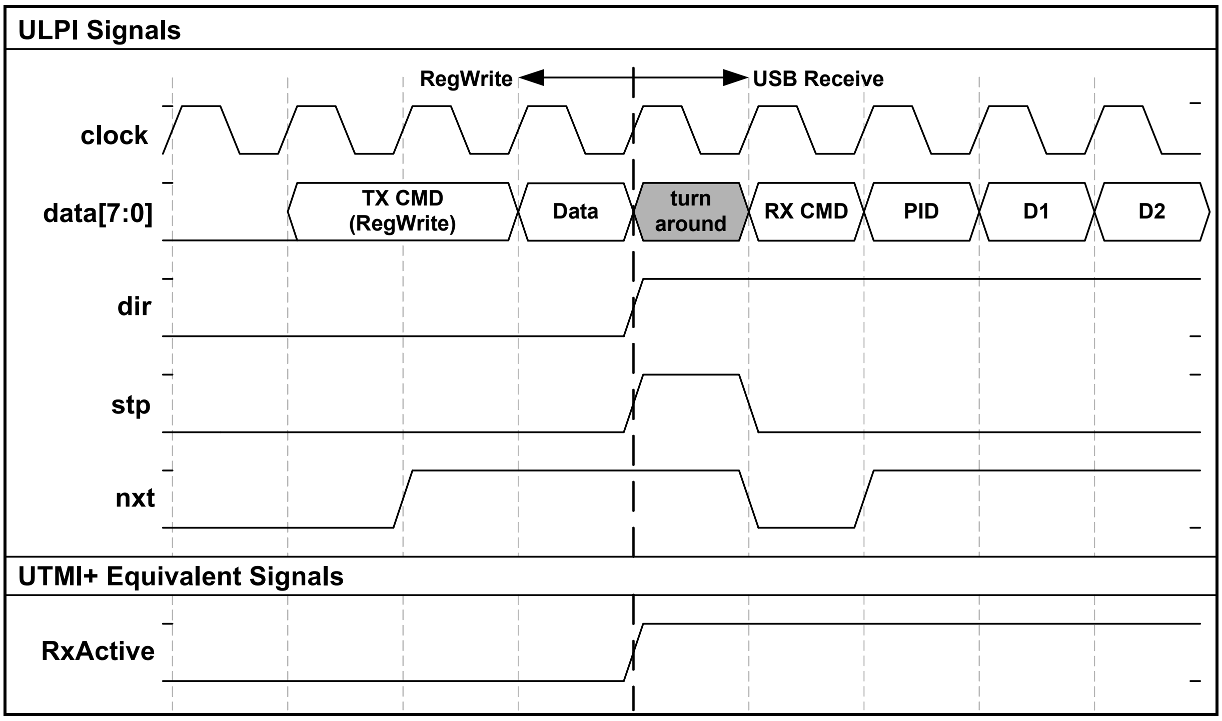

Figure 27 – Register write followed immediately by a USB receive during stp assertion

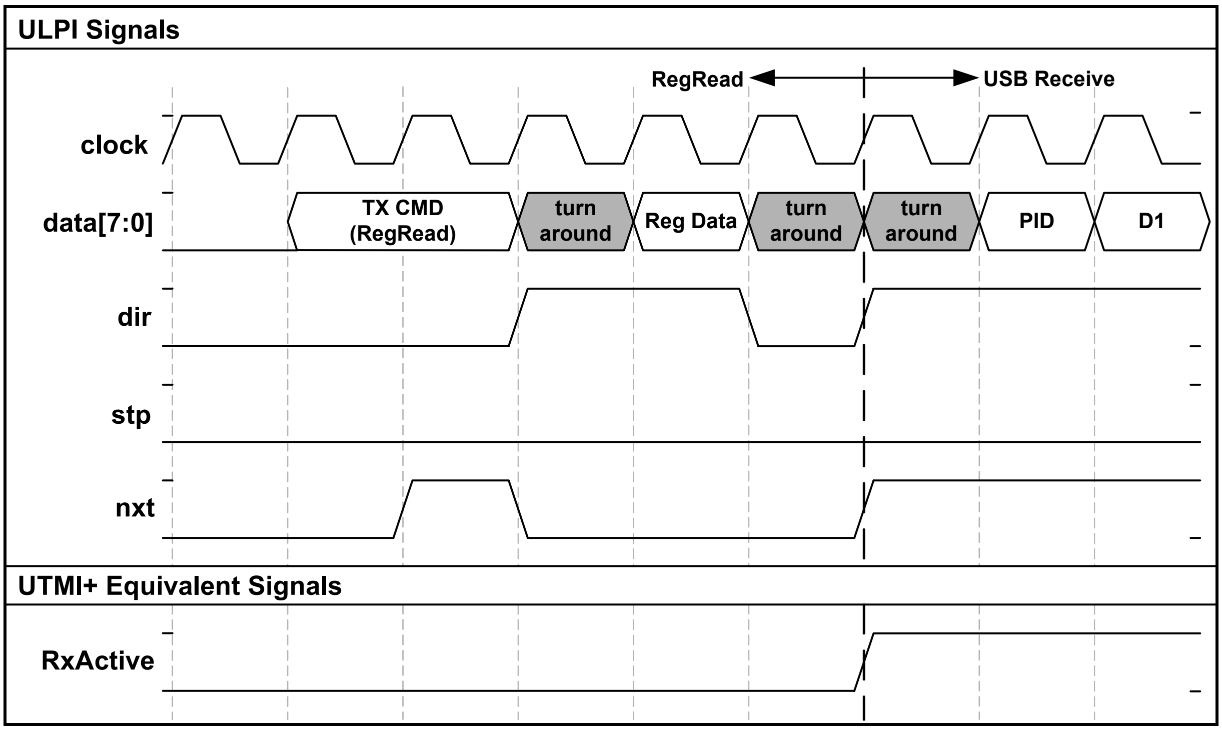

Figure 28 – Register read followed by a USB receive

3.8.3.4 Extended Register Read and Write

Access to immediate register address 2Fh indicates an access to the extended register set. In this case, the address is available in the next clock cycle.

For an extended register write, as shown in Figure 29~~Figure 29~~, the Link sends a register write command with the address set to 2Fh, and waits for **nxt** to assert. In the cycle after **nxt** asserts, the Link sends the extended register address and waits for **nxt** to assert again. When **nxt** asserts the second time, the Link sends the register write data and waits for **nxt** to assert again. When **nxt** asserts the third time, the Link asserts **stp** in the following cycle to complete the operation. If the PHY aborts the RegWrite by asserting **dir**, the Link must retry the RegWrite when the bus is idle.

For an extended register read, as shown in Figure 30~~Figure 30~~, the Link sends a register read command with the address set to 2Fh, and waits for **nxt** to assert. In the cycle after **nxt** asserts, the Link sends the extended register address and waits for **nxt** to assert again. When **nxt** asserts the second time, the PHY asserts **dir** to gain control of the data bus. In the cycle after **dir** asserts, the PHY returns the register read data. The PHY does not assert **nxt** when **dir** is asserted during the register read operation, even during the cycle that register read data is being returned. This allows a USB receive to over-ride the register read during any cycle, and is explained in 3.8.3.5. If the PHY aborts the RegRead by asserting **dir** earlier than shown in Figure 30~~Figure 30~~, the Link must retry the RegRead when the bus is idle.

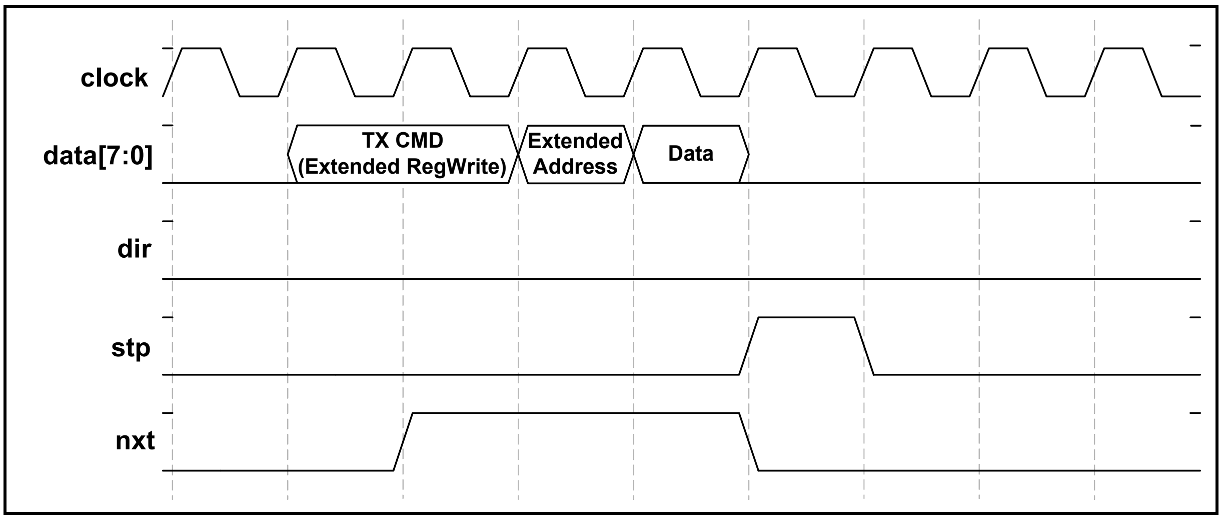

Figure 29 – Extended register write

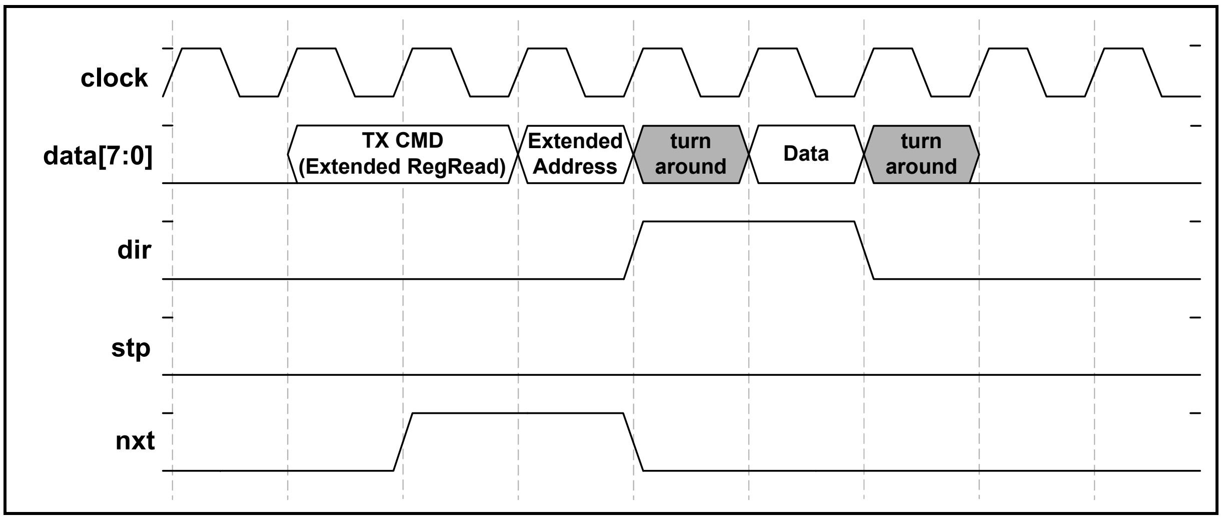

Figure 30 – Extended register read

3.8.3.5 Extended Register Read aborted by and back-to-back with USB Receive

An extended register read is identical to an immediate register read, except for the extended address byte cycle. Therefore, Figure 23~~Figure 23~~, Figure 24~~Figure 24~~, Figure 26~~Figure 26~~, and Figure 27~~Figure 27~~ also apply to extended register reads. There is one extra corner case and this is shown in Figure 31~~Figure 31~~, an extended register read aborted by a USB receive during the extended address cycle.

Extended register read and write operations are also aborted by the PHY sending an **RX CMD**, except during the cycle that register read data is being returned to the Link.

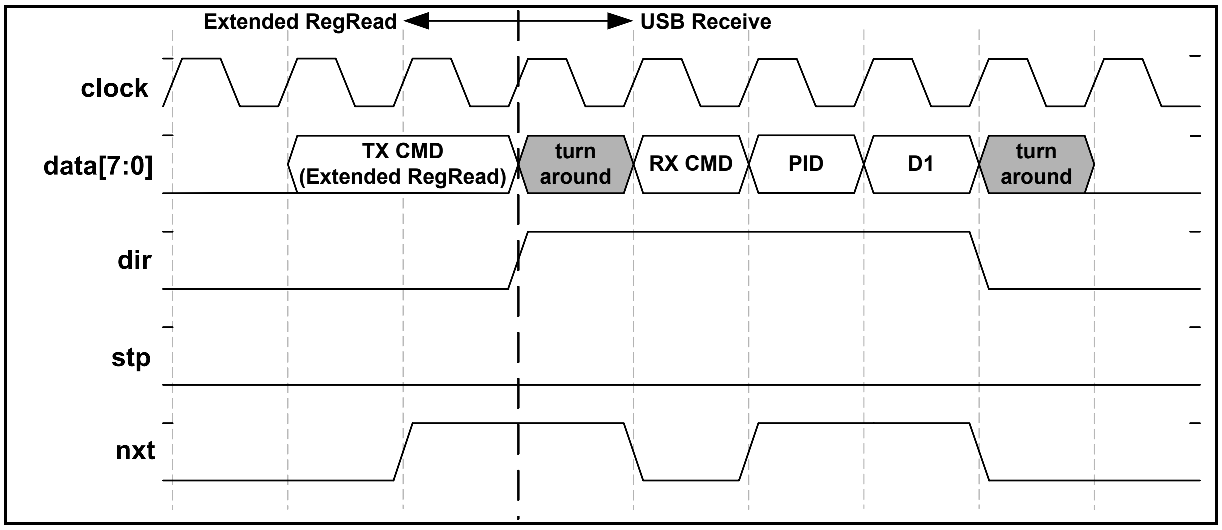

Figure 31 – Extended register read aborted by USB receive during extended address cycle

3.8.4 Aborting ULPI Transfers

3.8.4.1 Link aborted by PHY

When the Link is transferring data on the ULPI bus, the PHY can abort the Link by asserting **dir**. ULPI does not specify any conditions that will cause this to happen.

3.8.4.2 PHY aborted by Link

When the PHY has **dir** asserted in Synchronous Mode, the Link can abort the PHY by asserting **stp**, as shown in Figure 32~~Figure 32~~ and Figure 33~~Figure 33~~. In the next cycle, the PHY must de-assert **dir**, and keep **dir** de-asserted, until the Link transaction has completed. If the Link performs a register write or USB transmit as in Figure 32~~Figure 32~~, the PHY must wait for the **stp** pulse at the end of the sequence before re-asserting **dir**. If the Link performs a register read, the PHY must return the register read data as in Figure 33~~Figure 33~~, and is allowed to continue to assert **dir** if it needs to communicate USB Receive data or an **RX CMD**.

The Link cannot abort the PHY in the same cycle that **dir** is asserted, and the PHY must ignore **stp** in the same cycle that it first asserts **dir**.

If the Link asserts **stp** in the same cycle that **dir** is de-asserted, the PHY must still guarantee the Link transaction.

When the PHY is aborted, the Link must drive a **TX CMD** immediately after the turnaround cycle. If the Link does not immediately drive a **TX CMD**, the PHY is allowed to re-assert **dir**, as shown in Figure 34~~Figure 34~~.

All ULPI PHY implementations must support being aborted by the Link asserting **stp**. While this feature can be used at any time, it is provided primarily for the Link to shut down a babbling port by disabling the PHY. The PHY cannot guarantee the validity of USB data during the current packet and the next packet if the Link asserts **stp** during a USB receive packet.

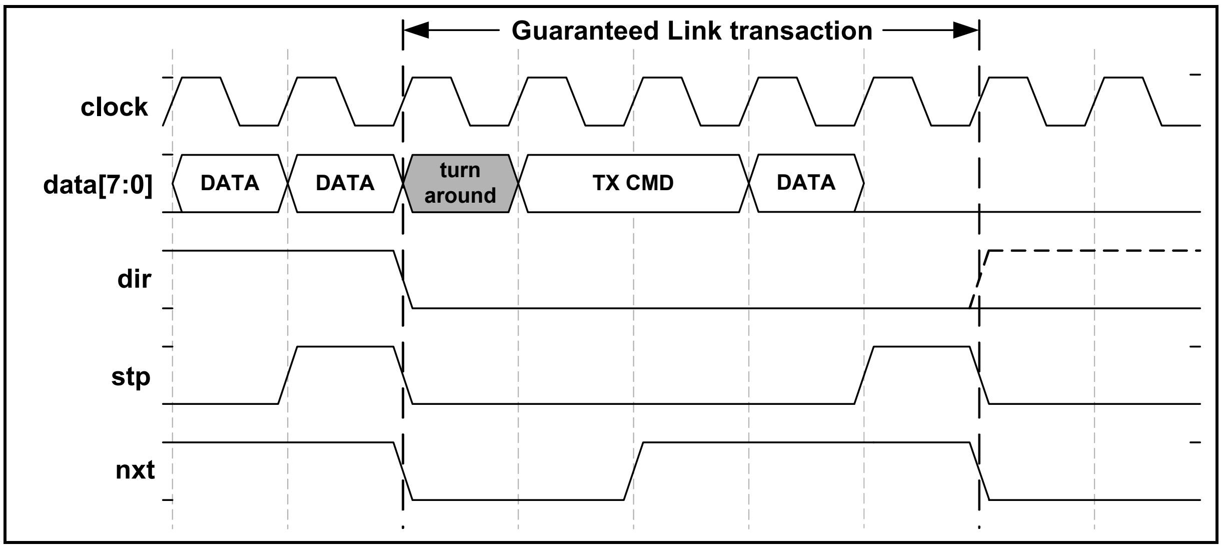

Figure 32 – PHY aborted by Link asserting stp. Link performs register write or USB transmit.

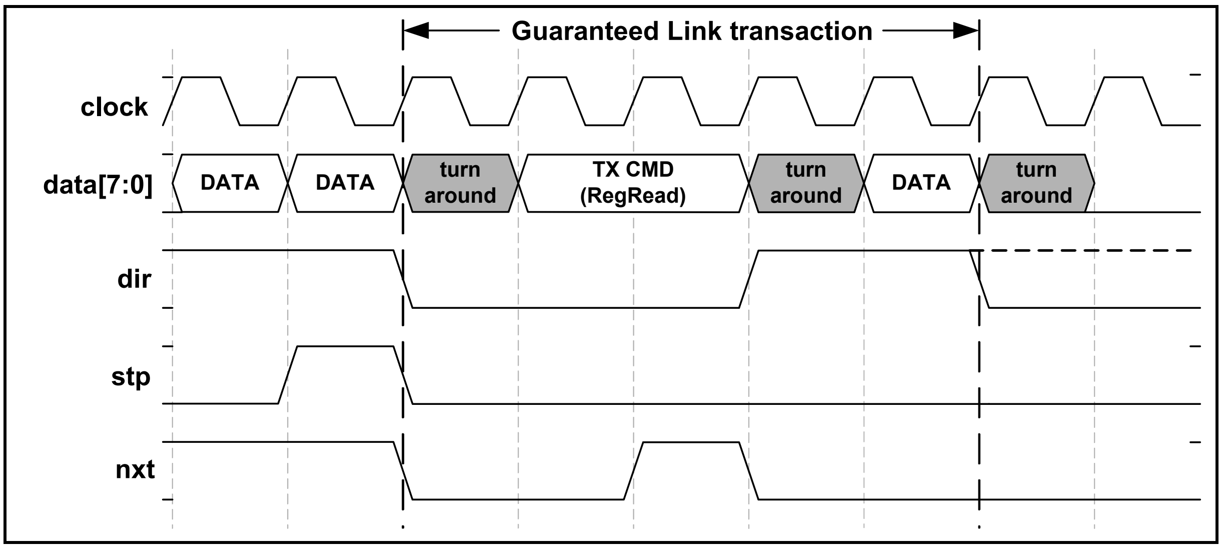

Figure 33 – PHY aborted by Link asserting stp. Link performs register read.

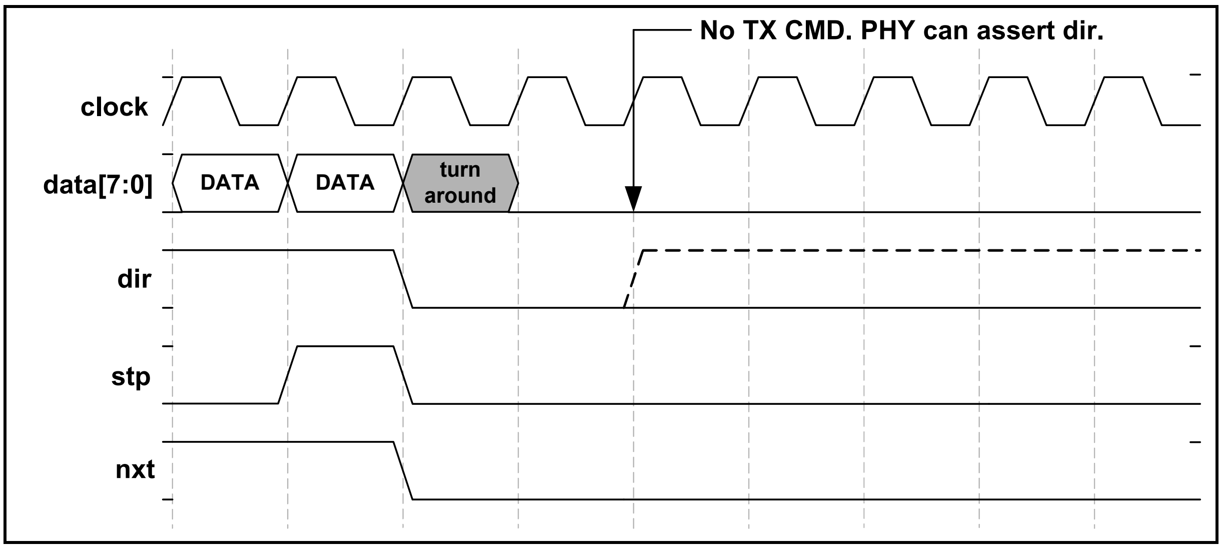

Figure 34 – Link aborts PHY. Link fails to drive a TX CMD. PHY re-asserts dir.

3.8.5 USB Operations

The following sections describe how the Link and PHY must communicate on the ULPI bus to perform specific USB operational sequences. The diagrams are compressed horizontally on the time axis and not to scale.

3.8.5.1 Hi-Speed Detection Handshake (Chirp)

Hi-Speed Detection Handshake, or Chirp, is shown in Figure 35~~Figure 35~~. Note that Figure 35~~Figure 35~~ timing is not to scale, and not all **RX CMD** updates are shown. Bus turnaround cycles are also not shown, and must occur for one cycle after every assertion and de-assertion of **dir**. The following sequence of events must be followed.

1. **FS/LS Detect** – The host detects a peripheral attachment as low speed if **D-** is high and as full speed if **D+** is high. If a host detects a low speed peripheral, it does not follow the remainder of this protocol.
2. **Host Drives** – If a host detects a full speed peripheral, it resets the peripheral by writing to the **Function Control** register and setting ***XcvrSelect*** = 00b (HS) and ***TermSelect*** = 0b which drives SE0 on the bus (**D+** and **D-** connected to ground via 45Ω). The host also sets ***OpMode*** = 10b for correct chirp transmit and receive[11]. The start of SE0 is labelled T0. The peripheral PHY asserts **dir** and informs the Link of the **LineState** change using an **RX CMD**.
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~~01b~~ (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).
4. **Host Responds** – If the host does not detect the peripheral chirp, it must continue the assertion of SE0 until the end of reset. If the host detects the peripheral chirp K for no less than 2.5us, then no more than 100us after the bus leaves the chirp K state, the host sends a **TX CMD** (NOPID) with an alternating sequence of chirp K's and J's. Each individual chirp K or J must last no less than 40us and no longer than 60us.
5. **HS Idle** – The peripheral must detect a minimum of chirp K-J-K-J-K-J. Each individual chirp K and J must be detected for at least 2.5us. After seeing that minimum sequence, the peripheral Link sets ***TermSelect*** = 0b and ***OpMode*** = 00b. The peripheral is now in Hi-Speed mode and sees !squelch (01b) on **LineState**. When the peripheral sees squelch (10b) on **LineState**, it knows the host has completed chirp and waits for Hi-Speed USB traffic to begin. After transmitting the chirp sequence, the host changes OpMode to 00b and begins sending USB packets.

[11] Receiving chirp signalling causes **LineState** to be encoded differently, and takes into account the Hi-Speed differential receiver output so as not to see false bus activity.

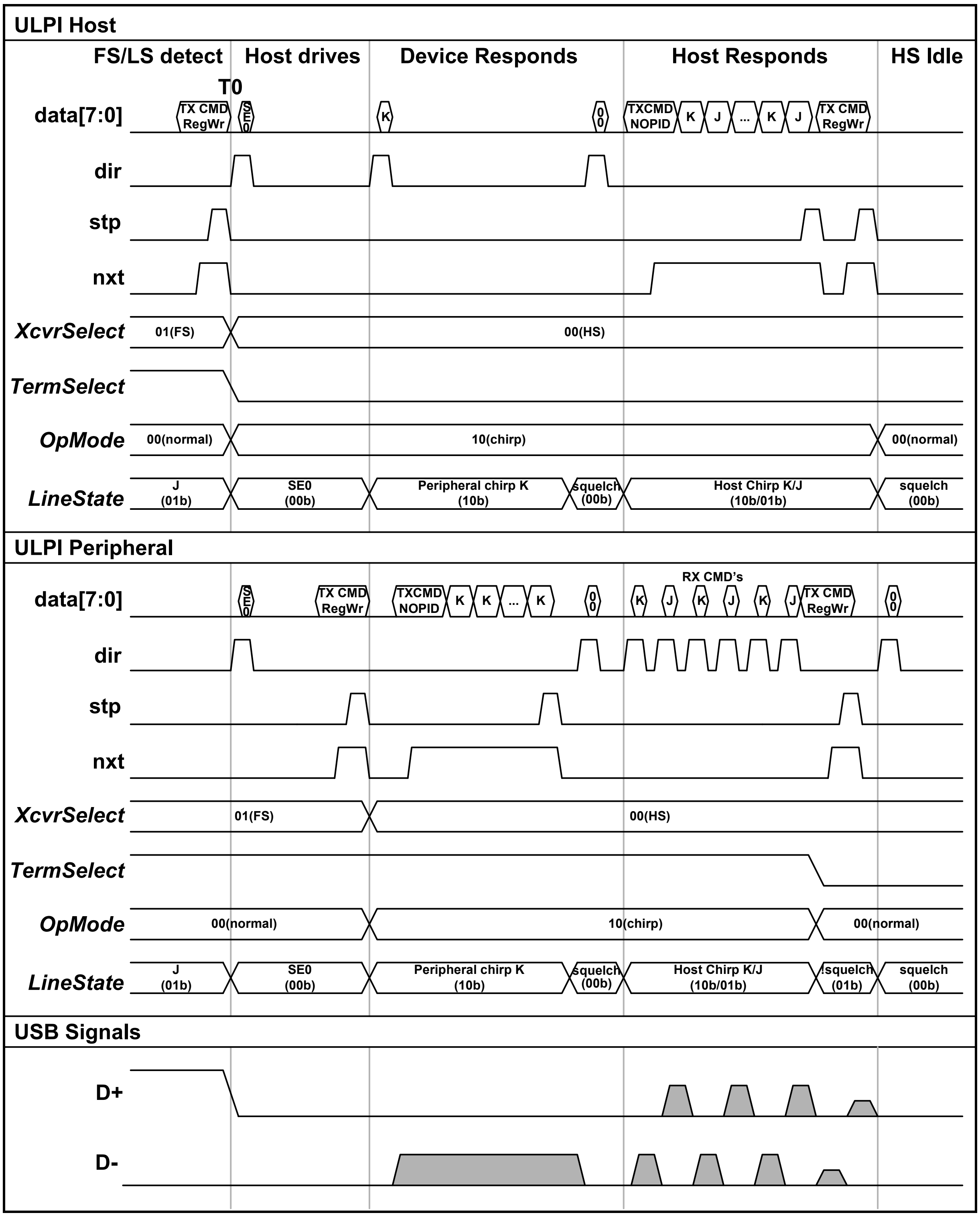

Figure 35 – Hi-Speed Detection Handshake (Chirp) sequence (timing not to scale)

3.8.5.2 Preamble

USB defines preamble packets as headers to low speed packets that must travel over a full speed bus, between host and hub. To enter preamble mode, the Link sets ***XcvrSelect*** = 11b in the **Function Control** register. When in preamble mode, the PHY operates identically to full speed mode, and sends all data with full speed rise and fall times. Whenever the Link transmits a USB packet in preamble mode, the PHY must automatically send a preamble header at full speed bitrate before sending the Link packet at low speed bitrate. The PHY must ensure a minimum gap of 4 full speed bit times between the last bit of the full speed PRE PID and the first bit of the low speed packet SYNC. The PHY must drive a J for at least 1 FS bit time after sending the PRE PID, after which the pull-up resistor can hold the J state on the bus.

In preamble mode, the PHY can also receive low speed packets from the full speed bus.

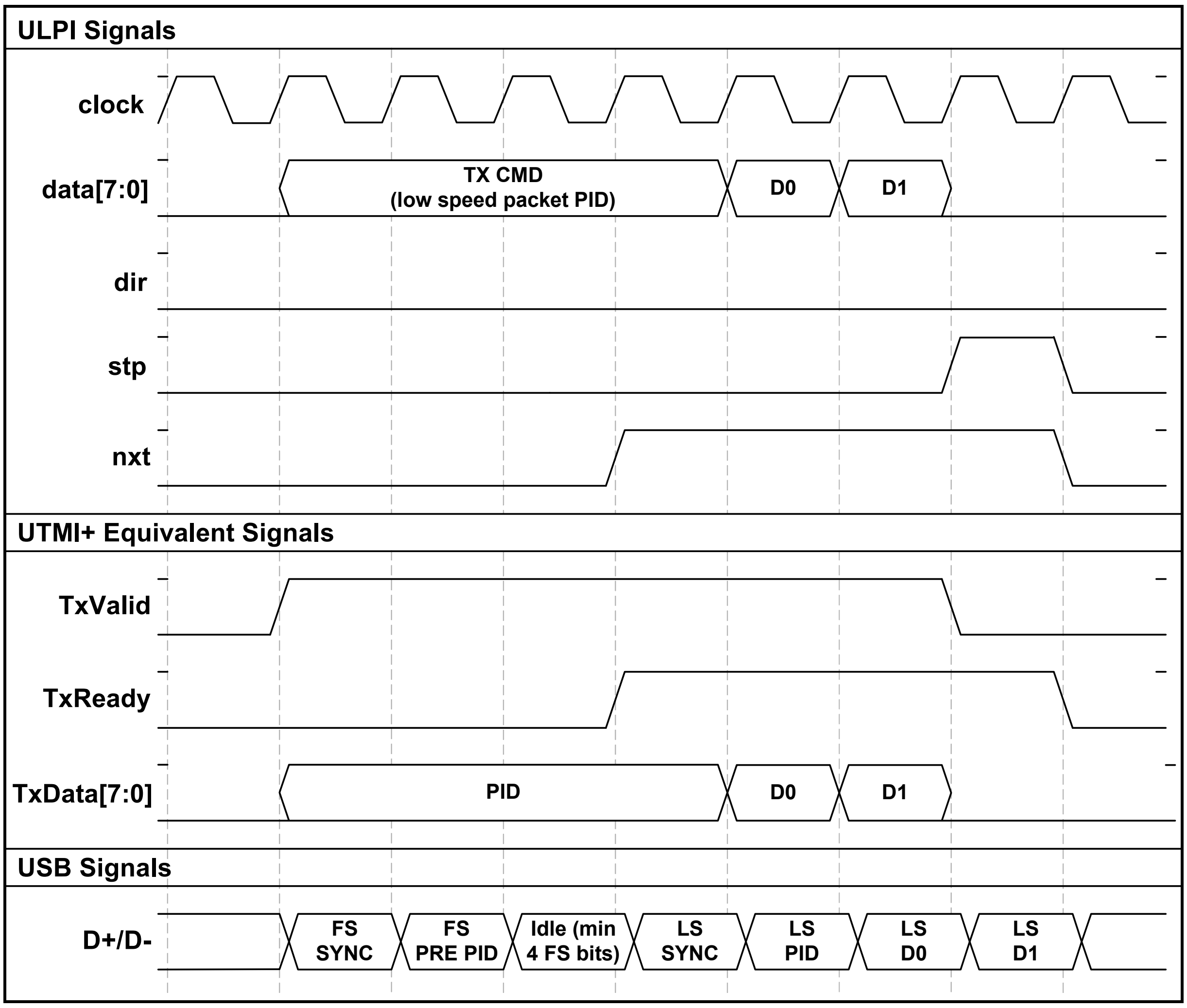

Figure 36 – Preamble sequence (D+/D- timing not to scale)

3.8.5.3 USB Suspend and Resume

This section deals with suspend and resume initiated by a host or hub.

3.8.5.3.1 Low Speed Suspend and Resume

Figure 37~~Figure 37~~ illustrates how a host or hub enters Low Speed suspend and then initiates resume signalling to wake up the downstream Low Speed peripheral. Note that Figure 37~~Figure 37~~ timing is not to scale, and does not show all **RX CMD Linestate** updates. Bus turnaround cycles are also not shown, and must occur for one cycle after every assertion and de-assertion of **dir**. The following text describes the sequence of events.

1. **LS Traffic** – Initially, the host and peripheral are sending low speed traffic over the USB bus (***XcvrSelect*** set to 10b). The host has its 15kΩ pull-downs enabled (***DpPulldown*** and ***DmPulldown*** set to 1b) and the 45Ω terminations disabled (***TermSelect*** set to 1b). The peripheral has the 1.5kΩ pull-up connected to **D-** (***TermSelect*** set to 1b).
2. **LS Suspend** – When the peripheral sees no bus activity for 3ms, it enters the suspend state. The peripheral Link places the PHY into Low Power Mode by setting the ***SuspendM*** bit as described in 3.9, causing the PHY to draw only suspend current. The host may or may not be powered down.
3. **Resume K** – When the host wants to wake up the peripheral, it sets ***OpMode*** to 10b and transmits a LS K for at least 20ms. The peripheral Link sees the Resume K (01b) on **LineState**, and asserts **stp** to wake up the PHY as described in 3.9.
4. **EOP** – The host PHY automatically appends a LS EOP (2 bits LS SE0 followed by 1 bit LS J) when **stp** is asserted. The host PHY knows to add the LS EOP because ***DpPulldown*** and ***DmPulldown*** are set to 1b for a host. After the LS EOP has completed, the host Link sets ***OpMode*** to 00b for normal LS operation. The peripheral Link sees the LS EOP and also resumes normal LS operation.

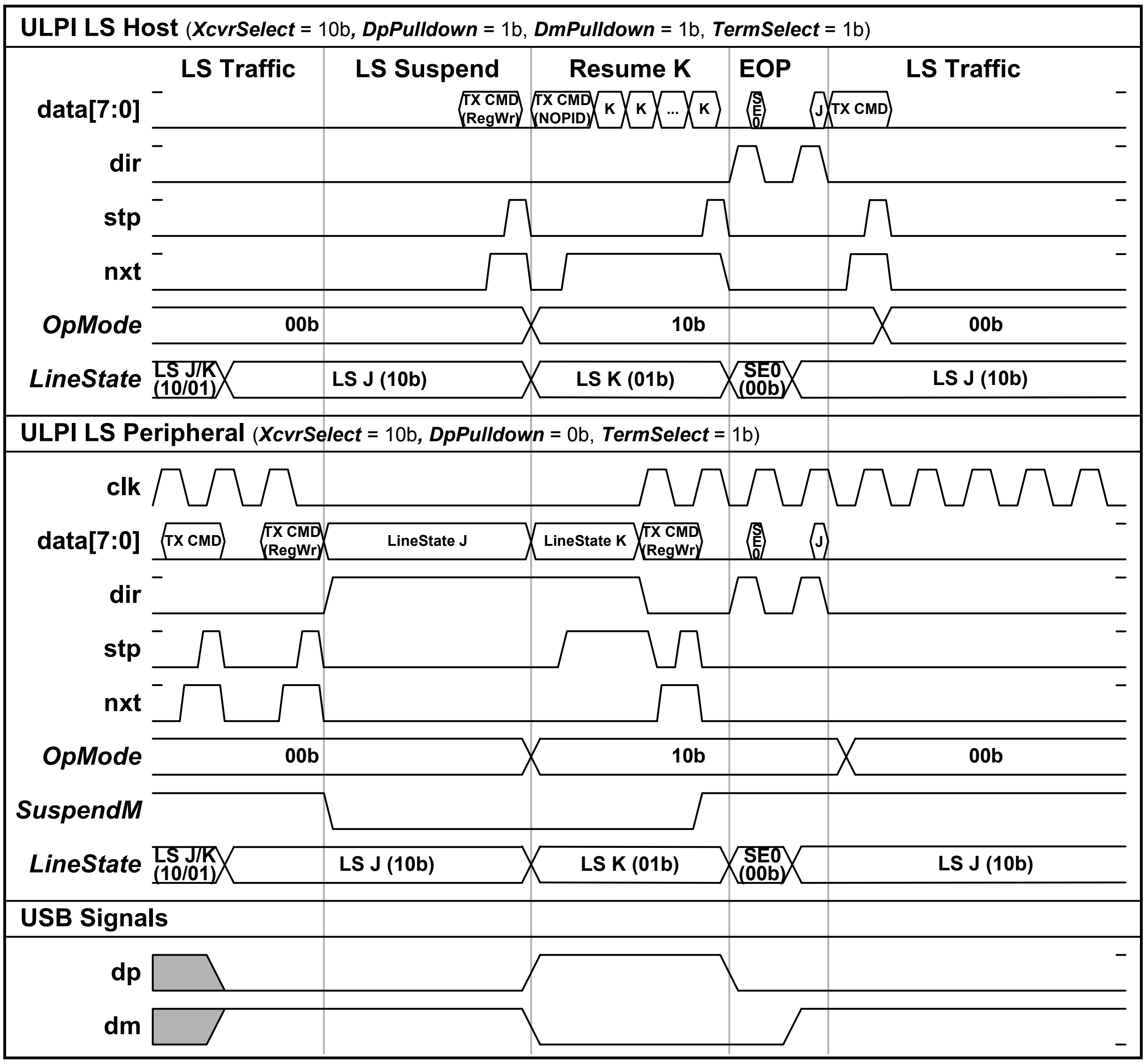

Figure 37 – LS Suspend and Resume (timing not to scale)

3.8.5.3.2 Full Speed Suspend and Resume

Figure 38 illustrates how a host or hub enters Full Speed suspend and then initiates resume signalling to wake up the downstream Full Speed peripheral. Note that Figure 38 timing is not to scale, and does not show all **RX CMD Linestate** updates. Bus turnaround cycles are also not shown, and must occur for one cycle after every assertion and de-assertion of **dir**. The following text describes the sequence of events.

1. **FS Traffic** – Initially, the host and peripheral are sending full speed traffic over the USB bus (***XcvrSelect*** set to 01b). The host has its 15kΩ pull-downs enabled (***DpPulldown*** and ***DmPulldown*** set to 1b) and the 45Ω terminations disabled (***TermSelect*** set to 1b). The peripheral has the 1.5kΩ pull-up connected to **D+** (***TermSelect*** set to 1b).
2. **FS Suspend** – When the peripheral sees no bus activity for 3ms, it enters the suspend state. The peripheral Link places the PHY into Low Power Mode by setting the ***SuspendM*** bit as described in 3.9, causing the PHY to draw only suspend current. The host may or may not be powered down.
3. **Resume K** – When the host wants to wake up the peripheral, it sets ***OpMode*** to 10b and transmits a FS K for at least 20ms. The peripheral Link sees the Resume K (10b) on **LineState**, and asserts **stp** to wake up the PHY as described in 3.9.
4. **EOP** – The host PHY automatically appends a LS EOP (2 bits LS SE0 followed by 1 bit FS J) when **stp** is asserted. The host PHY knows to add the LS EOP because ***DpPulldown*** and ***DmPulldown*** are set to 1b for a host. After the LS EOP has completed, the host Link sets ***OpMode*** to 00b for normal FS operation. The peripheral Link sees the LS EOP and also resumes normal FS operation.

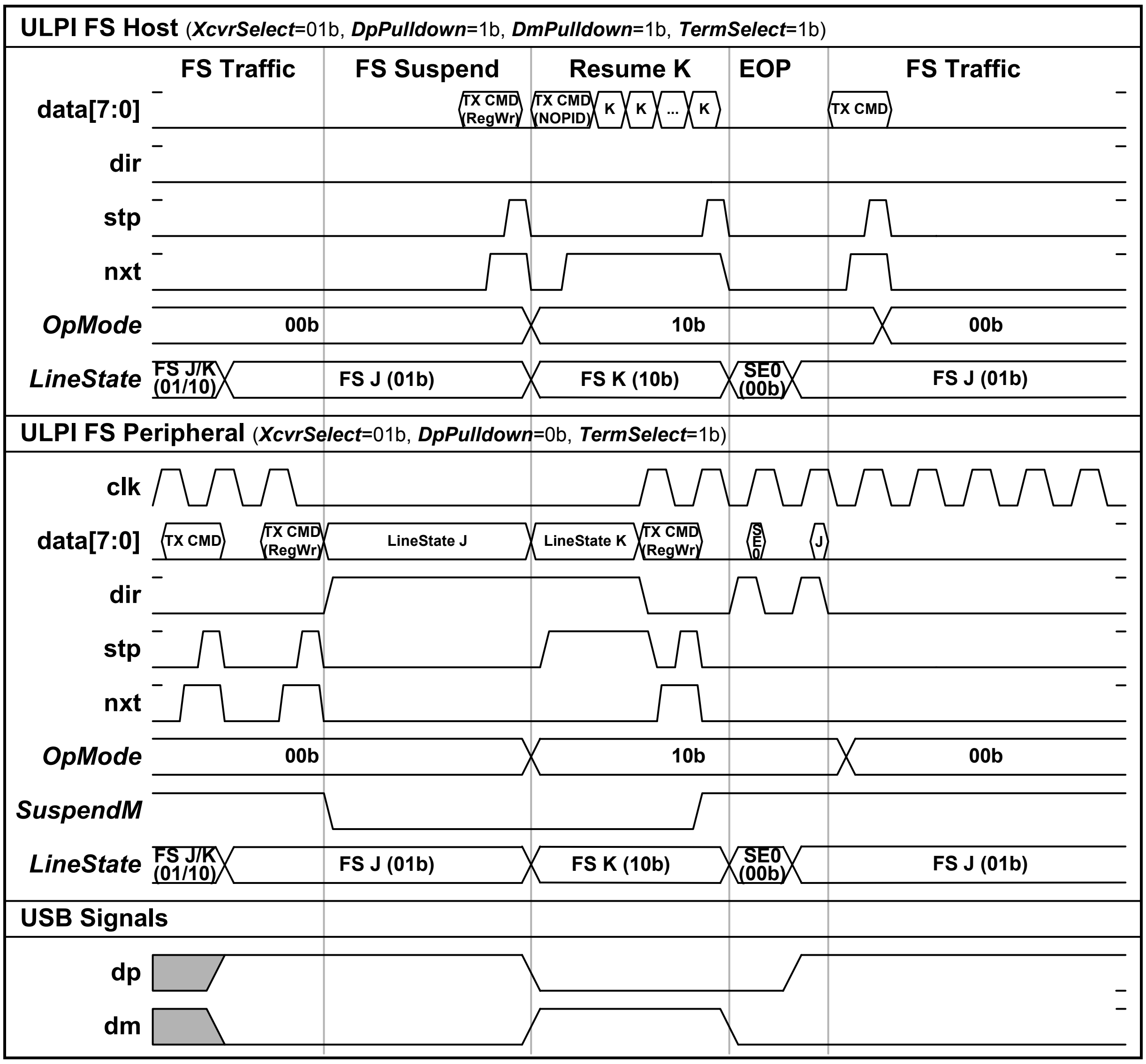

Figure 38 – FS Suspend and Resume (timing not to scale)

3.8.5.3.3 Hi-Speed Suspend and Resume

Figure 39 illustrates how a Hi-Speed host or hub enters Full Speed suspend and then initiates resume signalling to wake up the downstream Hi-Speed peripheral. Note that Figure 39 timing is not to scale, and does not show all **RX CMD Linestate** updates. Bus turnaround cycles are also not shown, and must occur for one cycle after every assertion and de-assertion of **dir**. The following text describes the sequence of events.

1. **HS Traffic** – Initially, the host and peripheral are sending Hi-Speed traffic over the USB bus (***XcvrSelect*** set to 00b). The host has its 15kΩ pull-downs enabled (***DpPulldown*** and ***DmPulldown*** set to 1b) and the 45Ω terminations enabled (***TermSelect*** set to 0b). The peripheral has its 45Ω terminations enabled (***TermSelect*** set to 0b).
2. **FS Suspend** – When the peripheral sees no bus activity for 3ms, it enters the suspend state. The peripheral Link places the PHY into Full Speed mode (***XcvrSelect*** set to 01b), removes the 45Ω terminations and enables the 1.5kΩ pull-up on **D+** (***TermSelect*** set to 1b). The peripheral Link then places the PHY into Low Power Mode by setting ***SuspendM*** as described in 3.9, causing the PHY to draw only suspend current. The host also changes to Full Speed (***XcvrSelect*** set to 01b), removes the 45Ω terminations (***TermSelect*** set to 1b), and then may or may not be powered down.
3. **Resume K** – When the host wants to wake up the peripheral, it sets ***OpMode*** to 10b and transmits a FS K for at least 20ms. The peripheral Link sees the Resume K (10b) on **LineState**, and asserts **stp** to wake up the PHY as described in 3.9.
4. **HS Traffic** – The host Link sets Hi-Speed (***XcvrSelect*** set to 00b), and enables its 45Ω terminations (***TermSelect*** set to 0b). The peripheral Link sees SE0 on the USB bus and also sets Hi-Speed (***XcvrSelect*** set to 00b), and enables its 45Ω terminations (***TermSelect*** set to 0b). The host Link sets ***OpMode*** to 00b for normal HS operation.

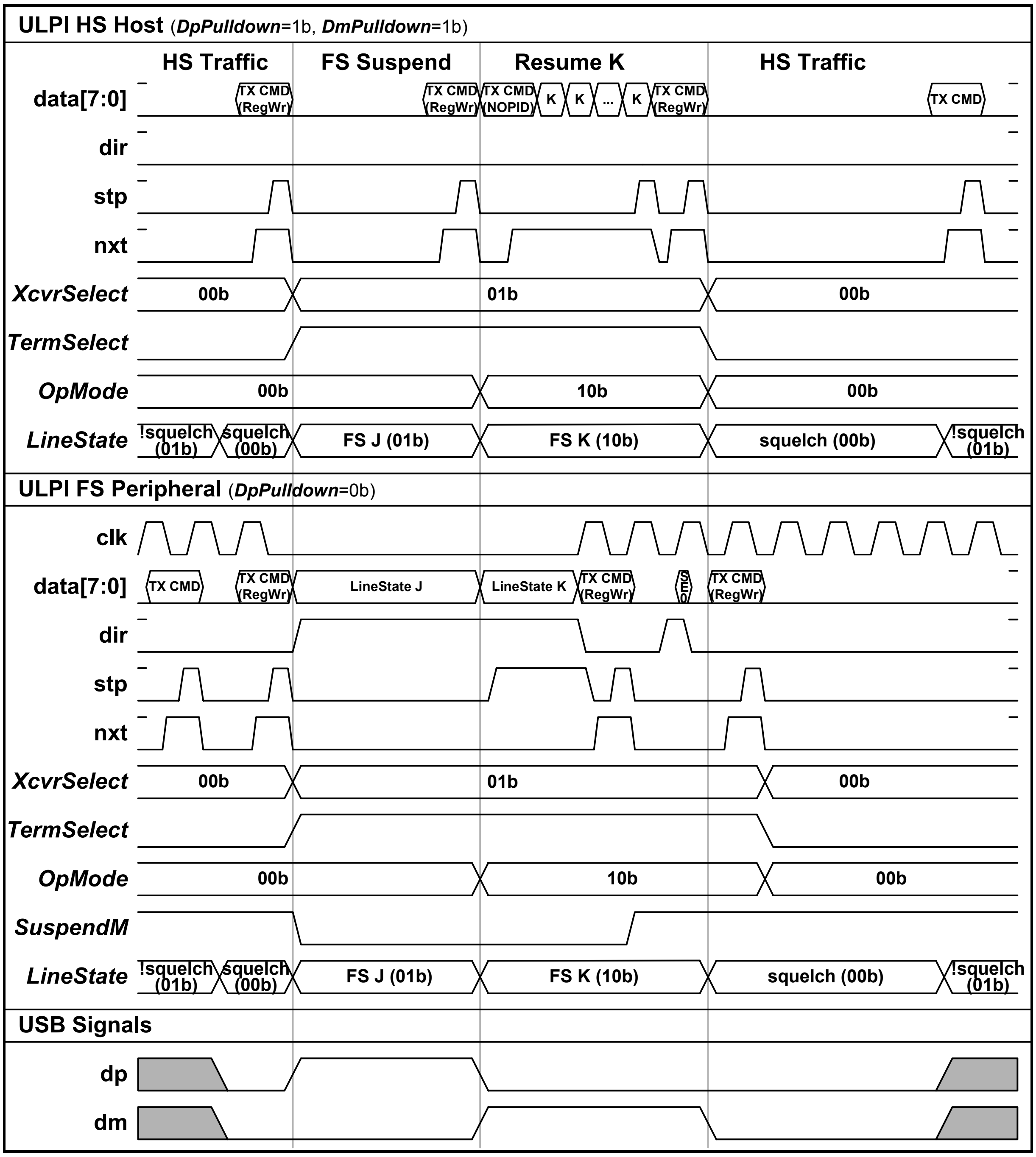

Figure 39 – HS Suspend and Resume (timing not to scale)

3.8.5.4 Remote Wake-up

This section deals with a peripheral initiating a remote wake-up (resume). When placed into USB suspend, the Link remembers what speed it was originally operating in. Depending on the original speed, the Link follows one of the protocols detailed below. Note that in the following figures, timing is not to scale, and not all **RX CMD Linestate** updates are shown. Bus turnaround cycles are also not shown, and must occur for one cycle after every assertion and de-assertion of **dir**.

3.8.5.4.1 Low Speed Remote Wake-up

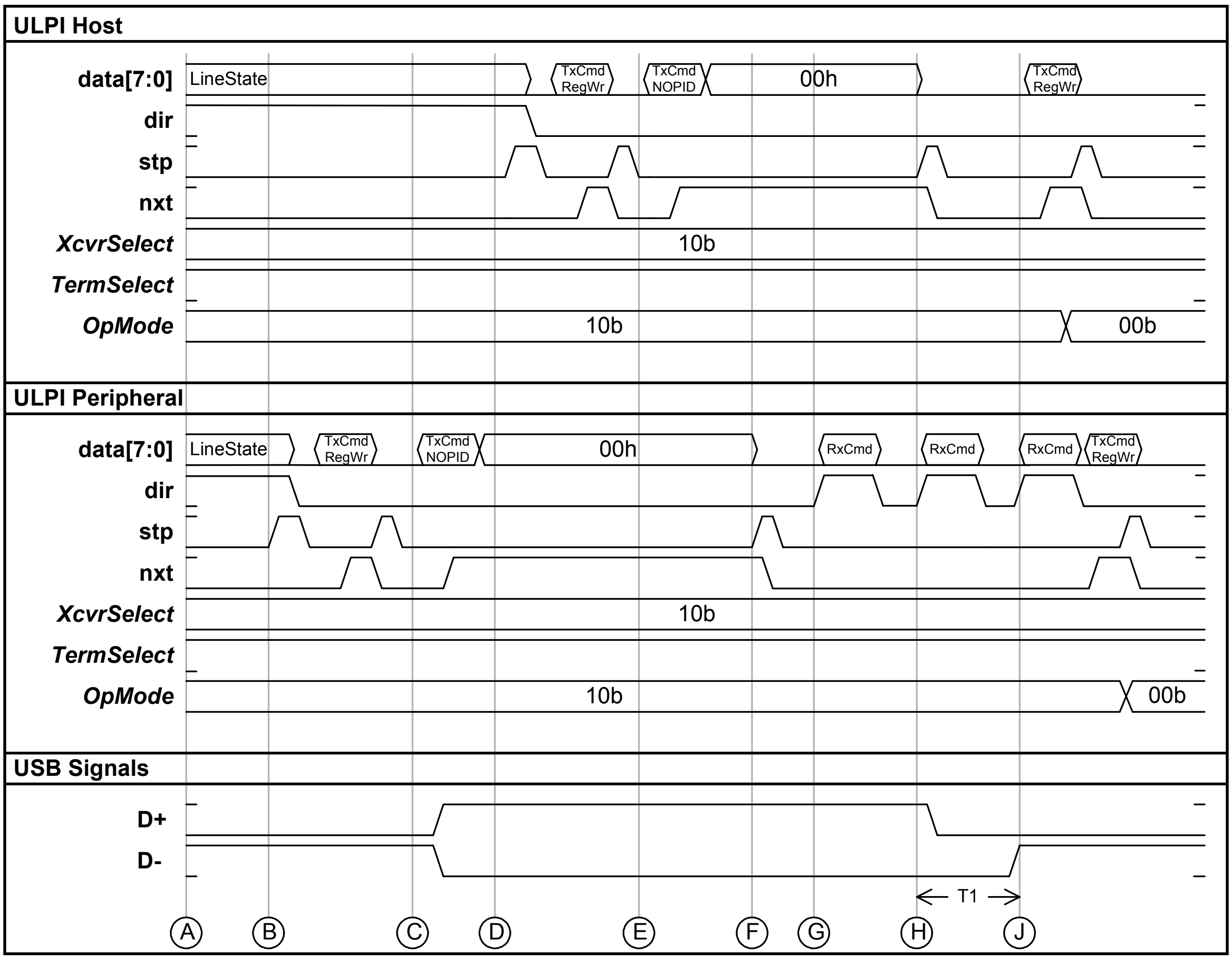

Figure 40 – Low Speed Remote Wake-Up from Low Power Mode (timing not to scale)

A) Both Host and Peripheral start out in Low Power Mode.
B) The Peripheral begins Remote Wake-Up by re-enabling its clock and setting its ***SuspendM*** bit.
C) The Peripheral begins driving K on the bus to signal Resume. Note that the peripheral Link should assume that ***LineState*** is K (01b) while transmitting.
D) The Host recognizes the resume, re-enables its clock and sets its ***SuspendM*** bit.
E) The Host takes over driving the resume within 1ms of detecting the remote wake-up[12].
F) The Peripheral stops driving resume.
G) The Peripheral sees the Host continuing to drive the resume.
H) The Host stops driving Resume and the PHY adds the LS EOP to the end of Low Speed Resume. The Peripheral recognizes the LS EOP as the end of Resume. (T1 is the LS EOP interval. Refer to the UTMI+ specification [Ref 4] for more details).
J) Both Host and Peripheral revert to normal operation by writing ***OpMode*** to normal.

[12] If the clock cannot be restarted in less than 1ms, the PHY must implement the ***AutoResume*** feature described in 3.8.5.4.4.

3.8.5.4.2 Full Speed Remote Wake-up

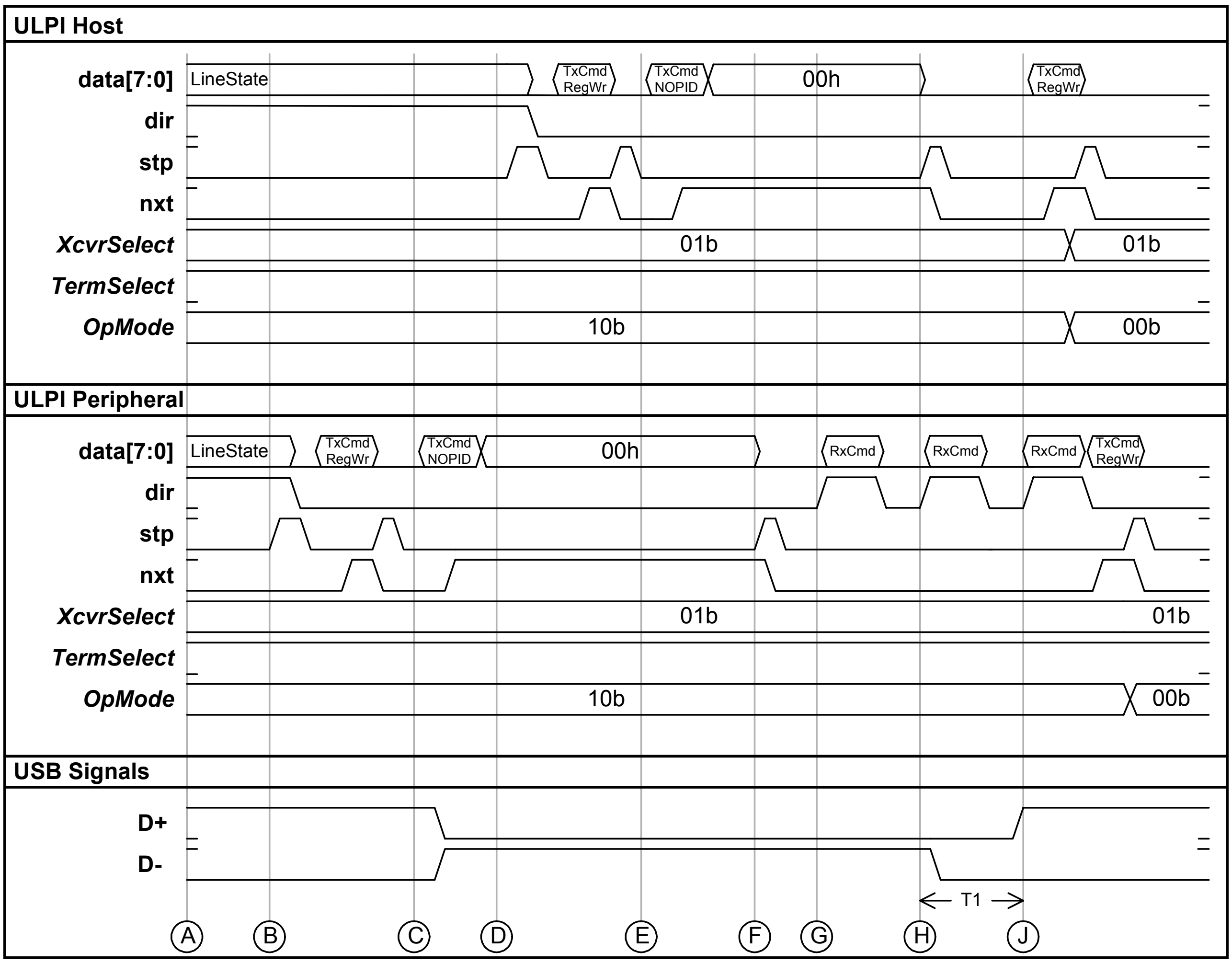

Figure 41 – Full Speed Remote Wake-Up from Low Power Mode (timing not to scale)

A) Both Host and Peripheral start out in Low Power Mode.
B) The Peripheral begins Remote Wake-Up by re-enabling its clock and setting its ***SuspendM*** bit.
C) The Peripheral begins driving K on the bus to signal Resume. Note that the peripheral Link should assume that **LineState** is K (10b) while transmitting.
D) The Host recognizes the resume, re-enables its clock and sets its ***SuspendM*** bit.
E) The Host takes over driving the resume within 1ms of detecting the remote wake-up[12~~11~~].
F) The Peripheral stops driving resume.
G) The Peripheral sees the Host continuing to drive the resume.
H) The Host stops driving Resume and the PHY adds the LS EOP to the end of Full Speed Resume. The Peripheral recognizes the LS EOP as the end of Resume. (T1 is the LS EOP interval. Refer to the UTMI+ specification [Ref 4] for more details)
J) Both Host and Peripheral revert to normal operation by writing ***OpMode*** and ***XcvrSelect***.

3.8.5.4.3 Hi-Speed Remote Wake-up

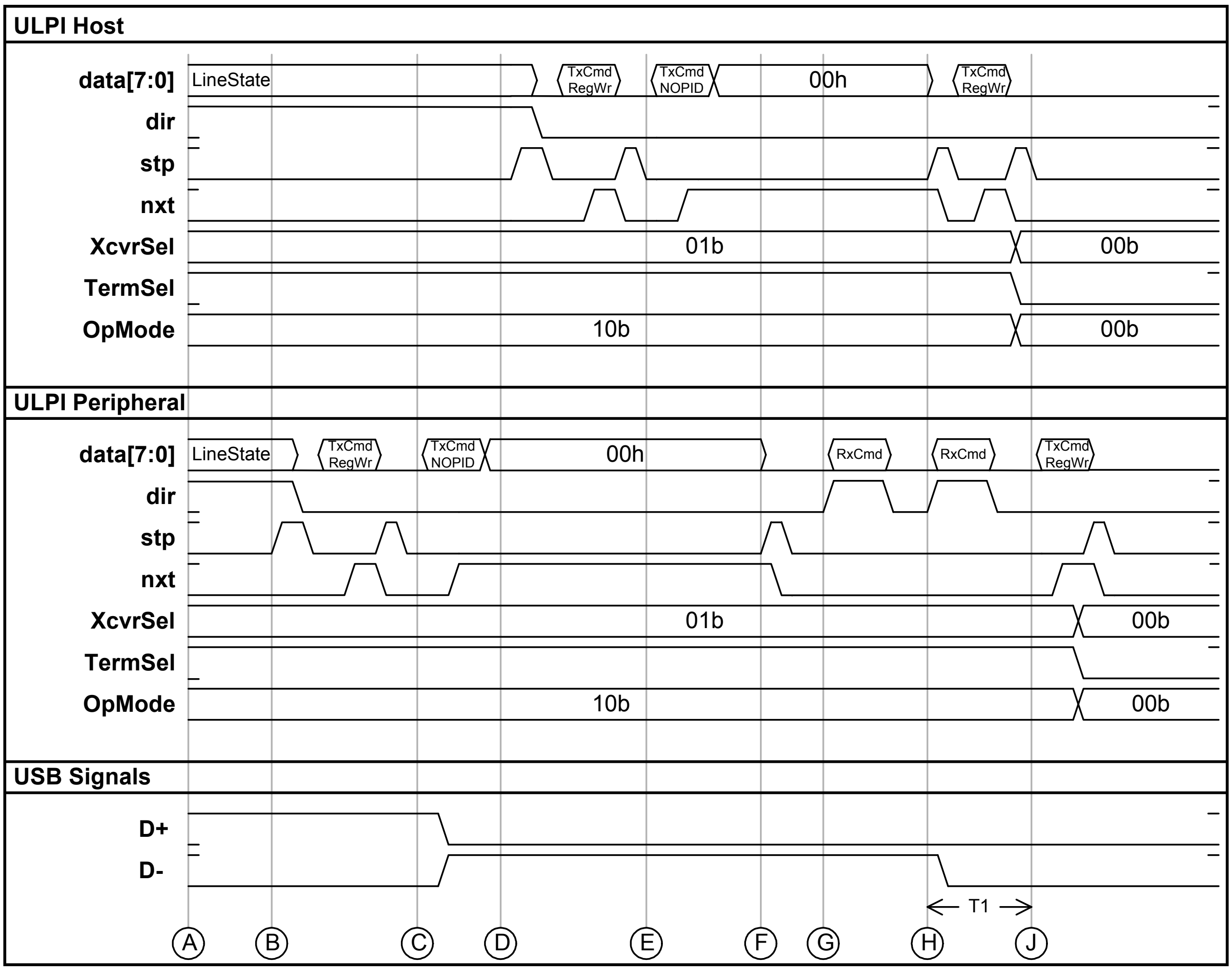

Figure 42 – Hi-Speed Remote Wake-Up from Low Power Mode (timing not to scale)

A) Both Host and Peripheral start out in Low Power Mode.
B) The Peripheral begins Remote Wake-Up by re-enabling its clock and setting its ***SuspendM*** bit (1).
C) The Peripheral begins driving K on the bus to signal Resume. Note that the peripheral Link should assume that **LineState** is K (10b) while transmitting.
D) The Host recognizes the resume, re-enables its clock and sets its ***SuspendM*** bit (2).
E) The Host takes over driving the resume within 1ms of detecting the remote wake-up[12~~11~~].
F) The Peripheral stops driving resume.
G) The Peripheral sees the Host continuing to drive the resume.
H) The Host stops driving Resume and the Bus returns to Hi-Speed Idle. The Peripheral recognizes the Hi-Speed Idle as the end of Resume. (T1 is the LS EOP interval. Refer to the UTMI+ specification [Ref 4] for more details)
J) Both Host and Peripheral revert to normal operation by writing ***OpMode***, ***XcvrSelect*** and ***TermSelect***.

3.8.5.4.4 AutoResume

When a USB host detects remote wake-up signaling (resume-K) from a downstream peripheral or hub, the host must take over driving of the resume-K signaling within 1ms. This is described in detail in the USB specification [Ref 1], sections 7.1.7.7 and 7.9.

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. If the clock cannot be restarted within 1ms, the PHY must provide an automatic resume feature. As shown in Figure 43~~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, t~~T~~he 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. ~~when the clock is restored and~~ ~~***SuspendM***~~ ~~is set to 1b.~~ Implementation details are left to the PHY vendor, however the PHY vendor must specify the clock wake-up time, T$_{\text{START\_HOST}}$~~PREP~~, as defined in Table 4~~Table 4~~.

If the clock can be restarted within 1ms, the PHY does not need to provide the automatic resume feature. Figure 40~~Figure 40~~, Figure 41~~Figure 41~~, Figure 42~~Figure 42~~ can be applied without modification.

The ***AutoResume*** bit in the **Interface Control** register controls the automatic resume feature.

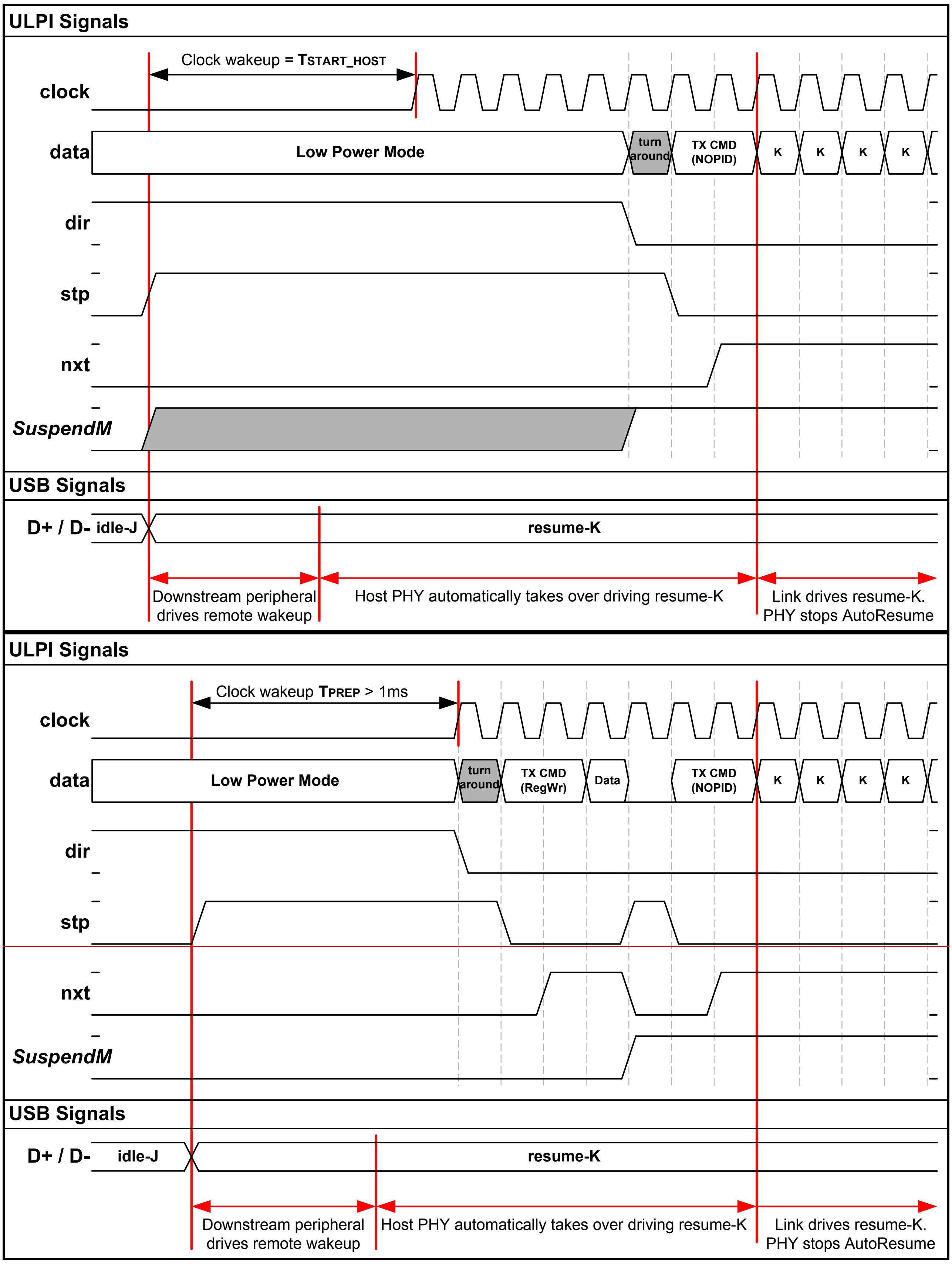
ULPI Signals
Clock wakeup = TSTART_HOST
clock
data
Low Power Mode
turn around
TX CMD (NOPID)
K
K
K
K
dir
stp
nxt
SuspendM
USB Signals
D+ / D-
idle-J
resume-K
Downstream peripheral drives remote wakeup
Host PHY automatically takes over driving resume-K
Link drives resume-K. PHY stops AutoResume
ULPI Signals
Clock wakeup TPREP > 1ms
clock
data
Low Power Mode
turn around
TX CMD (RegWr)
Data
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

Figure 43 – Automatic resume signalling (timing not to scale)

3.8.5.5 Peripheral Connect and Disconnect Detection

The ***HostDisconnect*** bit in the **USB Interrupt Status** register indicates when a peripheral is connected or disconnected, and is valid only when the PHY is used as a host (***DpPulldown*** and ***DmPulldown*** both set to 1b). When used as a host PHY, a change in an unmasked ***HostDisconnect*** causes the PHY to generate an interrupt event notification as described in section 3.6.

When used as a peripheral (***DpPulldown*** set to 0b), the PHY must never generate an interrupt event notification indicating a ***Hostdisconnect*** event.

The reader is referred to the UTMI+ specification [Ref 4] for a discussion on how to generate the ***HostDisconnect*** signal.

3.8.5.6 No SYNC and EOP Generation (OpMode 11b) (Optional)

This mode affects how packets are transmitted, and must be used for Hi-Speed only. When ***OpMode*** is set to 11b, the PHY will not automatically add the SYNC and EOP patterns when transmitting a packet. The PHY must still NRZI encode the data and perform bit-stuffing. When the Link wants to transmit a USB packet, it must send a **TX CMD** of the **NOPID** type. After the **NOPID** command, the Link sends a 4-byte SYNC pattern on the ULPI bus (00h, 00h, 00h, 80h), followed by the PID, data payload, and finally a 1-byte EOP (FEh), as shown in Figure 44~~Figure 44~~. The **TxBitstuffEnable** signal of UTMI+ is not supported. The PHY automatically enables bit-stuffing when a packet is started, and disables bit-stuffing when **stp** is asserted. The Link asserts **stp** in the same cycle it drives the EOP byte. If data is set to 00h when **stp** is asserted, the PHY will not transmit any EOP on the USB bus. The PHY must detect if the PID byte is A5h (SOF packet) and automatically send a long EOP when **stp** is asserted. For transmitting chirp and resume signalling, the Link must set ***OpMode*** to 10b.

Sending a **TX CMD** of the **PID** type when ***OpMode*** is set to 11b will result in undefined behaviour. Receiving **RX CMD** and USB receive packets when ***OpMode*** is 11b operates identically to ***OpMode*** 00b.

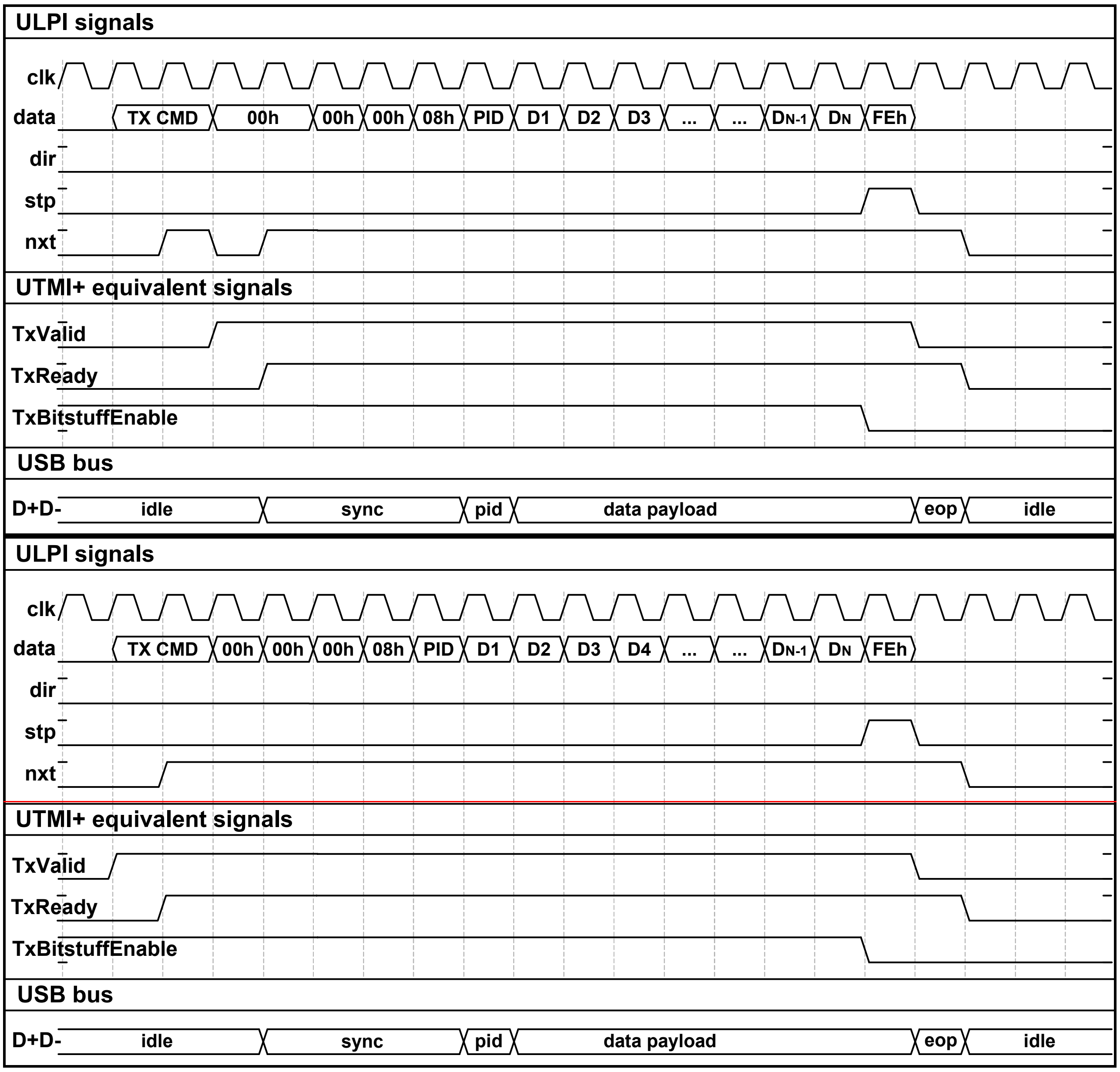

Figure 44 – USB packet transmit when OpMode is set to 11b.

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 11.

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 11 – OTG Control Register power control bits

3.8.7 OTG Operations

~~3.8.6.1~~3.8.7.1 Session Request Protocol (SRP)

ULPI provides full SRP support. The Link uses the ***ChrgVbus*** and ***DischrgVbus*** bits in the **OTG Control** register to begin and end a session.

~~3.8.6.2~~3.8.7.2 Host Negotiation Protocol (HNP) (Optional)

ULPI does not define HNP support. It is assumed that HNP is implemented in Link hardware and/or in software. This does not preclude PHY implementers from adding HNP support to any ULPI PHY so long as it does not interfere with or violate the ULPI protocol defined in this specification.

~~3.8.6.3~~3.8.7.3 VBUS Comparator Thresholds

While the OTG specification [Ref 2] provides for separate A-device and B-device session valid comparators, the A-device session valid comparator with a 0.8V to 2.0V threshold (VA_SESS_VLD) can also be used for the B-device session valid comparator, which requires a 0.8V to 4.0V threshold (VB_SESS_VLD), removing the need for a separate B-device comparator. Implementations with separate A-device and B-device session valid comparators can map into the single session valid definition (VSESS_VLD). ULPI Vbus comparator thresholds are shown in Table 12~~Table 12~~.

The purpose of the VA_VBUS_VLD threshold is to allow the A-device to determine whether or not it is able to output a valid voltage on VBUS. Thus, the upper limit on the VA_VBUS_VLD threshold is not specified by ULPI. However, the upper limit on this threshold is generally dependent on the characteristics of the A-device power supply.

Thus, if the A-device power supply operates by driving VBUS to a reference of VA_VBUS_REF, and the output voltage does not drop below x% when the B-devices on its Targeted Peripheral List are not drawing too much current, then the threshold voltage would be:

$4.4V < V_{A\_VBUS\_VLD} \leq (x / 100) * V_{A\_VBUS\_REF}$

where x is determined by the power supply designer.

Threshold Name	Threshold Range
B-device session end	$0.2 \leq V_{SESS\_END} < 0.8$
A/B-device session valid	$0.8 \leq V_{SESS\_VLD} < 2.0$
A-device Vbus Valid	$4.4 \leq V_{A\_VBUS\_VLD}$

Table 12 – 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 13 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 13 also indicates typical applications of each setting. Figure 45 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 13. 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.[13]

Table 13 – RX CMD VbusValid over-current conditions

[13] 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.

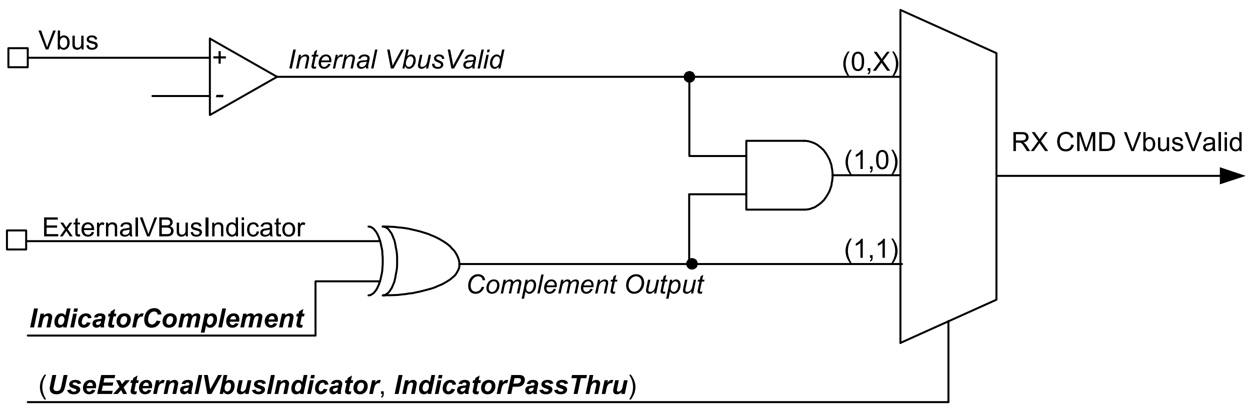

Figure 45 – RX CMD VA_VBUS_VLD ≤ Vbus indication source

Depending on the application, the link should enable or disable the appropriate VBUS interrupts. Example settings for typical applications are given in Table 14.

Application	VbusValid[14]	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 14 – Vbus indicators in the RX CMD required for typical applications

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

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~~ and Vbus comparators~~. 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.

3.9.1 Data Line Definition For Low Power Mode

When in Low Power Mode, the PHY drives **data**(3:0) with the signals listed in Table 15~~Table 12~~. When in Low Power Mode, **LineState** must be driven combinatorially from the FS receiver. The **int** pin is asserted whenever any unmasked interrupt occurs. The PHY must latch interrupt events directly from analog circuitry because **clock** is powered down. The Link ignores the **data**(7:4) signals for 8-bit data bus configurations. These rules must be followed for both input clock and output clock modes.

Signal	Maps to	Direction	Description
linestate(0)	**data**(0)	OUT	Combinatorial **LineState**(0) driven directly by FS analog receiver.
linestate(1)	**data**(1)	OUT	Combinatorial **LineState**(1) driven directly by FS analog receiver.
reserved	**data**(2)	OUT	Reserved. The PHY must drive this pin to low.
int	**data**(3)	OUT	Active high interrupt indication. Must be asserted whenever any unmasked interrupt occurs.

Table 15~~14~~ – Interface signal mapping during Low Power Mode

3.9.2 Entering Low Power Mode

The Link sets ***SuspendM*** in the **Function Control** register to 0b to place the PHY into Low Power Mode. The Link or PHY clock may be stopped a minimum of five cycles after the PHY accepts the register write data, as shown in Figure 46~~Figure 45~~. While in Low Power Mode, the PHY asserts **dir**, and holds **nxt** low. There is one cycle of data bus turnaround provided after the assertion of **dir**, during which the value on **data** is not valid. The PHY starts driving the signals of Table 15~~Table 12~~ immediately after the turnaround cycle.

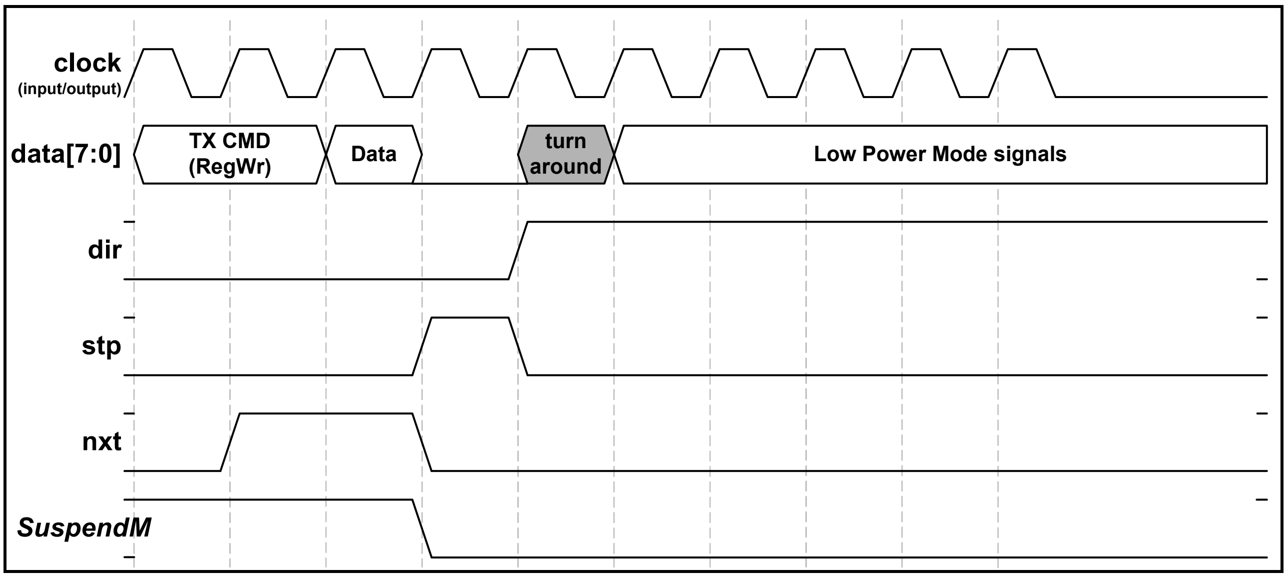

Figure 46 – Entering low power mode

3.9.3 Exiting Low Power Mode

As shown in Figure 47~~Figure 46~~ and Figure 48~~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 15~~Table 12~~ immediately before the turnaround cycle.

As shown in Figure 48~~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~~Table 4 and section 3.8.5.4.4~~ for more information on TPREP.

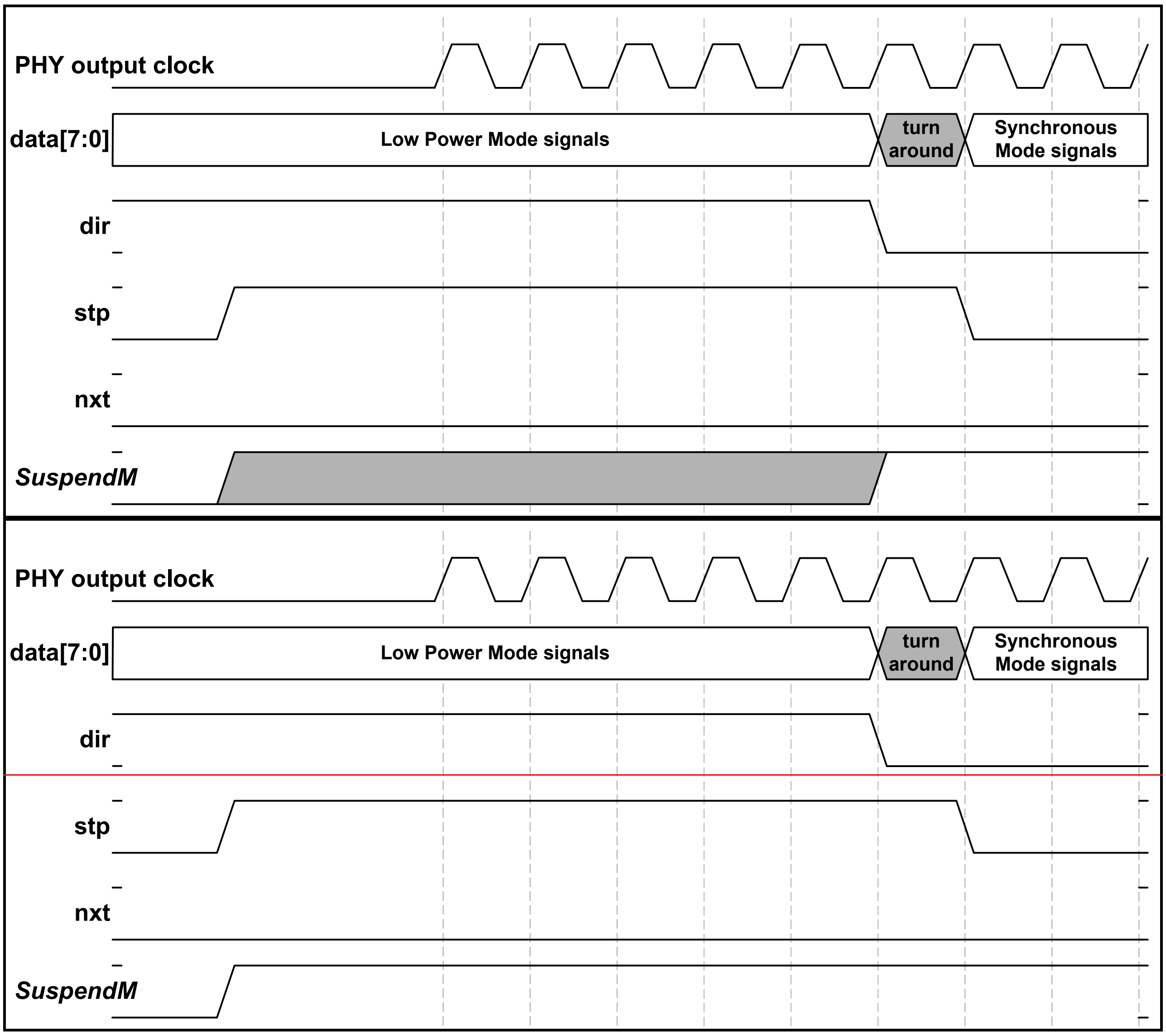

Figure 47 – Exiting low power mode when PHY provides output clock

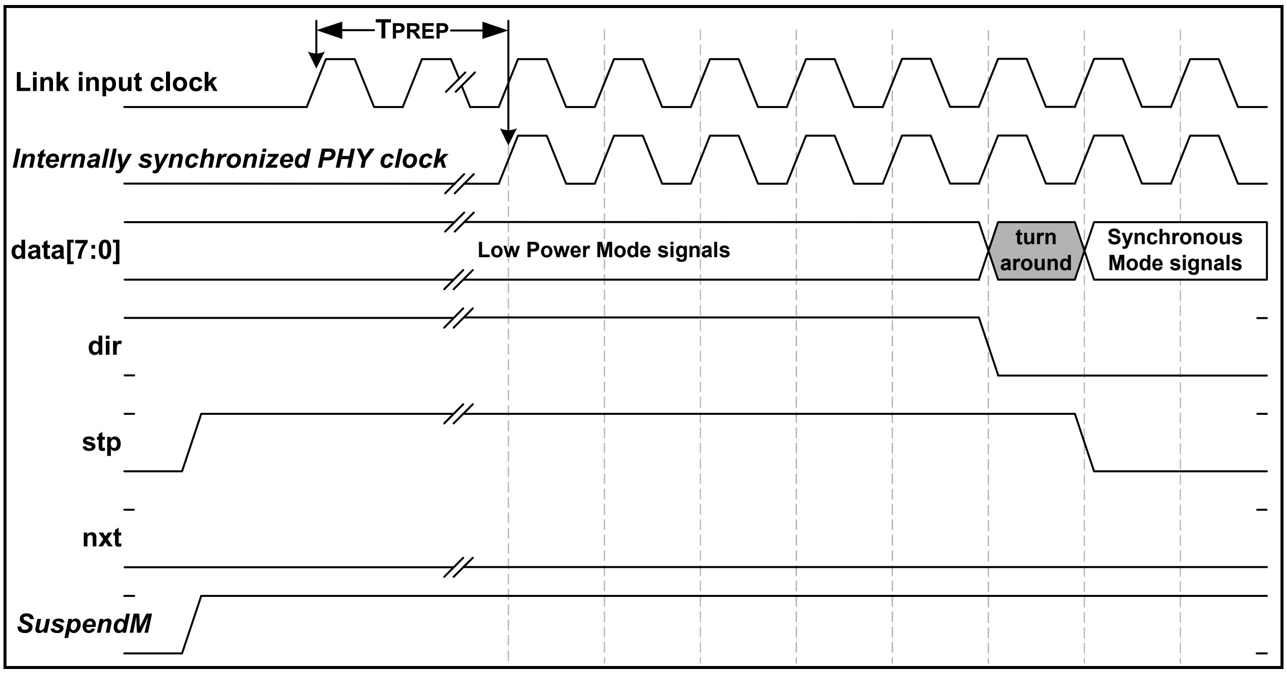

Figure 48 – Exiting low power mode when Link provides input clock

3.9.4 False Resume Rejection

Noise in the USB environment may cause momentary non-J-state conditions or glitches on the asynchronous **LineState** outputs from the PHY during Low Power Mode. Glitches on **LineState** can potentially cause momentary assertions of **stp** from the Link to the PHY. These should be considered as false resume events and should be rejected by the PHY. The PHY may optionally reject glitches on **stp** prior to re-starting the clock, as shown in Figure 49~~Figure 48~~. The PHY must qualify the **stp** assertion that is used to exit Low Power Mode by checking that **stp** is active on the clock edge that the PHY de-asserts **dir**. If **stp** is not asserted during the cycle that **dir** is de-asserted, then the PHY must re-assert **dir** and return to Low Power Mode without further action from the Link, as shown in Figure 50~~Figure 49~~.

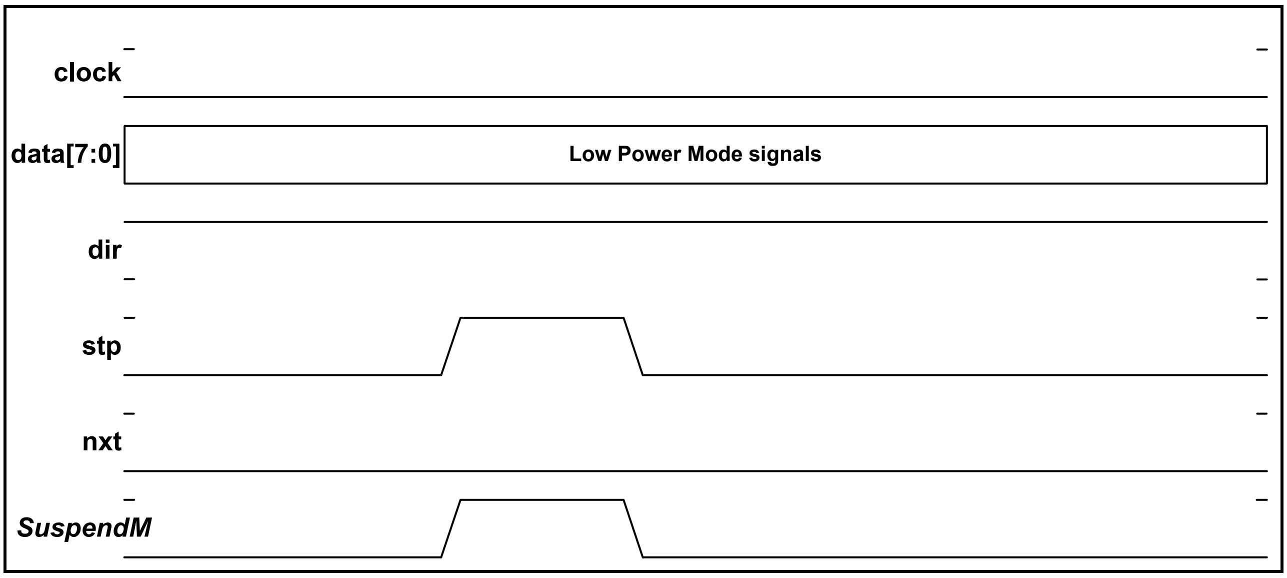

Figure 49 – PHY stays in Low Power Mode when stp de-asserts before clock starts.

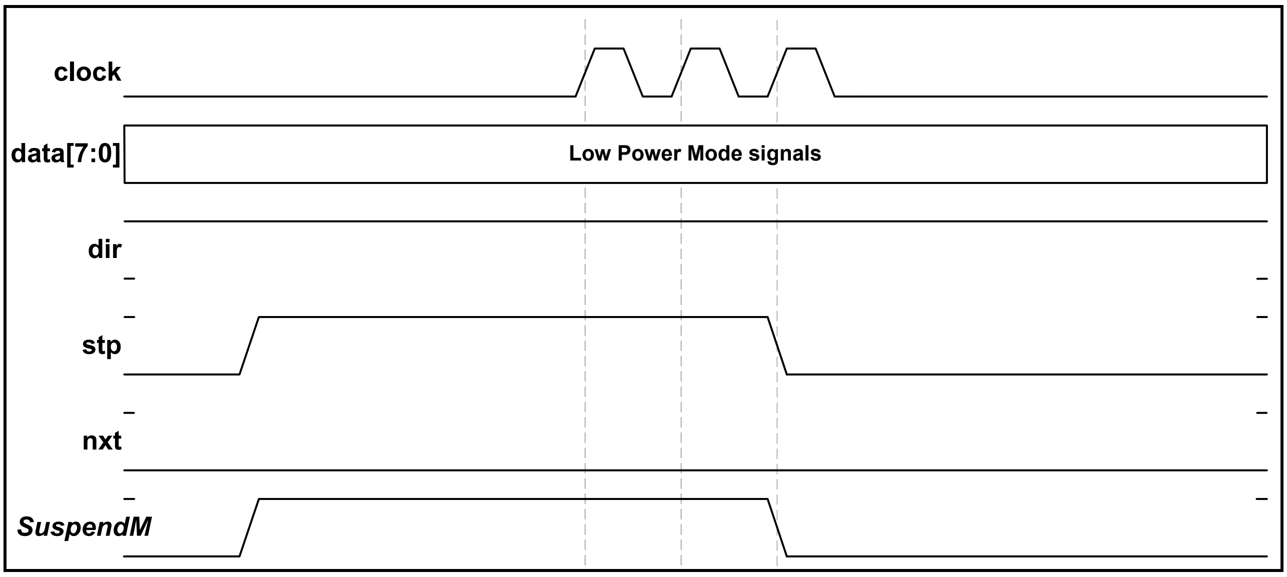

Figure 50 – PHY re-enters Low Power Mode when stp de-asserts before dir de-asserts.

3.10 Full Speed / Low Speed Serial Mode (Optional)

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. Unlike Low Power Mode, the FS/LS transmitters must be powered. Any function must be powered if its corresponding register bit is set, including interrupt sources and the charge pump.

3.10.1 Data Line Definition For FsLsSerialMode

The 6-pin serial mode provides a separate data line for each serial signal defined in UTMI+. The 3-pin serial mode compresses the USB receive and transmit data onto bi-directional data lines and is provided so that 4-bit data bus PHY implementations can support serial signalling. When in 6-pin serial mode, the PHY drives data(7:0) with the signals listed in Table 16~~Table 13~~. When in 3-pin serial mode, the PHY drives **data**(3:0) with the signals listed in Table 17~~Table 14~~.

In either mode, the active-low **tx_enable_n** signal of UTMI+ is converted to active high **tx_enable** for transmission over the ULPI bus. This allows the default ULPI bus idle state to remain at 00h.

In both serial modes, an interrupt pin is provided. The **int** pin is asserted whenever an unmasked interrupt event occurs.

Signal	Maps to	Direction	Description
tx_enable	**data**(0)	IN	Active high transmit enable.
tx_dat	**data**(1)	IN	Transmit differential data on **D+**/**D-**.
tx_se0	**data**(2)	IN	Transmit single-ended zero on **D+**/**D-**.
int	**data**(3)	OUT	Active high interrupt indication. Must be asserted whenever any unmasked interrupt occurs.
rx_dp	**data**(4)	OUT	Single-ended receive data from **D+**.
rx_dm	**data**(5)	OUT	Single-ended receive data from **D-**.
rx_rcv	**data**(6)	OUT	Differential receive data from **D+**/**D-**.
reserved	**data**(7)	OUT	Reserved. The PHY must drive this pin to low.

Table 16~~15~~ – Serial Mode signal mapping for 6-pin FsLsSerialMode

Signal	Maps to	Direction	Description
tx_enable	**data**(0)	IN	Active high transmit enable.
dat	**data**(1)	I/O	Transmit differential data on **D+**/**D-** when **tx_enable** is high. Receive differential data from **D+**/**D-** when **tx_enable** is low.
se0	**data**(2)	I/O	Transmit single-ended zero on **D+**/**D-** when **tx_enable** is high. Receive single-ended zero from **D+**/**D-** when **tx_enable** is low.
int	**data**(3)	OUT	Active high interrupt indication. Must be asserted whenever any unmasked interrupt occurs.

Table 17~~16~~ – Serial Mode signal mapping for 3-pin FsLsSerialMode

3.10.2 Entering FsLsSerialMode

To enter 6-pin serial mode, the Link sets the ***6-Pin FsLsSerialMode*** bit in the **Interface Control** register, causing the data bus to switch to asynchronous operation and take on a new definition as shown in Table 16~~Table 13~~.

To enter 3-pin serial mode, the Link sets the ***3-Pin FsLsSerialMode*** bit in the **Interface Control** register, causing the data bus to switch to asynchronous operation and take on a new definition as shown in Table 17~~Table 14~~.

By default, the output **clock** is powered down to reduce power consumption in serial mode. Entering Serial Mode and stopping the clock shown in Figure 51~~Figure 50~~. The clock can be stopped a minimum of 5 clock cycles after Serial Mode is entered. When the clock is not usable, the PHY must assert **dir**. There is one cycle of data bus turnaround provided after the assertion of **dir**, during which the value on **data** is not valid. The PHY starts driving the serial mode signals immediately after the turnaround cycle.

If the Link requires the clock to be running in Serial Mode, it can set the ***ClockSuspendM*** signal in the **Interface Control** register before entering Serial Mode. Entering Serial Mode with the clock running is shown in Figure 52~~Figure 51~~.

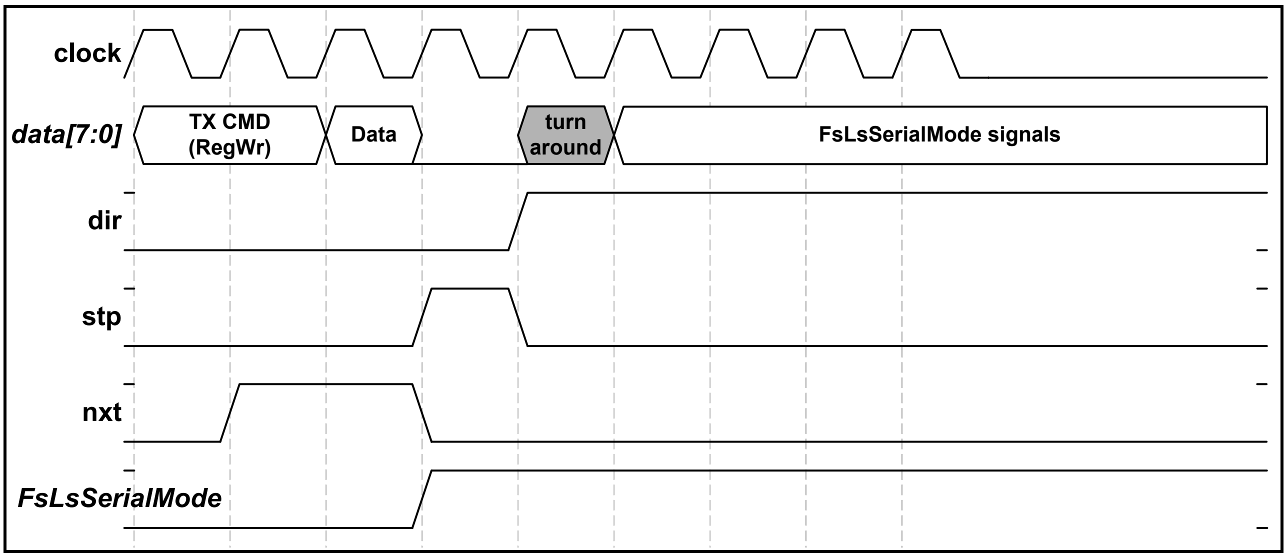

Figure 51 – Interface behaviour when entering Serial Mode and clock is powered down

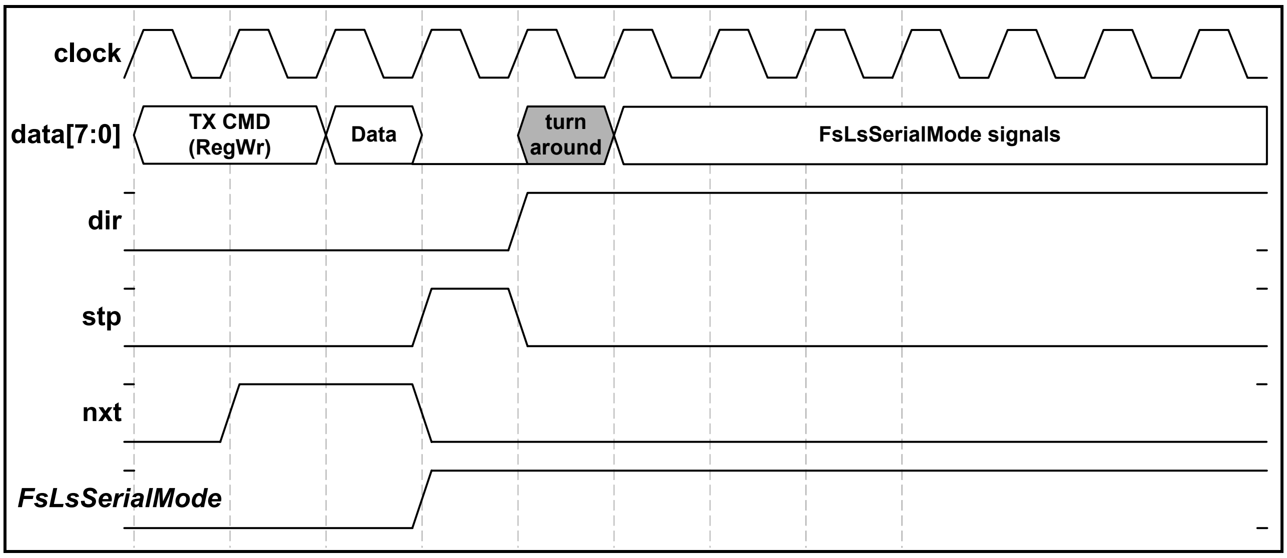

Figure 52 – Interface behaviour when entering Serial Mode and clock remains powered

3.10.3 Exiting FsLsSerialMode

When the Link detects **int** as high, it should exit ***FsLsSerialMode*** by asserting **stp**. When the ULPI interface has returned to Synchronous Mode, the Link can read the **USB Interrupt Latch** register to determine the source of the interrupt. Exiting ***FsLsSerialMode*** is identical to exiting Low Power Mode if the clock is not running. Exiting ***FsLsSerialMode*** when the clock is not running is shown in Figure 53~~Figure 52~~.

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 54~~in the following cycle~~. 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.

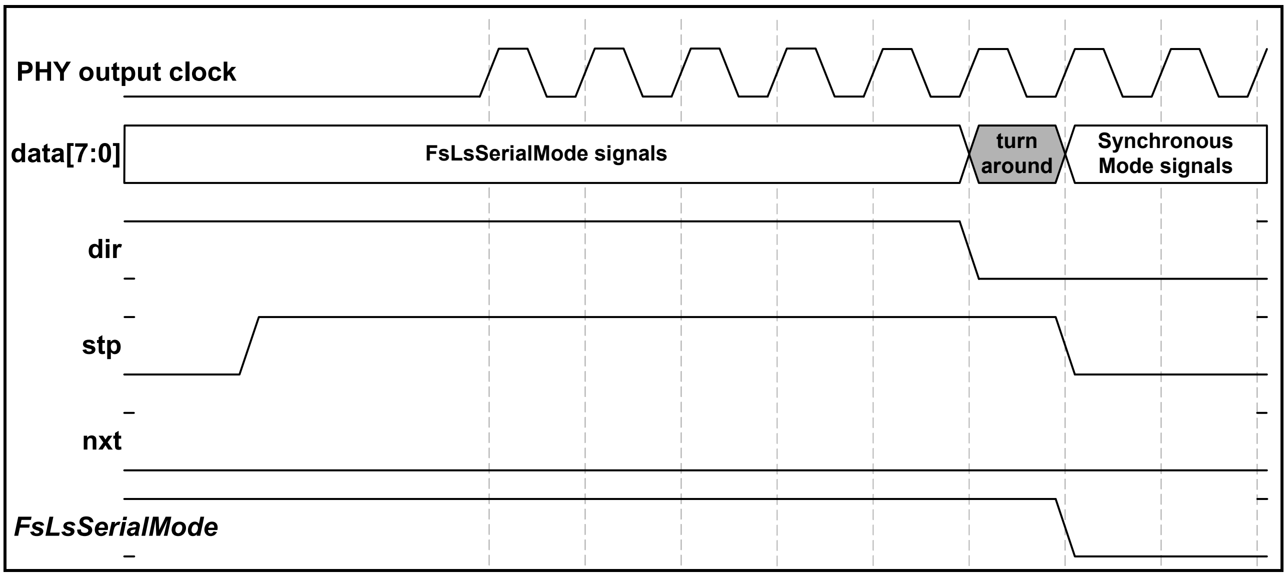

Figure 53 – Interface behaviour when exiting Serial Mode and clock is not running

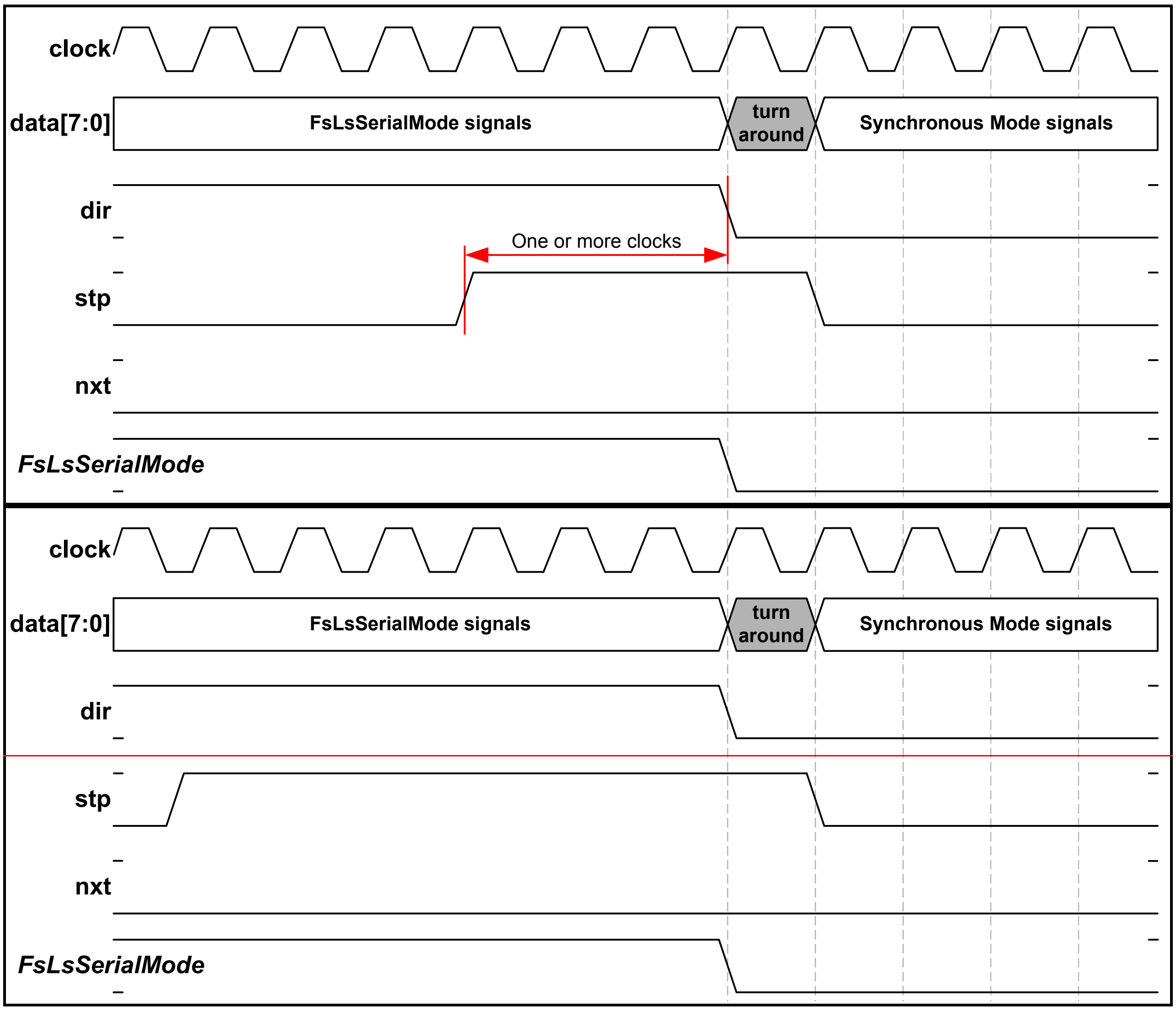

Figure 54 – Interface behaviour when exiting Serial Mode and clock is running

3.11 Carkit Mode (Optional)

This mode is selected when the ***CarkitMode*** bit in the **Interface Control** register is set. It allows the link to communicate through the PHY to a remote carkit using UART signalling. Refer to [Ref 6] for more information on carkit. In cases of conflict between this specification and the Carkit specification, the Carkit specification shall take precedence.

By default, the clock is powered down when the PHY enters Carkit Mode. Entering and exiting Carkit Mode is identical to Serial Mode. If the Link requires the clock to be running in Carkit Mode, it can set the ***ClockSuspendM*** signal in the **Interface Control** register before entering Carkit Mode.

The **int** pin is asserted whenever an unmasked interrupt event occurs. When the Link detects **int** as high, it should wake-up the clock (if powered down) by asserting **stp**. If the clock is already running, the Link asserts **stp** for 1 cycle to switch the interface to Synchronous Mode. When the PHY is in Synchronous Mode, the Link can read the **Carkit Interrupt Latch** register to determine the source of the interrupt.

Signal	Maps to	Direction	Description
txd	**data**(0)	IN	UART TXD signal that is routed to **D-** pin.
rxd	**data**(1)	OUT	UART RXD signal that is routed from **D+** pin.
reserved	**data**(2)	-	Reserved.
int	**data**(3)	OUT	Active high interrupt indication. Must be asserted whenever any unmasked interrupt occurs.

Table 18~~17~~ – Carkit signal mapping

3.12 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 55 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.

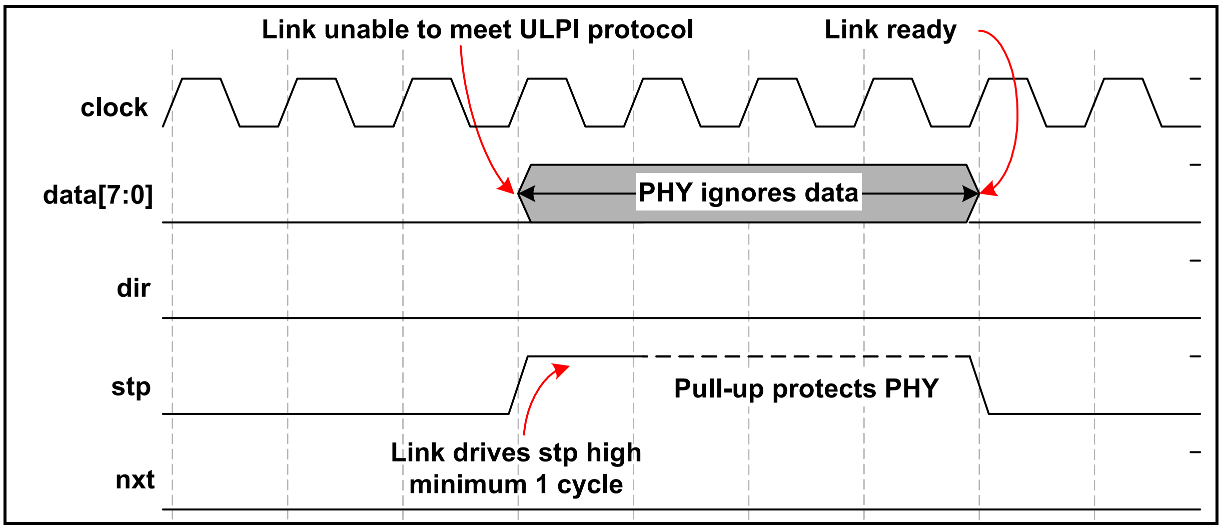

Figure 55 – PHY interface protected when the clock is running

During power up or when the clock is not running, the PHY always asserts **dir**, protecting its **data** inputs. As shown in Figure 56 and Figure 57, 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 57, 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.

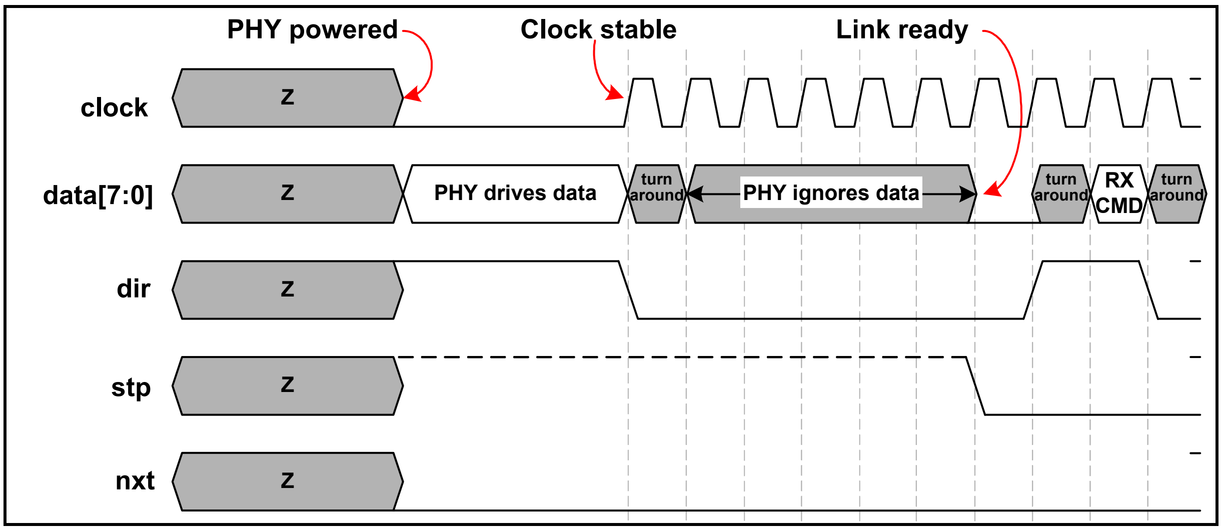

Figure 56 – Power up sequence when PHY powers up before the link. Interface is protected.

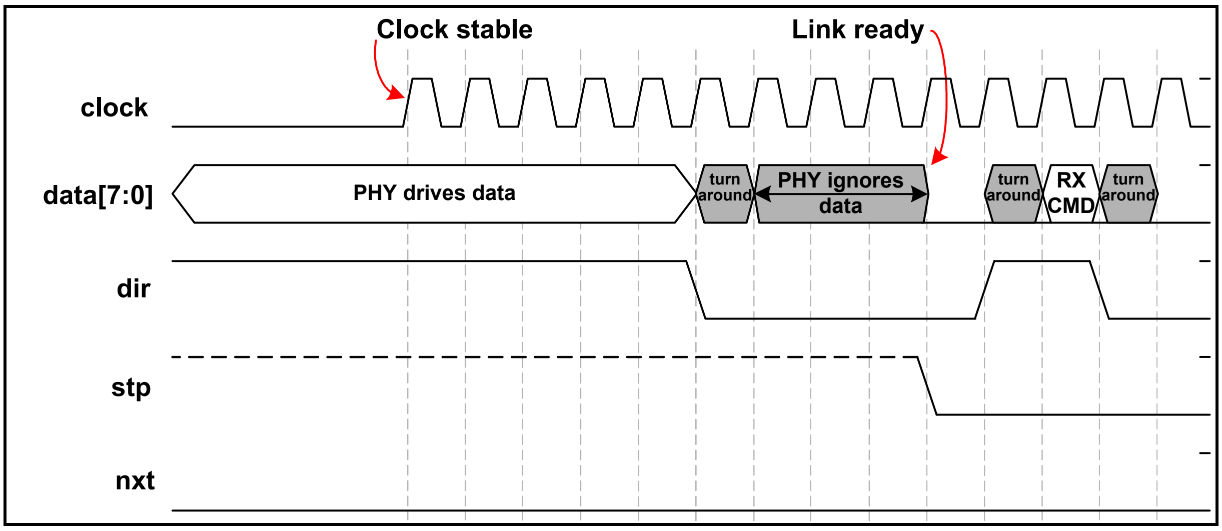

Figure 57 – PHY automatically exits Low Power Mode with interface protected

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 58. 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.

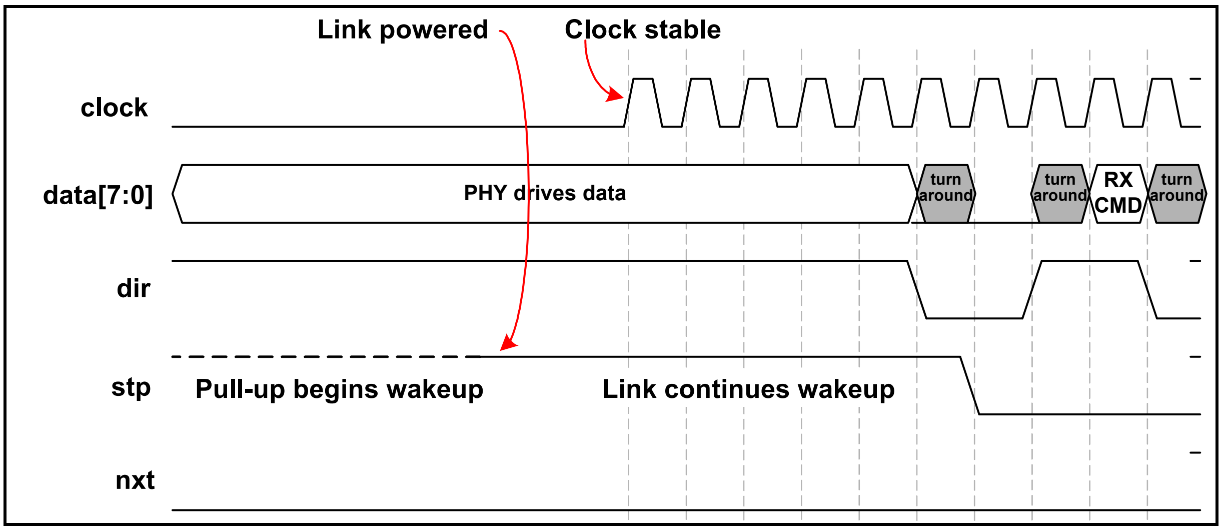

Figure 58 – Link resumes driving ULPI bus and asserts stp because clock is not running

As shown in Figure 59, 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.

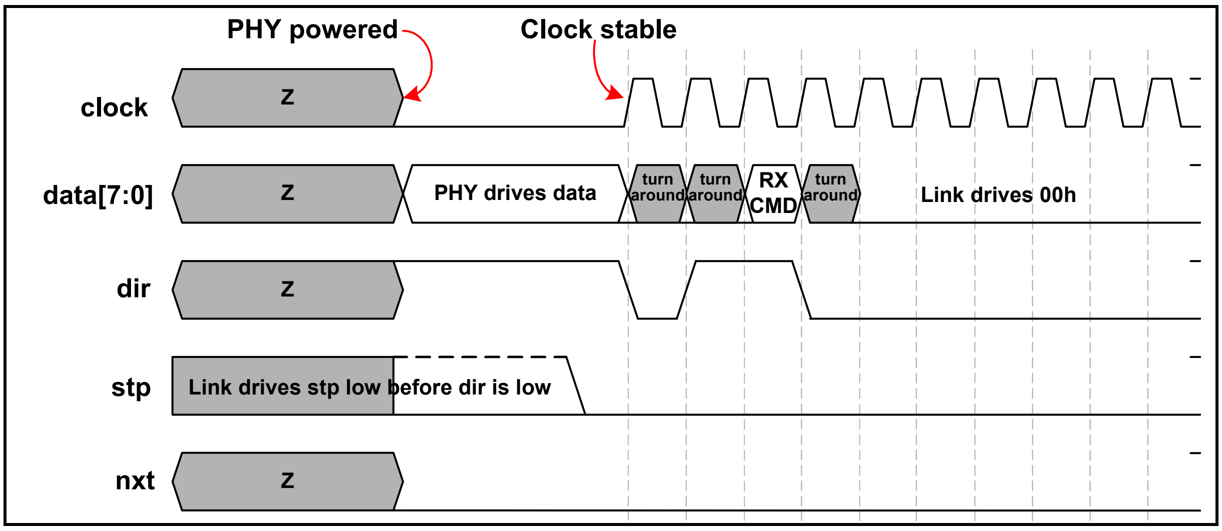

Figure 59 – Power up sequence when link powers up before PHY (ULPI 1.0 compliant links)

4. Registers

4.1 Register Map

As shown in Table 19~~Table 16~~, ULPI provides an immediate register set with a 6-bit address that forms part of the **Transmit Command Byte**. An extended register set is also provided with an 8-bit address that requires an extra clock cycle to complete. The immediate register set is mirrored into the lower end of the extended address space. That is, reading or writing to extended address “00XXXXXX” will in fact operate on the immediate register set. The PHY must support both immediate and extended register operations. The register access legend of Table 20~~Table 17~~ applies.

Field name	Size (bits)	Address (6 bits)			
		Rd	Wr	Set	Clr
Immediate Register Set					
Vendor ID Low	8	00h	-	-	-
Vendor ID High	8	01h	-	-	-
Product ID Low	8	02h	-	-	-
Product ID High	8	03h	-	-	-
Function Control	8	04-06h	04h	05h	06h
Interface Control	8	07-09h	07h	08h	09h
OTG Control	8	0A-0Ch	0Ah	0Bh	0Ch
USB Interrupt Enable Rising	8	0D-0Fh	0Dh	0Eh	0Fh
USB Interrupt Enable Falling	8	10-12h	10h	11h	12h
USB Interrupt Status	8	13h	-	-	-
USB Interrupt Latch	8	14h	-	-	-
Debug	8	15h	-	-	-
Scratch Register	8	16-18h	16h	17h	18h
Carkit Control (*Optional*)	8	19-1Bh	19h	1Ah	1Bh
Carkit Interrupt Delay (*Optional*)	8	1Ch	1Ch		
Carkit Interrupt Enable (*Optional*)	8	1Dh-1Fh	1Dh	1Eh	1Fh
Carkit Interrupt Status (*Optional*)	8	20h	-	-	-
Carkit Interrupt Latch (*Optional*)	8	21h	-	-	-
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	-	-
Reserved	8	28~~22~~-2Eh			
Access Extended Register Set (see below)	8	-	2Fh	-	-
Vendor-specific	8	30-3Fh			
Extended Register Set		**Address (8 bits)**			
Maps to Immediate Register Set above	8	00-3Fh			
Reserved	8	40-7Fh			
Vendor-Specific	8	80-FFh			

Table 19~~18~~ – Register map

Access Code	Expanded Name	Meaning
rd	Read	Register can be read. Read-only if this is the only mode given.
wr	Write	Pattern on the data bus will be written over all bits of the register.
s	Set	Pattern on the data bus is OR'd with and written into the register.
c	Clear	Pattern on the data bus is a mask. If a bit in the mask is set, then the corresponding register bit will be set to zero (cleared).

Table 20~~19~~ – Register access legend

4.2 Immediate Register Set

4.2.1 Vendor ID and Product ID

Address: 00h-03h Read-only.

Register	Bits	Access	Address	Reset	Description
Vendor ID Low	7:0	rd	00h	Fixed	Lower byte of Vendor ID supplied by USB-IF.
Vendor ID High	7:0	rd	01h	Fixed	Upper byte of Vendor ID supplied by USB-IF.
Product ID Low	7:0	rd	02h	Fixed	Lower byte of Product ID supplied by Vendor.
Product ID High	7:0	rd	03h	Fixed	Upper byte of Product ID supplied by Vendor.

Table 21~~20~~ – Vendor ID and Product ID register description

4.2.2 Function Control

Address: 04h-06h (Read), 04h (Write), 05h (Set), 06h (Clear).

Controls UTMI function settings of the PHY.

Field name	Bit	Access	Reset	Description
XcvrSelect	1:0	rd/wr/s/c	01b	Selects the required transceiver speed. 00b : Enable HS transceiver 01b : Enable FS transceiver 10b : Enable LS transceiver 11b : Enable FS transceiver for LS packets (FS preamble is automatically pre-pended)
TermSelect	2	rd/wr/s/c	0b	Controls the internal 1.5kΩ pull-up resistor and 45Ω HS terminations. Control over bus resistors changes depending on ***XcvrSelect***, ***OpMode***, ***DpPulldown*** and ***DmPulldown***, as shown in Table 41~~Table 34~~. Since Low Speed peripherals never support Full Speed or Hi-Speed, providing the 1.5kΩ on **D-** for low speed is optional.
OpMode	4:3	rd/wr/s/c	00b	Selects the required bit encoding style during transmit. 00b : Normal operation 01b : Non-driving 10b : Disable bit-stuff and NRZI encoding 11b : Optional. Do not automatically add SYNC and EOP when transmitting. Must be used only for HS packets.
Reset	5	rd/wr/s/c	0b	Active high transceiver reset. After the Link sets this bit, the PHY must assert **dir** and reset the UTMI+ core. When the reset is completed, the PHY de-asserts **dir** and automatically clears this bit. After de-asserting **dir**, the PHY must re-assert **dir** and send an **RX CMD** update to the Link. The Link must wait for **dir** to de-assert before using the ULPI bus. Does not reset the ULPI interface or ULPI register set.
SuspendM	6	rd/wr/s/c	1b	Active low PHY suspend. Put PHY into Low Power Mode. The PHY can power down all blocks except the full speed receiver, OTG comparators, and the ULPI interface pins. The PHY must automatically set this bit to '1' when Low Power Mode is exited. 0b : Low Power Mode 1b : Powered
Reserved	7	rd/wr/s/c	0b	Reserved.

Table 22~~21~~ – Function Control register

4.2.3 Interface Control

Address: 07h-09h (Read), 07h (Write), 08h (Set), 09h (Clear).

Enables alternative interfaces and PHY features. All bits in this register are optional features of the PHY.

Field name	Bits	Access	Reset	Description
6-pin FsLsSerialMode	0	rd/wr/s/c	0b	Changes the ULPI interface to 6-pin Serial Mode. The PHY must automatically clear this bit when serial mode is exited. It is optional for any ULPI PHY to support this mode. A PHY that has only 4 data pins can never support this mode. 0b : FS/LS packets are sent using parallel interface. 1b : FS/LS packets are sent 6-pin using serial interface.
3-pin FsLsSerialMode	1	rd/wr/s/c	0b	Changes the ULPI interface to 3-pin Serial Mode. The PHY must automatically clear this bit when serial mode is exited. It is optional for any ULPI PHY to support this mode. 0b : FS/LS packets are sent using parallel interface. 1b : FS/LS packets are sent using 4-pin serial interface.
CarkitMode	2	rd/wr/s/c	0b	Changes the ULPI interface to carkit interface. The PHY must automatically clear this bit when carkit mode is exited. It is optional for any ULPI PHY to support this mode. Refer to [Ref 6] for more information on carkit. 0b : Disable serial carkit mode. 1b : Enable serial carkit mode.
ClockSuspendM	3	rd/wr/s/c	0b	Active low clock suspend. Valid only in Serial Mode and Carkit Mode. Powers down the internal clock circuitry only[15]. Valid only when ***SuspendM*** = 1b. The PHY must ignore ***ClockSuspend*** when ***SuspendM*** = 0b. By default, the clock will not be powered in Serial and Carkit Modes. 0b : Clock will not be powered in Serial and Carkit Modes. 1b : Clock will be powered in Serial and Carkit Modes.
AutoResume	4	rd/wr/s/c	Xb	Enables the PHY to automatically transmit resume signaling. Any PHY that cannot wake up its clock in less than 1ms must implement this feature and have it turned on by default. The Link can disable this feature if needed. Any PHY that can wake up its clock in less than 1ms can optionally implement this feature and should have it turned off by default. Refer to USB specification 7.1.7.7 and 7.9 for more details.
Indicator Complement	5	rd/wr/s/c	0b	Tells the PHY to invert the **ExternalVbusIndicator** input signal, generating the *Complement Output*. Refer to 3.8.7.3 and Figure 45 for more details. 0b: PHY will not invert **ExternalVbusIndicator** signal (default). 1b: PHY will invert **ExternalVbusIndicator** signal.

[15] Useful for Link implementations that switch to serial or carkit mode(s), and do not require a clock source from the PHY.

Field name	Bits	Access	Reset	Description
Indicator PassThru	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 **RX CMD**. Refer to 3.8.7.3 and Figure 45 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.
Interface Protect Disable~~Reserved~~	7~~7:5~~	rd/wr/s/c ~~rd/wr/s/c~~	0b~~000b~~	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 3.12 for more details. 0b : Enables the interface protect circuit (default). 1b : Disables the interface protect circuit.~~Reserved.~~

Table 23~~22~~ – Interface Control register

4.2.4 OTG Control

Address: 0Ah-0Ch (Read), 0Ah (Write), 0Bh (Set), 0Ch (Clear). Controls UTMI+ OTG functions of the PHY.

Field name	Bits	Access	Reset	Description
IdPullup	0	rd/wr/s/c	0b	Connects a pull-up to the **ID** line and enables sampling[16] of the signal level. 0b : Disable sampling of ID line. 1b : Enable sampling of ID line.
DpPulldown	1	rd/wr/s/c	1b	Enables the 15k Ohm pull-down resistor on **D+**. 0b : Pull-down resistor not connected to **D+**. 1b : Pull-down resistor connected to **D+**.
DmPulldown	2	rd/wr/s/c	1b	Enables the 15k Ohm pull-down resistor on **D-**. 0b : Pull-down resistor not connected to **D-**. 1b : Pull-down resistor connected to **D-**.
DischrgVbus	3	rd/wr/s/c	0b	Discharge VBUS through a resistor. A minimum of 656Ω is defined in the OTG specification [Ref 2]. If the Link sets this bit to 1, it waits for an **RX CMD** indicating ***SessEnd*** has transitioned from 0 to 1, and then resets this bit to 0 to stop the discharge. 0b : do not discharge VBUS 1b : discharge VBUS
ChrgVbus	4	rd/wr/s/c	0b	Charge VBUS through a resistor. Used for VBUS pulsing SRP. A minimum of 281Ω output impedance with a voltage source of 3.0V is one example given in the OTG specification [Ref 2]. The Link must first check that VBUS has been discharged (see ***DischrgVbus*** bit), and that both **D+** and **D-** data lines have been low (SE0) for 2ms. 0b : do not charge VBUS 1b : charge VBUS
DrvVbus	5	rd/wr/s/c	0b	Signals the internal charge pump or external supply to drive 5V on VBUS. ~~An internal charge pump is optional.~~ 0b : do not drive Vbus~~BUS~~ (default) 1b : drive 5V on Vbus~~BUS~~
DrvVbus External	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.~~Signals the external charge pump to drive 5V on VBUS. External charge pump support is optional.~~ 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. ~~0b : Do not drive VBUS using external charge pump.~~ ~~1b : Drive VBUS using external charge pump.~~
UseExternal VbusIndicator	7	rd/wr/s/c	~~Xb~~0b	Tells the PHY to use an external VBUS over-current indicator. This bit is optional. Refer to 3.8.7.3. 0b: Use the internal OTG comparator (VA_VBUS_VLD) or

[16] If the Carkit Mode registers are implemented, the pull-up resistor and detection circuitry shall comply with [Ref 6].

Field name	Bits	Access	Reset	Description
				internal VBUS valid indicator (default). 1b: Use external VBUS valid indicator signal~~Reserved.~~

Table 24~~23~~ – OTG Control register

4.2.5 USB Interrupt Enable Rising

Address: 0Dh-0Fh (Read), 0Dh (Write), 0Eh (Set), 0Fh (Clear).

If set, the bits in this register cause an interrupt event notification to be generated when the corresponding PHY signal changes from low to high. By default, all transitions are enabled. **RxActive** and **RxError** must always be communicated immediately and so are not included in this register. Interrupt circuitry can be powered down in any mode when both rising and falling edge enables are disabled. To ensure interrupts are detectable when **clock** is powered down, the link should enable both rising and falling edges.

Field name	Bits	Access	Reset	Description
Hostdisconnect Rise	0	rd/wr/s/c	1b	Generate an interrupt event notification when **Hostdisconnect** changes from low to high. Applicable only in host mode (***DpPulldown*** and ***DmPulldown*** both set to 1b).
VbusValid Rise	1	rd/wr/s/c	1b	Generate an interrupt event notification when **VbusValid** changes from low to high.
SessValid Rise	2	rd/wr/s/c	1b	Generate an interrupt event notification when **SessValid** changes from low to high. **SessValid** is the same as UTMI+ **AValid**.
SessEnd Rise	3	rd/wr/s/c	1b	Generate an interrupt event notification when **SessEnd** changes from low to high.
IdGnd Rise	4	rd/wr/s/c	1b	Generate an interrupt event notification when **IdGnd** changes from low to high. **IdGnd** is valid 50ms after ***IdPullup*** is set to 1b, otherwise **IdGnd** is undefined and should be ignored.
Reserved	7:5	rd/wr/s/c	0b	Reserved.

Table 25~~24~~ – USB Interrupt Enable Rising register

4.2.6 USB Interrupt Enable Falling

Address: 10h-12h (Read), 10h (Write), 11h (Set), 12h (Clear).

If set, the bits in this register cause an interrupt event notification to be generated when the corresponding PHY signal changes from high to low. By default, all transitions are enabled. **RxActive** and **RxError** must always be communicated immediately and so are not included in this register. Interrupt circuitry can be powered down in any mode when both rising and falling edge enables are disabled. To ensure interrupts are detectable when **clock** is powered down, the link should enable both rising and falling edges.

Field name	Bits	Access	Reset	Description
Hostdisconnect Fall	0	rd/wr/s/c	1b	Generate an interrupt event notification when **Hostdisconnect** changes from high to low. Applicable only in host mode.
VbusValid Fall	1	rd/wr/s/c	1b	Generate an interrupt event notification when **VbusValid** changes from high to low.
SessValid Fall	2	rd/wr/s/c	1b	Generate an interrupt event notification when **SessValid** changes from high to low. **SessValid** is the same as UTMI+ **AValid**.
SessEnd Fall	3	rd/wr/s/c	1b	Generate an interrupt event notification when **SessEnd** changes from high to low.
IdGnd Fall	4	rd/wr/s/c	1b	Generate an interrupt event notification when **IdGnd** changes from high to low. **IdGnd** is valid 50ms after ***IdPullup*** is set to 1b, otherwise **IdGnd** is undefined and should be ignored.
Reserved	7:5	rd/wr/s/c	0b	Reserved.

Table 26~~25~~ – USB Interrupt Enable Falling register

4.2.7 USB Interrupt Status

Address: 13h (Read-only).

Indicates the current value of the interrupt source signal. Interrupt circuitry can be powered down in any mode when both rising and falling edge enables are disabled. To ensure interrupts are detectable when **clock** is powered down, the link should enable both rising and falling edges.

Field name	Bits	Access	Reset	Description
Hostdisconnect	0	rd	0b	Current value of UTMI+ **Hostdisconnect** output. Applicable only in host mode. Automatically reset to 0b when Low Power Mode is entered.
VbusValid	1	rd	0b	Current value of UTMI+ **VbusValid** output.
SessValid	2	rd	0b	Current value of UTMI+ **SessValid** output. **SessValid** is the same as UTMI+ **AValid**.
SessEnd	3	rd	0b	Current value of UTMI+ **SessEnd** output.
IdGnd	4	rd	0b	Current value of UTMI+ **IdGnd** output. **IdGnd** is valid 50ms after ***IdPullup*** is set to 1b, otherwise **IdGnd** is undefined and should be ignored.
Reserved	7:5	rd	Xb	Reserved.

Table 27~~26~~ – USB Interrupt Status register

4.2.8 USB Interrupt Latch

Address: 14h (Read-only with auto-clear).

These bits are set by the PHY when an unmasked change occurs on the corresponding internal signal. The PHY will automatically clear all bits when the Link reads this register, or when Low Power Mode is entered. The PHY also clears this register when Serial Mode or Carkit Mode is entered regardless of the value of ***ClockSuspendM***. Interrupt circuitry can be powered down in any mode when both rising and falling edge enables are disabled. To ensure interrupts are detectable when **clock** is powered down, the link should enable both rising and falling edges.

The PHY must follow the rules in Table 29~~Table 29~~ for setting any latch register bit. It is important to note that if register read data is returned to the Link in the same cycle that a **USB Interrupt Latch** bit is to be set, the interrupt condition is given immediately in the register read data and the Latch bit is not set.

Note that it is optional for the Link to read the **USB Interrupt Latch** register in Synchronous Mode because the **RX CMD** byte already indicates the interrupt source directly.

Field name	Bits	Access	Reset	Description
Hostdisconnect Latch	0	rd	0b	Set to 1b by the PHY when an unmasked event occurs on **Hostdisconnect**. Cleared when this register is read. Applicable only in host mode.
VbusValid Latch	1	rd	0b	Set to 1b by the PHY when an unmasked event occurs on **VbusValid**. Cleared when this register is read.
SessValid Latch	2	rd	0b	Set to 1b by the PHY when an unmasked event occurs on **SessValid**. Cleared when this register is read. **SessValid** is the same as UTMI+ **AValid**.
SessEnd Latch	3	rd	0b	Set to 1b by the PHY when an unmasked event occurs on **SessEnd**. Cleared when this register is read.
IdGnd Latch	4	rd	0b	Set to 1b by the PHY when an unmasked event occurs on **IdGnd**. Cleared when this register is read. **IdGnd** is valid 50ms after ***IdPullup*** is set to 1b, otherwise **IdGnd** is undefined and should be ignored.
Reserved	7:5	rd	0b	Reserved.

Table 28~~27~~ – USB Interrupt Latch register

Input Conditions		Resultant value of Latch Register bit
Register read data returned in current clock cycle	**Interrupt Latch bit is to be set in current clock cycle**	
No	No	0
No	Yes	1
Yes	No	0
Yes	Yes	0

Table 29~~28~~ – Rules for setting Interrupt Latch register bits

4.2.9 Debug

Address: 15h (Read-only).

Indicates the current value of various signals useful for debugging.

Field name	Bits	Access	Reset	Description
LineState0	0	rd	0b	Contains the current value of **LineState**(0)
LineState1	1	rd	0b	Contains the current value of **LineState**(1)
Reserved	7:2	rd	0b	Reserved.

Table 30~~29~~ – Debug register

4.2.10 Scratch Register

Address: 16h-18h (Read), 16h (Write), 17h (Set), 18h (Clear).

Field name	Bits	Access	Reset	Description
Scratch	7:0	rd/wr/s/c	00h	Empty register byte for testing purposes. Software can read, write, set, and clear this register and the PHY functionality will not be affected.

Table 31~~30~~ – Scratch register

4.2.11 Carkit Control

Address: 19h-1Bh (Read), 19h (Write), 1Ah (Set), 1Bh (Clear).

This register is optional. Controls the operation of the carkit circuitry within the PHY. The ***TxdEn*** and ***RxdEn*** bits are ignored if the ***CarkitMode*** bit in the ***Interface Control*** register is not set. If the ***CarkitMode*** bit is set, and the ***RxdEn*** bit is not set, then the data(1) pin is held at logic high.

Refer to [Ref 6] for more information on carkit.

Field name	**Bits**	**Access**	**Reset**	**Description**
CarkitPwr	0	rd/wr/s/c	0b	Applies power to carkit circuitry.
IdGndDrv	1	rd/wr/s/c	0b	Drives ID pin to ground.
TxdEn	2	rd/wr/s/c	0b	Routes TXD signal from data(0) pin onto **D-** pin.
RxdEn	3	rd/wr/s/c	0b	Routes RXD signal from **D+** pin to data(1) pin.
SpkLeftEn	4	rd/wr/s/c	0b	Routes audio signal from SPKR_L pin to **D-** pin.
SpkRightEn	5	rd/wr/s/c	0b	Routes audio signal from SPKR_MIC pin **D+** pin.
MicEn	6	rd/wr/s/c	0b	Routes audio signal from **D+** pin to SPKR_MIC pin.
Reserved	7	-	Xb	Reserved.

Table 32~~31~~ – Carkit Control Register

4.2.12 Carkit Interrupt Delay

Address: 1Ch (Read), 1Ch (Write).

This register is optional. When a carkit interrupts a phone, it pulls **D+** low for an extended period of time. As per [Ref 6], the phone must detect an interrupt if **D+** is below the VPH_DP_LO voltage threshold for a time of TPH_DP_INT.

The Carkit Interrupt Timer in the PHY is enabled whenever the ***CarIntDet*** bit in the **Carkit Interrupt Enable** register is set. If enabled, the Carkit Interrupt Timer increments when the **D+** voltage is below VPH_DP_LO. If the **D+** voltage is above VPH_DP_LO, then the Carkit Interrupt Timer resets. The Carkit Interrupt Timer must reset when **D+** is above VPH_DP_LO for times as short as TCR_INJ_WDTH.

When the Carkit Interrupt Timer reaches the value stored in ***CarIntDly***, the ***CarIntDet*** bit in the **Carkit Interrupt Latch** register is set and an interrupt is generated.

Field name	**Bits**	**Access**	**Reset**	**Description**
CarIntDly	7:0	rd/wr	18h	Sets interrupt time in units of 0.25 ms.

Table 33~~32~~ – Carkit Interrupt Delay register

4.2.13 Carkit Interrupt Enable

Address: 1Dh-1Fh (Read), 1Dh (Write), 1Eh (Set), 1Fh (Clear).

This register is optional. If set, the bits in this register cause an interrupt event notification to be generated when the corresponding PHY signal changes. By default, all bits in this register are cleared.

Refer to [Ref 6] for more information on carkit.

Field name	Bits	Access	Reset	Description
IdFloat Rise	0	rd/wr/s/c	0b	Generate an interrupt event notification when the **ID** pin changes from not floating to floating. The ***IdPullup*** bit in the ***OTG Control*** register must be set, and the **ID** pull-up and detection circuitry must conform to that defined in [Ref 6].
IdFloat Fall	1	rd/wr/s/c	0b	Generate an interrupt event notification when the **ID** pin changes from floating to not floating. The ***IdPullup*** bit in the **OTG Control** register must be set, and the **ID** pull-up and detection circuitry must conform to that defined in [Ref 6].
CarIntDet	2	rd/wr/s/c	0b	Generate an interrupt event notification when the Carkit Interrupt Timer reaches the value stored in the **Carkit Interrupt Delay** register.
CarDp Rise	3	rd/wr/s/c	0b	Generate an interrupt event notification when a rising voltage on the **D+** line crosses the VPH_DP_LO threshold.
CarDp Fall	4	rd/wr/s/c	0b	Generate an interrupt event notification when a falling voltage on the **D+** line crosses the VPH_DP_LO threshold.
Reserved	7:5	rd/wr/s/c	0b	Reserved.

Table 34~~33~~ – Carkit Interrupt Enable register

4.2.14 Carkit Interrupt Status

Address: 20h (Read-only).

This register is optional. When a carkit interrupt event notification occurs, the link can read this register to determine which event triggered the interrupt.

Refer to [Ref 6] for more information on carkit.

Field name	Bits	Access	Reset	Description
IdFloat	0	rd	0b	Asserted when the **ID** pin is floating.
CarIntDet	1	rd	0b	Asserted when the ***CarIntDet*** bit in the **Carkit Interrupt Latch** register is set.
CarDp	2	rd	0b	Asserted when the **D+** line is above the VPH_DP_LO threshold.
Reserved	7:3	rd	0b	Reserved.

Table 35~~34~~ – Carkit Interrupt Status Register

4.2.15 Carkit Interrupt Latch

Address: 21h (Read-only with auto-clear).

This register is optional. These bits are set by the PHY when an unmasked carkit event occurs. The PHY will automatically clear all bits when the Link reads this register, or when Low Power Mode is entered.

The PHY must follow the rules in Table 29~~Table 26~~ for setting any latch register bit. It is important to note that if register read data is returned to the Link in the same cycle that a **Carkit Interrupt Latch** bit is to be set, the interrupt condition is given immediately in the register read data and the latch bit is not set.

Refer to [Ref 6] for more information on carkit.

Field name	Bits	Access	Reset	Description
IdFloat Latch	0	rd	0b	Asserted if the ***IdFloat Rise*** bit in the **Carkit Interrupt Enable** register is set, and the **ID** line changes from not floating to floating. Also asserted if the ***IdFloat Fall*** bit in the **Carkit Interrupt Enable** register is set, and the **ID** line changes from floating to not floating.
CarIntDet Latch	1	rd	0b	Asserted if the ***CarIntDet*** bit in the **Carkit Interrupt Enable** register is set, and the Carkit Interrupt Timer reaches the value stored in the **Carkit Interrupt Delay** register.
CarDp Latch	2	rd	0b	Asserted if the ***CarDp Rise*** bit in the **Carkit Interrupt Enable** register is set, and the **D+** line rises above the VPH_DP_LO threshold. Also asserted if the ***CarDp Fall*** bit is set, and the **D+** line falls below the VPH_DP_LO threshold.
Reserved	7:3	rd	0b	Reserved.

Table 36~~35~~ – Carkit Interrupt Latch register

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 37 – 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 6]. 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 VPH_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 6] for more information on carkit.

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

Table 38 – 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 6] for more information on carkit.

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

Table 39 – 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 6], 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 6] for more information on carkit.

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

Table 40 – Receive Polarity Recovery

~~4.2.16~~4.2.20 Reserved

Address: 28~~1E~~h-2Eh.
These registers are reserved for future use.

~~4.2.17~~4.2.21 Access Extended Register Set

Address: 2Fh (Read/Write).
Accesses the extended register set. See 3.8.3.4 for an explanation of Extended Register accesses.

~~4.2.18~~4.2.22 Vendor-specific

Address: 30h-3Fh.
Allocated for vendor-specific use.

4.3 Extended Register Set

Addresses 00h to 3Fh of the Extended Register Set map directly into the Immediate Set defined in 4.2. A read, write, set, or clear operation to extended address 00h to 3Fh operates on the immediate register set. Addresses 40h to 7Fh are reserved for future use. Addresses 80h to FFh are allocated for Vendor-specific use.

4.4 Register Settings for all Upstream and Downstream signalling modes

Table 41~~Table 34~~ shows what register settings must be applied by the Link to achieve the desired signalling mode. The PHY must generate the correct signalling, prepend the SYNC and append the EOP automatically as required. The table also shows which resistors are enabled as a result of the register settings. The following signals conceptually exist inside the PHY. All resistor signals are active high, a value of 1b enables the resistor, a value of 0b disables the resistor.

- **rpu_dp_en** enables the 1.5kΩ pull-up resistor on **D+**
- **rpu_dm_en** enables the 1.5kΩ pull-up resistor on **D-**
- **rpd_dp_en** enables the 15kΩ pull-down resistor on **D+**
- **rpd_dm_en** enables the 15kΩ pull-down resistor on **D-**
- **hsterm_en** enables the 45Ω termination resistors on **D+** and **D-**

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 41~~40~~ – Upstream and downstream signalling modes

5. T&MT Connector

5.1 General

This chapter specifies the Transceiver and Macrocell Tester's (T&MT) mechanical and electrical interface for testing the ULPI PHY. The T&MT is typically a ULPI Link FPGA/ASIC board, and the PHY daughter-card (UUT) is plugged onto this.

The daughter-card and placement of the interfacing connector are shown in this chapter. The connector pin assignment is described here using the naming convention of the UTMI+ Low Pin Interface and the original T&MT specification [Ref 7].

The ULPI T&MT interface has the following features:

- 100 pin Amp connector.
- Maintains compatibility with UTMI and UTMI+ T&MT connector pin assignments where possible.
- Routes all ULPI digital interface pins between Link and PHY.
- Provides 15 general-purpose I/O pins for vendor-specific functions.
- Multiple power and ground pins.
- Switchable VBUS power control.
- Defines separate I/O voltage pins. I/O voltage is vendor-specific. Typical I/O voltages include 3.3 ±0.3V and 1.8 ±0.15V.

5.2 Daughter-card (UUT) Specification

The PHY IC resides on the daughter-card. Daughter-card dimensions are given in the T&MT specification [Ref 7]. An example daughter-card and PCI-based Link board is depicted in Figure 60~~Figure 54~~. Note the orientation of the 100 pin Amp connector and the USB connector.

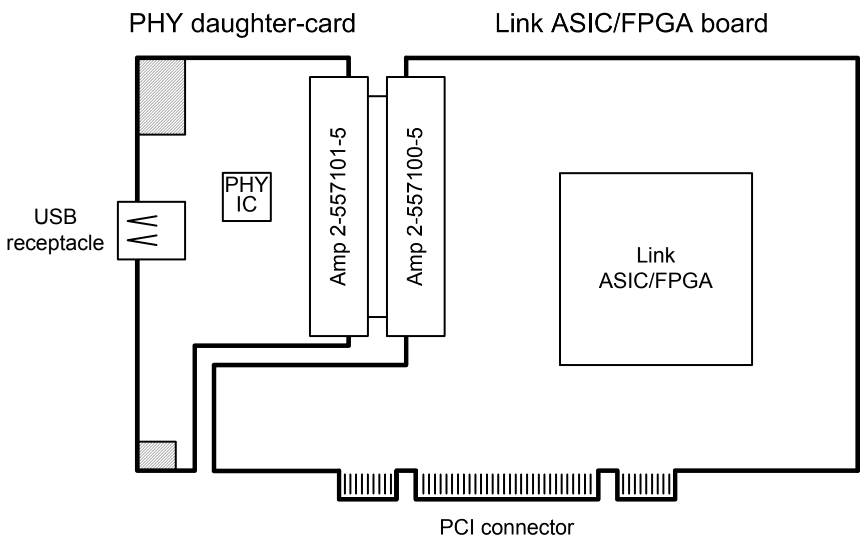

Figure 60 – Recommended daughter-card configuration (not to scale)

With the daughter-card viewed from the backside of the Amp connector (USB receptacle side), and with the Amp connector at the bottom, the connector pin numbers are shown in Table 42~~Table 35~~. Pins shown in blue are ground pins, red indicates power pins, and green indicates I/O power pins.

49	47	45	43	41	39	37	35	33	31	29	27	25	23	21	19	17	15	13	11	9	7	5	3	1
50	48	46	44	42	40	38	36	34	32	30	28	26	24	22	20	18	16	14	12	10	8	6	4	2
99	97	95	93	91	89	87	85	83	81	79	77	75	73	71	69	67	65	63	61	59	57	55	53	51
100	98	96	94	92	90	88	86	84	82	80	78	76	74	72	70	68	66	64	62	60	58	56	54	52

Table 42~~41~~ – T&MT connector pin view

The Amp connector on the daughter-card has the pin assignment shown in Table 43~~Table 36~~. A description of each pin is given in Table 44~~Table 37~~.

1	GPIO0	26	GPIO1	51	GND	76	GND
2	GND	27	GND	52	sys_clk	77	Reserved
3	GPIO2	28	VBUS_out	53	GND	78	GPIO14
4	GND	29	GPIO3	54	GND	79	Reserved
5	GPIO4	30	Reserved	55	Reserved	80	GND
6	GPIO5	31	data7	56	GPIO6	81	VIO
7	GPIO7	32	GND	57	VDD	82	data6
8	VDD	33	data5	58	GPIO8	83	data4
9	GND	34	data3	59	Reserved	84	GND
10	Reserved	35	GND	60	Reserved	85	data2
11	GPIO9	36	data1	61	GPIO10	86	data0
12	GPIO11	37	Reserved	62	GND	87	VIO
13	GND	38	VIO	63	GPIO12	88	Reserved
14	Reserved	39	GND	64	Reserved	89	Reserved
15	SPKR_L	40	Reserved	65	GND	90	clock
16	VDD	41	Reserved	66	Reserved	91	Reserved
17	reset	42	Reserved	67	Reserved	92	GND
18	Reserved	43	GND	68	GND	93	Reserved
19	Reserved	44	Reserved	69	VDD	94	Reserved
20	Reserved	45	SPKR_MIC	70	dir	95	GND
21	GND	46	GND	71	nxt	96	stp
22	Reserved	47	VBUS_in	72	Reserved	97	Reserved
23	Reserved	48	Reserved	73	GND	98	Reserved
24	GND	49	dc_psnt_n	74	Reserved	99	Reserved
25	Reserved	50	GPIO13	75	Reserved	100	psu_shd_n

Table 43~~42~~ – T&MT connector pin allocation

Name	Direction on UUT	Active Level	Description
clock	I/O	n/a	Clock pin. Can be output from the PHY and/or input from the Link.
data0-7	I/O	n/a	ULPI bi-directional **data** bus.
dir	Output	High	ULPI **dir** signal.
stp	Input	n/a	ULPI **stp** signal.
nxt	Output	n/a	ULPI **nxt** signal.
reset	Input	High	Optional reset pin.
dc_psnt_n	Output	Low	When low, the Link detects the presence of the daughter-card. A pull-down resistor of 200Ω is required
psu_shd_n	Input	Low	When low, the PHY power supply unit is shutdown and can be sourced from the Link through the T&MT connector pins.
sys_clk	Input	n/a	Optional vendor-specific input clock.
Reserved	I/O	n/a	Reserved.
GPIO0-14	I/O	n/a	General Purpose I/O. Spare interconnects between the UUT and T&MT.
GND	I/O	n/a	Ground plane.
VBUS_in	I/O	n/a	Used by the Link to monitor VBUS pin of the USB connector.
VBUS_out	I/O	n/a	5 Volt supply from the Link. Used primarily by the PHY for driving VBUS. The Link controls VBUS using the ***DrvVbusExternal*** register bit in the PHY.
VDD	I/O	n/a	PHY power supply.
VIO	I/O	n/a	Vendor-specific pad ring power supply. The voltage on VIO determines the switching thresholds of the ULPI interface signals.
SPKR_L	Input	n/a	In carkit mono mode, inputs mono speaker signal. In carkit stereo mode inputs left speaker signal.
SPKR_MIC	I/O	n/a	In carkit mono mode, outputs microphone signal. In carkit stereo mode, inputs right speaker signal.

Table 44~~43~~ – T&MT pin description