NET2272

Local Bus to USB 2.0 Peripheral Controller

Patent Pending

For Revision 1A

Data Book

Version	1.91
August	**2005**
Website	www.plxtech.com
Technical Support	www.plxtech.com/support/
Phone	408 774-9060
	800 759-3735
FAX	408 774-2169

Copyright Information

PLX Technology provides this documentation without warranty, term or condition of any kind, either express or implied, including, but not limited to, express and implied warranties of merchantability, fitness for a particular purpose, and non-infringement. While the information contained herein is believed to be accurate, such information is preliminary, and no representations or warranties of accuracy or completeness are made. In no event will PLX Technology be liable for damages arising directly or indirectly from any use of or reliance upon the information contained in this document. PLX Technology may make improvements or changes in the product(s) and/or the program(s) described in this documentation at any time.

PLX Technology retains the right to make changes to this product at any time, without notice. Products may have minor variations to this publication, known as errata. PLX Technology assumes no liability whatsoever, including infringement of any patent or copyright, for sale and use of PLX Technology products.

Revision History

Version	Date	Description of Changes
1.0	May 5, 2003	Silicon Initial Release.
1.1	October 7, 2003	Silicon Release.
1.2	October 15, 2003	Power Consumption Update.
1.3	June 15, 2004	Minor Clarifications.
1.4	February 17, 2005	Increased DMA Burst Write Recovery time. Added FBGA package.
1.5	March 18, 2005	Added 2.5V I/O Electrical Specifications.
1.6	April 13, 2005	Changed temperature references from Industrial to Commercial (includes change to BGA ordering number). Moved NAND Tree Test to new Chapter 10. Reorganized Chapter 11, Electrical Specifications, and created separate set of 2.5V and 3.3V VDDIO Test Condition tables. Applied miscellaneous corrections and changes for readability and consistency.
1.7	May 27, 2005	Added 1.8V I/O Electrical Specifications. Reorganized Chapter 11, Electrical Specifications to reflect the *USB r2.0* specification table sequences. Moved Physical Pin Assignment figures from Chapter 2 to Chapter 12. Applied miscellaneous corrections and changes for readability and consistency.
1.8	June 10, 2005	Corrected 1.8V-related content. Replaced Chapter 11 DC Specification.
1.9	August 10, 2005	Revised crystal Note in Section 1.4.1. Updated VDDIO Local Bus AC Specifications for all three voltages. Added ordering p/n explanation to Appendix A.
1.91	August 18, 2005	Moved the Physical Pin and Ball assignment diagrams from Chapter 12 to Chapter 2. Appended “at 3.3V VDDIO” to title of Sections 4.5.5 and 4.6.6. Removed references to other voltages in note preceding table in Section 4.6.6.

Preface

The information contained in this document is subject to change without notice. This PLX Document to be periodically updated as new information is made available.

Scope

This document describes the NET2272 USB 2.0 Peripheral Controller operation and provides operational data for customer use. It also provides functional details of the PLX Technology NET2272 for hardware designers and software/firmware engineers. This data book assumes that the reader has access to and is familiar with the documents referenced in the following section.

Supplemental Documentation

USB Implementers Forum
5440 SW Westgate Drive #217, Portland, OR 97221 USA
Tel: 503 296-9892, Fax: 503 297-1090, www.usb.org
Universal Serial Bus Specification Revision 2.0 (*USB r2.0*)

Terminology

Term	Definition
Big Endian	Most significant byte (MSB) in a scalar is located at Address 0.
Byte	8-bit data quantity.
Clock cycle	One period of the internal 60-MHz clock.
Little Endian	Least significant byte (LSB) in a scalar is located at Address 0.
Local transaction	Read or Write operation on the Local Bus. Includes an Address phase, followed by one Data transfer.
Local transfer	During a *transfer*, data is moved from the source to the destination on the Local Bus.
Scalar	Multi-byte data element.
Target	Device that responds to a transaction initiated by an external CPU or DMA controller.
Word	16-bit data quantity.

Contents

Registers

Chapter 1 Introduction

1.1 Features

- *Universal Serial Bus Specification r2.0* (*USB r2.0*) compliant
- USB full (12 Mbps) and high (480 Mbps) speeds
- Three Configurable Physical Endpoints, in addition to Endpoint 0
- Supports 30 Configurable Virtual endpoints
- Configurable endpoints can be Isochronous, Bulk, or Interrupt, as well as IN or OUT
- High-Bandwidth Isochronous mode
- Maximum Packet Size up to 1 KB, double-buffered
- Internal 3-KB memory provides Transmit and Receive buffers
- Local CPU Bus easily interfaces to generic CPUs
- 8- or 16-bit CPU or DMA Bus transfers
- Optional DMA Split Bus mode, with dedicated DMA and CPU access
- Multiple register Address modes support direct and indirect register addressing
- Automatic Retry of failed packets
- Diagnostic register allows forced USB errors
- Software-controlled disconnect allows re-enumeration
- Atomic operation to set and clear Status bits simplifies software
- 30-MHz oscillator with internal Phase-Locked Loop (PLL) multiplier
- Output clock to Local Bus – Eight programmable frequencies, from OFF to 60 MHz
- 1.8V, 2.5V, and 3.3V operating voltages, 5V tolerant I/O
- Low-power CMOS technology in 64-Pin Plastic TQFP or 64-Ball FBGA package
- Lead-Free and RoHS (Reduction of Hazardous Substances) compliant

1.2 Overview

The NET2272 USB Peripheral Controller allows *Control*, *Isochronous*, *Bulk*, and *Interrupt* transfers between a Local Bus and Universal Serial Bus (USB). The NET2272 supports the device side of a connection between a USB host computer and intelligent peripherals, *such as* printers, image scanners, and digital cameras.

The six main modules are as follows:

- USB Transceiver
- Serial Interface Engine (SIE)
- USB Protocol Controller
- Endpoint Packet Buffers
- Local Bus Interface
- Configuration Registers

Each module is briefly described in the sections that follow.

1.2.1 USB Transceiver

- Supports Full- (12 Mbps) or High-Speed (480 Mbps) operation
- Serial Data transmitter and receiver
- Parallel Data interface to Serial Interface Engine (SIE)
- Single parallel Data Clock output with on-chip PLL to generate High-Speed Serial Data clocks
- Data and clock recovery from USB Serial Data stream
- SYNC/End-Of-Packet (SYNC/EOP) generation and checking
- Bit-stuffing/unstuffing; Bit-Stuff error detection
- Logic to facilitate Resume signaling
- Logic to facilitate Wakeup and Suspend detection
- Ability to switch between Full- and High-Speed terminations and signaling

1.2.2 Serial Interface Engine (SIE)

- Interface between Packet buffers and USB transceiver
- CRC generator and checker
- Packet Identifier (PID) decoder
- Forced error conditions
- *USB r2.0* Test modes

1.2.3 USB Protocol Controller

- Host-to-device communication
- Automatic Retry of failed packets
- Up to three Isochronous, Bulk, or Interrupt endpoints, each with a Configurable Packet buffer
- Supports up to 30 Virtual endpoints, with the ability to be mapped to Physical endpoints
- Configurable Control Endpoint 0
- Interface to Packet buffers
- Software-controlled disconnect signaling allows device re-enumeration
- Software-controlled USB Suspend and Root-Port Reset detection
- Software-controlled Device-Remote Wakeup
- Software-controlled Root-Port Wakeup

1.2.4 Endpoint Packet Buffers

- Choice of four preset configurations to simplify programming
- Separate 128-Byte Packet buffer for Physical Endpoint 0
- 3 KB of configurable Packet Buffer memory for Physical Endpoints A, B, and C
- Maximum Packet Size up to 1 KB, double-buffered

1.2.5 Local Bus Interface

- Target interfaces to 8- or 16-bit CPU
- Access to internal Transmit and Receive Packet buffers
- DMA Split Bus transactions (DMA and CPU on separate Data buses)
- DMA Burst mode
- Supports DMA and Interrupt transfers
- Optional Multiplexed Address/Data bus, using ALE for low pin/ball-count applications
- Indirect addressing, allowing access to all registers with only a single Address bit
- 1.8V, 2.5V, and 3.3V operating voltages, 5V tolerant I/O

1.2.6 Configuration Registers

- Internal registers accessible from Local Bus
- Main Control registers for common functions
- USB Control registers for USB Protocol Controller module
- Control registers for each endpoint

1.3 NET2272 Block Diagrams

Figure 1-1. NET2272 Block Diagram

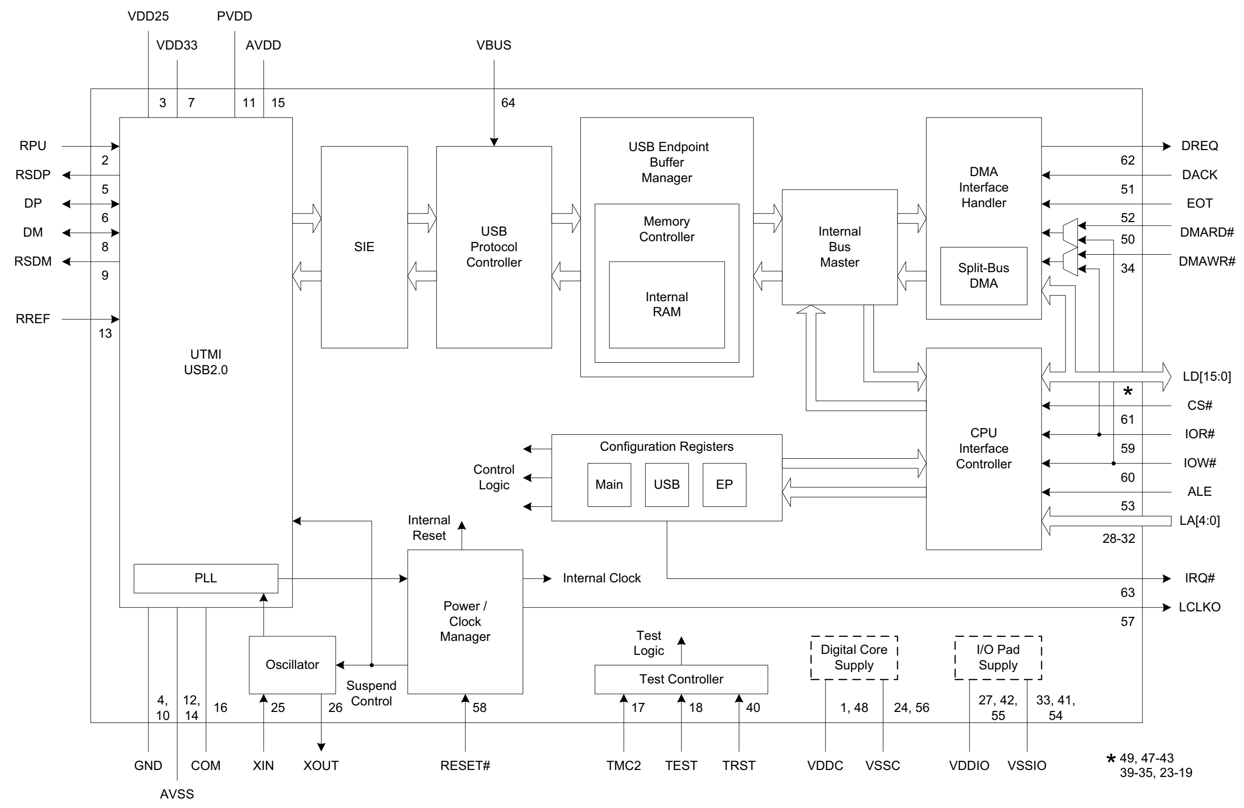
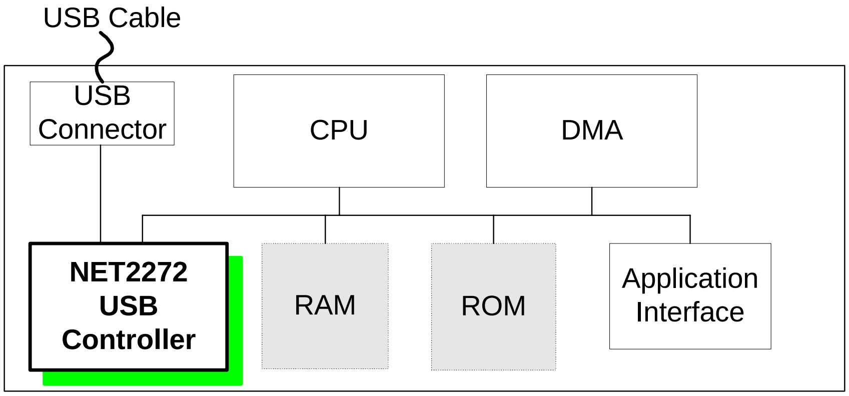
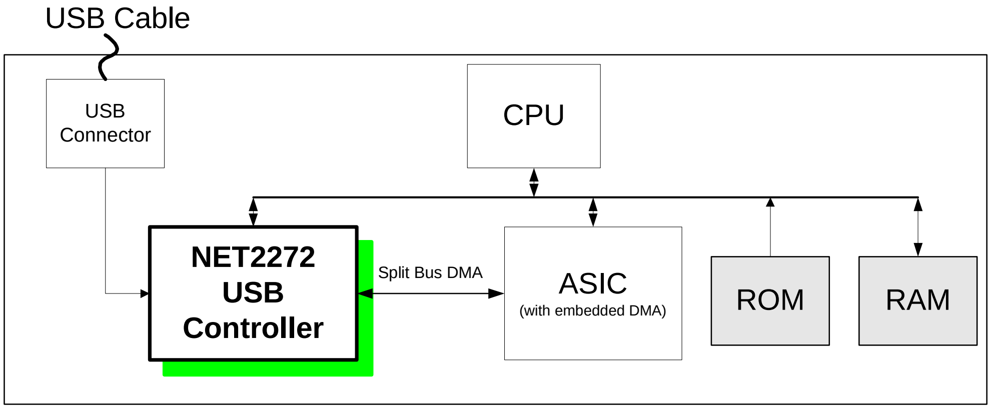


Figure 1-2. CPU-Based Device Controller Diagram



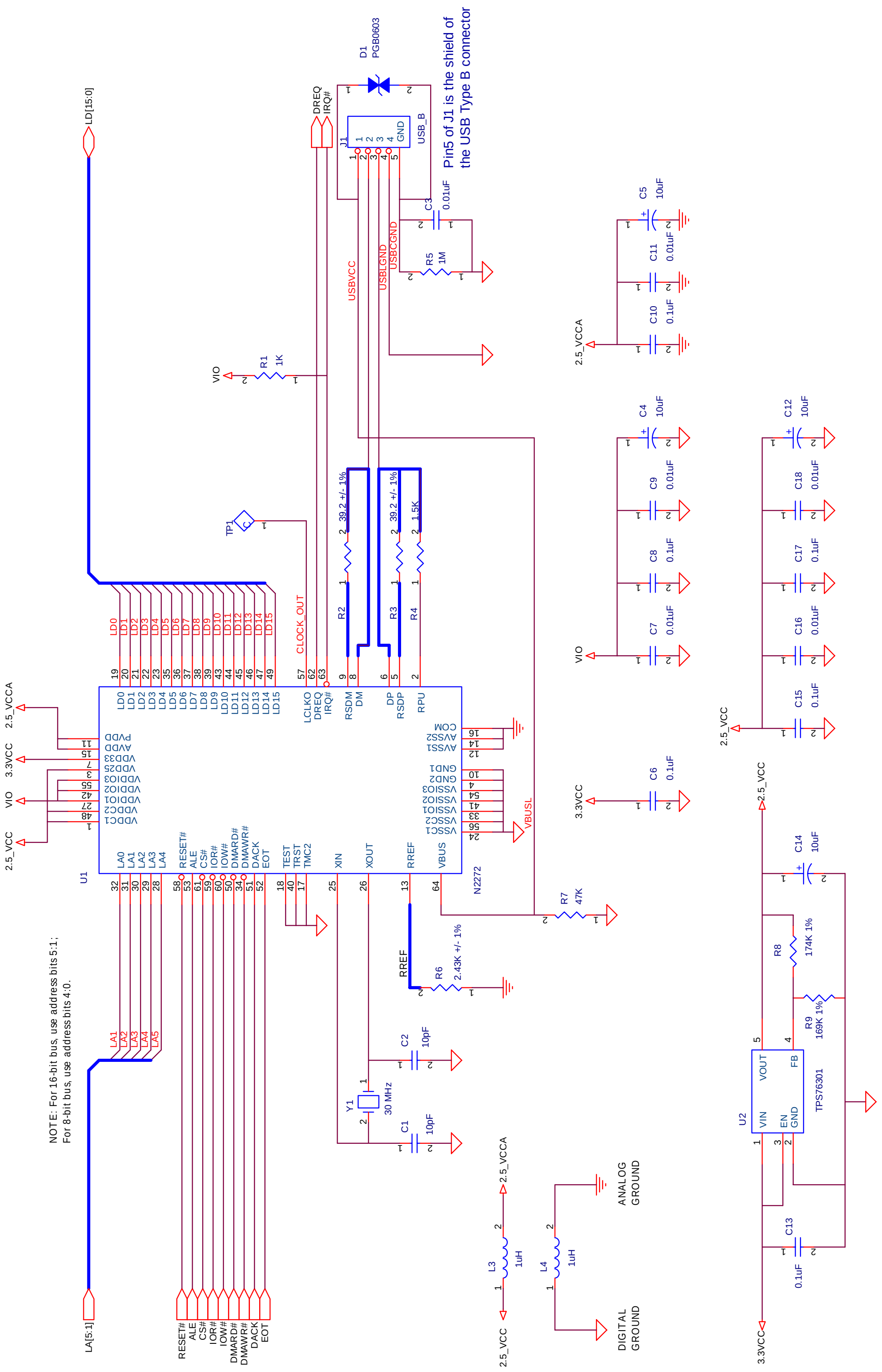
CPU-based Device Controller

Figure 1-3. ASIC with DMA Split Bus Diagram



1.4 Connections to NET2272

Figure 1-4. Sample Connections to NET2272



NOTE: For 16-bit bus, use address bits 5:1;
For 8-bit bus, use address bits 4:0.

1.4.1 Third-Party Part Numbers

Table 1-1. Third-Party Part Numbers

Part	Manufacturer	Part Number	Website
30 MHz Fundamental Crystal (Y1)	KDS Daishinku Corp.	AT-49 30.000-16	www.kdsj.co.jp/english.html
1 µH Inductor, 10%, 0805 Package (L3-L4)	Taiyo Yuden	LK21251R0K	www.t-yuden.com/inductors/index.cfm
ESD Suppressor, 0603 Package (D1)	Littelfuse	PGB0010603MR	www.littelfuse.com/data/en/Data_Sheets/PGB0603.pdf
2.43K, 1% resistor, 0.1W, 0603 Package (R6)	Panasonic	ERJ6ENF2431V	www.panasonic.com/industrial/components/pdf/002_er13_erj_2r_3r_6r_3e_6e_8e_14_12_dne.pdf
USB B Connector	Newnex	URB-1001	www.newnex.com

Note: The crystal should maintain a tolerance of ±0.01% (100 ppm) to guarantee a 480 Mbps ±500 ppm data rate.

1.4.2 General PCB Layout Guidelines

The *USB r2.0* high-speed 480 Mbps Data transfers utilize 400 mV differential signaling. This requires special printed circuit board (PCB) layout requirements. (Refer to the Intel USB layout guidelines.)

Warning: The following guidelines must be observed to ensure proper NET2272 operation. It is strongly recommended that schematics and PCB layout be submitted to PLX Technical Support for review prior to PCB fabrication.

1.4.2.1 USB Differential Signals

- Consult with board manufacturer to determine layer separation, trace width, and trace separation for maintaining 90-Ohm differential impedance.
- Maintain equal trace lengths for D+ and D-.
- Minimize number of *vias* and *curves* on D+ and D- traces.
- Use two 45° turns, instead of one 90° turn.
- Minimize trace lengths shown in **bold** in the schematic provided in Figure 1-4.
- Prevent D+ and D- traces from crossing a power-plane void. The same ground layer to remain next to the D+ and D- traces.
- Place Digital Ground (VSSC and VSSIO) layer next to the layer where D+ and D- are routed.
- Avoid using stubs or test points for observing USB signals.
- Maximize the distance of D+ and D- from other signals, to prevent crosstalk.

1.4.2.2 Analog VDDC and VDDIO (Power)

- Analog power must be filtered from the digital power, using the provided example circuit (Figure 1-4).
- Analog and digital VDDC and VDDIO must be connected by way of a 1 µH inductor.
- Analog VDDC and VDDIO must be separated from digital VDDC and VDDIO. If analog and digital VDDC and VDDIO are in the same layer, split the layer to accommodate the two Power signals.
- Connect the AVDD and PVDD pins/balls to analog VDDC and VDDIO.

1.4.2.3 Analog VSSC and VSSIO (Ground)

- Analog Ground must be filtered from the digital Ground, using the provided example circuit (Figure 1-4).
- Analog and digital VSSC and VSSIO must be connected by way of a 1 µH inductor.
- Connect the AVSS and COM pins/balls and the RREF pin/ball's resistor to analog VSSC and VSSIO.

1.4.2.4 Decoupling Capacitors

- For every two pairs of digital/analog VDDC and VDDIO and VSSC and VSSIO, locate at least one 0.1 µF and one 0.01 µF decoupling capacitor near the NET2272 device.
- Decoupling capacitors can be placed on either side of the PCB.
- Provide at least one 10 µF decoupling capacitor for every five 0.1 µF or 0.01 µF decoupling capacitors.
- Use capacitors with good quality at high frequency for low equivalent series resistance (ESR), *such as* ***tantalum*** or ***ceramic*** capacitors. Do ***not*** use electrolytic capacitors.

1.4.2.5 EMI Noise Suppression

- A common-mode choke coil can effectively suppress electromagnetic interference (EMI) noise, although such a coil can affect the *USB r2.0* signal quality.
- Use a good quality noise filter, if necessary.
- For typical implementation, a choke is not required.
- Use quality, shielded cables.

Chapter 2 Pin/Ball Description

2.1 Pin/Ball Description Abbreviations

Table 2-1. Pin/Ball Description Abbreviations

Abbreviation	Description
I	Input
O	Output
I/O	Bi-Directional
S	Schmitt Trigger
TS	Three-State
TP	Totem Pole
OD	Open Drain
#	Active Low

Note: *Input pins/balls that do not have an internal pull-up nor pull-down resistor must be externally driven when the NET2272 is in the Low-Power Suspend state.*

2.2 Packaging Information

For NET2272 packaging information, refer to Chapter 12, “Mechanical Specifications.”

2.3 Pin/Ball Descriptions (TQFP and FBGA Packages)

2.3.1 Digital Power and Ground Pins/Balls

Table 2-2. Digital Power and Ground (10 Pins/Balls)

Signal Name	Pins/Balls		Type	Description
	TQFP	FBGA		
VDDC	1, 48	A7, H8	Power	**Digital Core Supply Voltage** Connect to 2.5V supply.
VDDIO	27, 42, 55	D7, E3, F5	Power	**I/O Interface Supply Voltage** Connect to voltage between 1.8V and 3.3V.
VSSC	24, 56	D2, D8	Ground	**Digital Core Ground** Connect to Ground.
VSSIO	33, 41, 54	F8, H1, H4	Ground	**I/O Interface Ground** Connect to Ground.

2.3.2 USB Transceiver Pins/Balls

Table 2-3. USB Transceiver (15 Pins/Balls)

Signal Name	Pins/Balls		Type	Description
	TQFP	FBGA		
AVDD	15	C3	Power	**Analog Supply Voltage** Connect to analog 2.5V supply.
AVSS	12, 14	B2, B6	Ground	**Analog Ground** Connect to analog Ground.
COM (AVSS)	16	A2	Ground	**PLL Ground** Connect to analog Ground.
DM	8	B3	I/O	**High-Speed USB Negative Data Port** DM is the High-Speed Negative Differential Data signal of the USB Data port. DM also acts as the Full-Speed Negative Differential Input Data port. DM connects directly to the USB connector.
DP	6	B5	I/O	**High-Speed USB Positive Data Port** DP is the High-Speed Positive Differential Data signal of the USB Data port. DP also acts as the Full-Speed Positive Differential Input Data port. DP connects directly to the USB connector.
GND	4, 10	A4, A6	Ground	**Digital Ground** Connect to Ground.
PVDD	11	C4	Power	**PLL Supply Voltage** Connect to analog 2.5V supply.
RPU	2	D5	O	**DP Pull-Up Resistor** Connect to the DP pin/ball through a 1.5K-Ohm resistor.
RREF	13	D4	I	**Reference Resistor** Connect 2.43K-Ohm ±1% resistor to analog Ground. Typical voltage on this pin/ball is 1.27V.
RSDM	9	A3	O	**Full-Speed USB Negative Output Data Port** RSDM is the Full-Speed Negative Differential Output Data signal of the USB Data port. Connect through a 39.2-Ohm ±1% resistor to the USB connector.
RSDP	5	A5	O	**Full-Speed USB Positive Output Data Port** RSDP is the Full-Speed Positive Differential Output Data signal of the USB Data port. Connect through a 39.2-Ohm ±1% resistor to the USB connector.
VDD25	3	C5	Power	**Supply Voltage** Connect to digital 2.5V supply.
VDD33	7	B4	Power	**Supply Voltage** Connect to digital 3.3V supply.

2.3.3 Clock, Reset, and Miscellaneous Pins/Balls

Table 2-4. Clock, Reset, and Miscellaneous (8 Pins/Balls)

Signal Name	Pins/Balls		Type	Description
	TQFP	FBGA		
LCLKO	57	E6	O 12 mA TS	**Local Clock Output** Buffered Clock output from the internal PLL, with the frequency depending on the **LOCCTL** register *Local Clock Output* field state. LCLKO stops oscillating when the NET2272 is placed in the Low-Power Suspend state. LCLKO is not driven while the NET2272 is suspended. When the internal oscillator is started, LCLKO is prevented from driving for 2 ms. LCLKO does not oscillate while the NET2272 in Power-Down mode.
RESET#	58	C6	I S	**External Reset** Connect to Local or Power-On Reset. To reset when the oscillator is stopped (initial power-up or in the Low-Power Suspend state), assert for at least 2 ms. When oscillator is running, assert for at least five 60-MHz clock periods.
TEST	18	E4	I	**Test Input** Connect to Ground for normal operation. To enable NAND tree, set TEST and LA[4:3] high. (Refer to Chapter 10, “NAND Tree Test,” for further details.)
TMC2	17	E5	I	**TMC2 Test Input. I/O Buffer Control** Connect to Ground for normal operation.
TRST	40	G5	I	**TRST Test Input. Test Access Port (TAP) Controller Reset** Connect to Ground for normal operation.
VBUS	64	B8	I S	**USB VBUS** VBUS indicates when the NET2272 is connected to a powered-up USB host connector. Connect a 47K-Ohm pull-down resistor to VBUS to hold VBUS low when not connected to the USB.
XIN	25	E1	I	**Oscillator Input** Connect to 30-MHz crystal or external oscillator module.
XOUT	26	D1	O	**Oscillator Output** Connect to crystal, or leave open when using an external oscillator module. The oscillator stops when the NET2272 is in the Low-Power Suspend state.

2.3.4 Local Bus Pins/Balls

Table 2-5. Local Bus (31 Pins/Balls)

Signal Name	Pins/Balls		Type	Description
	TQFP	FBGA		
ALE	53	E8	I	**Address Latch Enable** When operating in Multiplexed Address and Data mode, the 5-bit Address bus is latched on the trailing (negative) edge of ALE. The NET2272 automatically detects ALE use to indicate multiplexed address and data use on the LD[4:0] pins/balls. Connect to ground when operating in Non-Multiplexed register addressing modes.
CS#	61	C7	I	**Chip Select** CS# enables access to registers within the NET2272. Asserting CS# during the Low-Power Suspend state wakes up the NET2272. Asserting CS# during RESET# holds the NET2272 in the Low-Power Suspend state by disabling the internal oscillator.
DACK	51	F7	I	**DMA Acknowledge** DACK is used to transfer data to and from the Packet buffer in response to DREQ. DACK is ignored unless the **LOCCTL1** register *DMA DACK Enable* bit is set. DACK polarity is programmable.
DMARD#	50	G7	I	**DMA Read Strobe** The DMA Bus master asserts DMARD# during a DMA Read transaction when DMA Split Bus mode is selected
DMAWR#	34	H3	I	**DMA Write Strobe** The DMA Bus master asserts DMAWR# during a DMA Write transaction when DMA Split Bus mode is selected.
DREQ	62	A8	O 3 mA TS	**DMA Request** DREQ requests DMA transfers from an external DMA controller. DREQ floats when the USB host suspends the NET2272. DREQ polarity is programmable.
EOT	52	E7	I	**End of Transfer** EOT (from an external DMA controller) is used to terminate a DMA transfer. The current word is transferred; however, no additional transfers are requested. EOT can be programmed to cause an interrupt. EOT polarity is programmable.
IOR#	59	C8	I	**Read Strobe** The Local Bus master asserts IOR# during a Read transaction.
IOW#	60	B7	I	**Write Strobe** The Local Bus master asserts IOW# during a Write transaction.

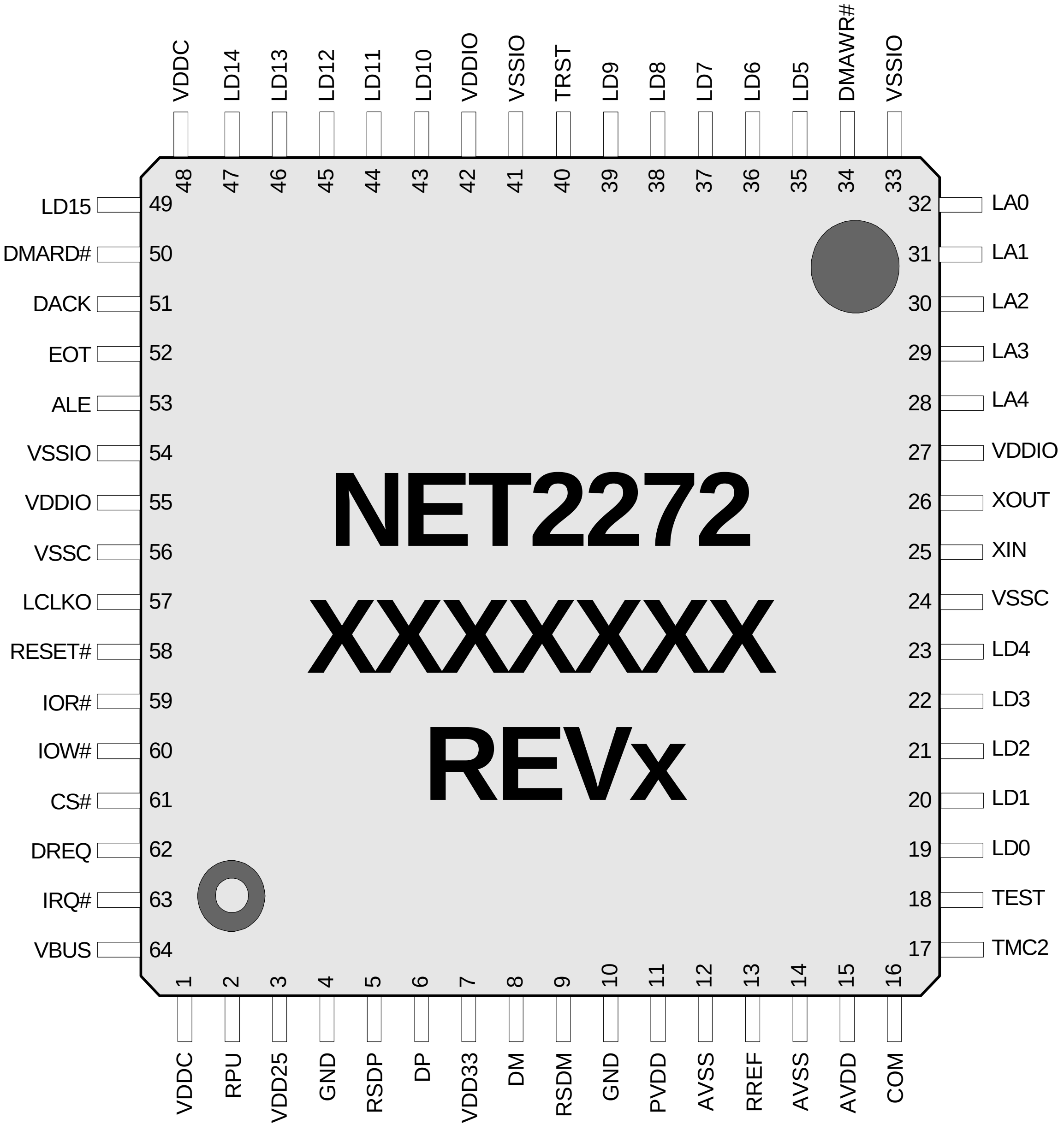
Table 2-5. Local Bus (31 Pins/Balls) *(Cont.)*

Signal Name	Pins/Balls		Type	Description
	TQFP	FBGA		
IRQ#	63	D6	O 12 mA OD	**Interrupt Request Output** IRQ# interrupts the local processor, based on events selected in the internal program registers. Because IRQ# is an open-drain pin/ball, an external 1K-Ohm pull-up resistor is required.
LA[4:0]	28, 29, 30, 31, 32	F1, E2, F2, G2, G1	I	**Address Bus** Five Address bits can directly address most NET2272 internal registers (Non-Multiplexed Direct Address mode). A minimum of one Address bit is required to operate the NET2272, using an optional register Non-Multiplexed Indirect Address mode. A third addressing mode (Multiplexed Address and Data) uses ALE with LD[4:0] to provide five bits of register addressing. (Refer to Chapter 4, "Local Bus Interface," for further details.) To enable NAND tree, set LA[4:3] and TEST high. (Refer to Chapter 10, "NAND Tree Test," for further details.)
LD[15:0]	49, 47, 46, 45, 44, 43, 39, 38, 37, 36, 35, 23, 22, 21, 20, 19	G8, H7, H6, F6, H5, G6, F4, G4, F3, G3, H2, D3, C1, C2, B1, A1	I/O 6 mA	**Data Bus** LD[15:0] serve as the Local CPU I/O Data bus. In Multiplexed Address and Data mode, ALE can be used with LD[4:0] to provide five Address bits on the falling edge of ALE. In 16-Bit mode, the Data bus is 16 bits wide. (Refer to Section 4.2.3, "Multiplexed Address and Data Mode," for further details.)

2.4 Physical Pin/Ball Assignment

2.4.1 64-Pin Plastic TQFP

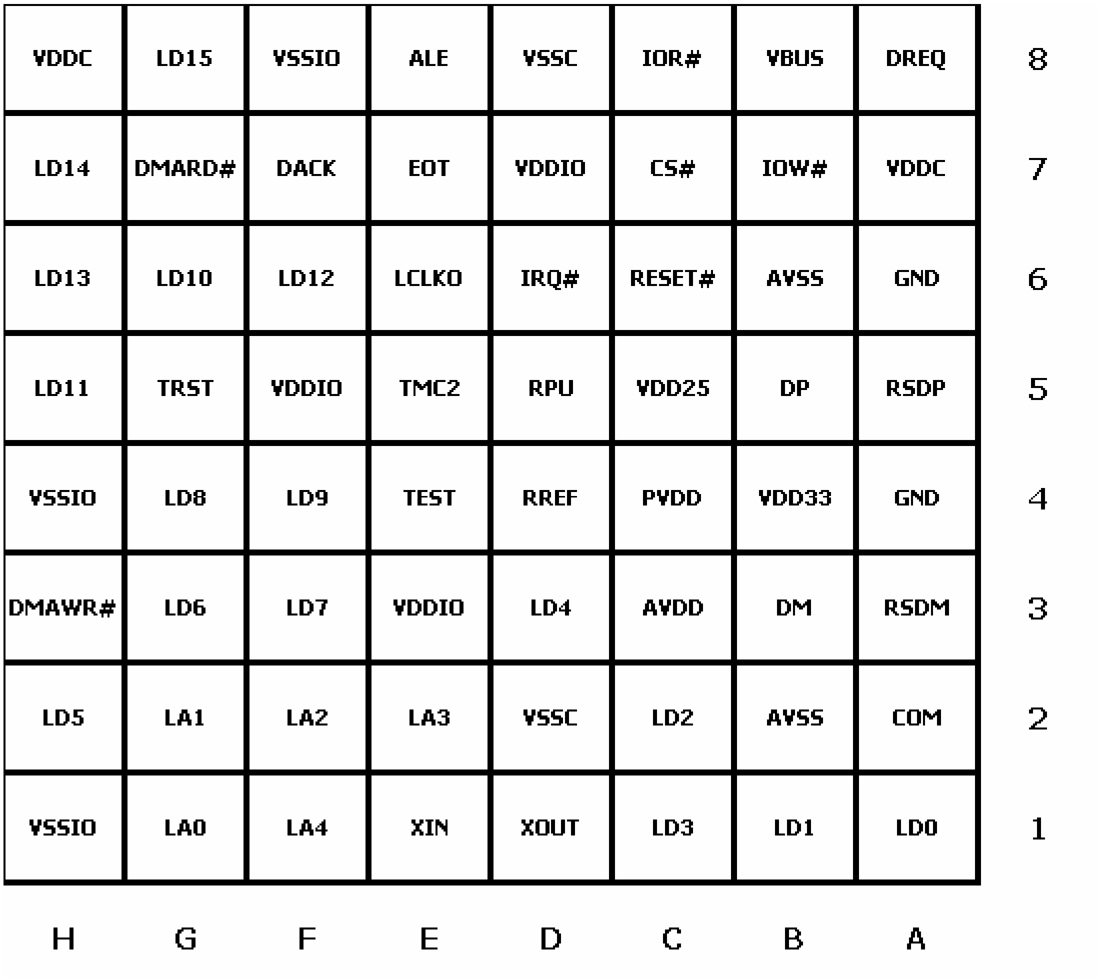
Figure 2-1. Physical Pin Assignment (64-Pin Plastic TQFP Package)



Note: *This drawing is for informational purposes only. Contact PLX for additional chip marking, PCB layout, and manufacturing information.*

2.4.2 64-Ball Plastic FBGA

Figure 2-2. Physical Ball Assignment (64-Ball Plastic FBGA Package, Underside View)



VDDC	LD15	VSSIO	ALE	VSSC	IOR#	VBUS	DREQ	8
LD14	DMARD#	DACK	EOT	VDDIO	CS#	IOW#	VDDC	7
LD13	LD10	LD12	LCLKO	IRQ#	RESET#	AVSS	GND	6
LD11	TRST	VDDIO	TMC2	RPU	VDD25	DP	RSDP	5
VSSIO	LD8	LD9	TEST	RREF	PVDD	VDD33	GND	4
DMAWR#	LD6	LD7	VDDIO	LD4	AVDD	DM	RSDM	3
LD5	LA1	LA2	LA3	VSSC	LD2	AVSS	COM	2
VSSIO	LA0	LA4	XIN	XOUT	LD3	LD1	LD0	1
H	G	F	E	D	C	B	A	

Note: *This drawing is for informational purposes only. Contact PLX for additional chip marking, PCB layout, and manufacturing information.*

Chapter 3 Reset and Initialization

3.1 Overview

The NET2272 normal initialization sequence consists of the following:

1. The NET2272 RESET# signal is asserted and de-asserted.
2. Local CPU initializes USB and Local Bus Configuration registers.

3.2 RESET# Pin/Ball

The RESET# pin/ball resets all NET2272 logic to its default state. RESET# is typically connected to a Power-On Reset circuit.

3.3 Root-Port Reset

If the NET2272 detects a single-ended zero on the root port for greater than 2.5 µs, the single-ended zero is interpreted as a Root-Port Reset. Root-Port Reset is recognized only when the VBUS input pin/ball is high, and the **USBCTL0** register *USB Detect Enable* bit is set. The following resources are reset:

- SIE
- USB state machines
- Local state machines
- **OURADDR** register
- Buffer pointers

Root-Port Reset does not affect the remainder of the Configuration registers. The *Root-Port Reset Interrupt* bit is set when a Root-Port Reset is detected. The Local CPU takes appropriate action when this interrupt occurs.

According to the *USB r2.0*, the USB reset width is minimally 10 ms and can be longer, depending on the upstream host or hub. There is no specified maximum USB reset width.

3.4 Reset Summary

Table 3-1 delineates which device resources are reset when either of the two reset sources are asserted.

Table 3-1. Reset Summary

Reset Sources	Device Resources		
	USB, SIE Modules, OURADDR Register	All Configuration Registers	Endpoint Buffer Pointers
RESET#	X	X	X
USB Root-Port Reset	X	–	X

Chapter 4 Local Bus Interface

4.1 Overview

The Local Bus interface allows the NET2272 to easily interface with many generic processors and custom ASIC interfaces. Both Multiplexed and Non-Multiplexed buses are supported.

4.2 Register Addressing Modes

The NET2272 provides three register addressing modes that support various user architectures:

- Non-Multiplexed Direct Address Mode
- Non-Multiplexed Indirect Address Mode
- Multiplexed Address and Data Mode

These addressing modes always remain active and no special effort is required to use them. The modes act on a transaction-by-transaction basis, allowing DMA and CPU to operate with inherently differing bus architectures. *For example*, the CPU can operate with a Multiplexed bus (using ALE to de-multiplex the Address/Data bus), while DMA can operate using a Non-Multiplexed Data bus.

4.2.1 Non-Multiplexed Direct Address Mode

Non-Multiplexed Direct Address mode uses LA[4:0] to directly access the first 32 Configuration registers.

4.2.2 Non-Multiplexed Indirect Address Mode

Non-Multiplexed Indirect Address mode uses the **REGADDRPTR** and **REGDATA** registers (Addresses 00h and 01h, respectively) to provide a Command/Data interface to the NET2272 internal registers and buffers. All CPU transactions performed with **REGDATA** have their address sourced by **REGADDRPTR**. The Local CPU first programs **REGADDRPTR** with the needed register address, then reads or writes to **REGDATA** with the data intended for the register pointed to by **REGADDRPTR**. This addressing mode requires only one physical Address bit (to access Address 00h or 01h). All NET2272 unused Address bits must be connected to Ground. When all five Address bits are being used, this addressing mode allows access to registers above Address 1Fh.

4.2.3 Multiplexed Address and Data Mode

Multiplexed Address and Data mode uses the ALE pin/ball to de-multiplex Data bus addresses. The NET2272 automatically detects ALE use, and utilizes the address represented by LD[4:0] on the falling edge of ALE as the current Transaction address. This addressing mode is supported by several common microcontrollers. ALE is grounded when this mode is not used.

4.3 Control Signal Definitions

Control signals direct the flow of data across the Local Bus. A Write transaction is performed by asserting CS# and IOW#. The address and data must be valid on the trailing (rising) edge of IOW#. A Read transaction is performed by asserting CS# and IOR#.

4.4 Bus Width and Byte Alignment

The Local Bus supports 8- or 16-bit buses. In 8-Bit mode, all Configuration registers and buffers are accessed one byte at a time. A typical 8-bit application connects the CPU Address bits A[4:0] to the NET2272 Address bus LA[4:0].

In 16-Bit mode, the Configuration registers remain accessed one byte at a time; however, the buffers are accessed one word at a time. The **LOCCTL** Configuration register *Byte Swap* bit determines whether the bytes are swapped as they are written into the buffer in 16-Bit mode. This allows for connections to Little or Big Endian processors. A typical 16-bit application connects the CPU Address bits A[5:1] to the NET2272 Address bus LA[4:0].

4.5 I/O Transactions

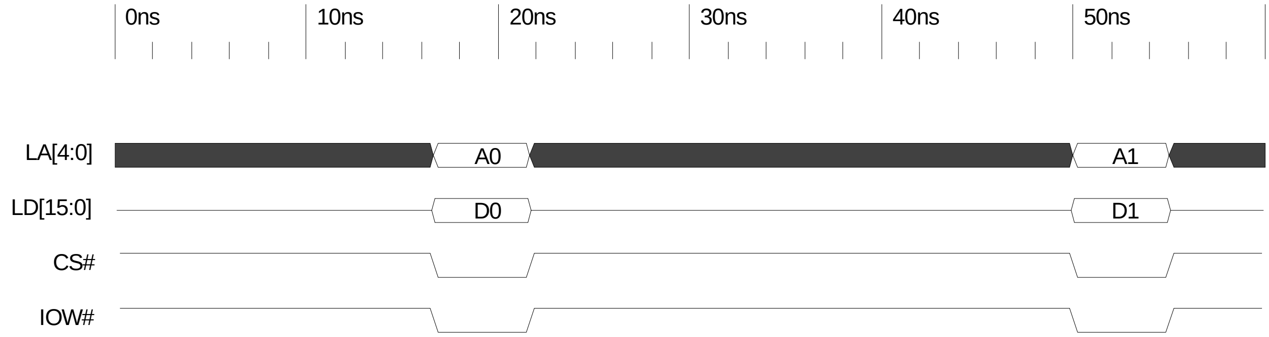
I/O transactions are those in which a CPU on the Local Bus accesses registers or Packet buffers within the NET2272.

4.5.1 Non-Multiplexed I/O Write

Non-Multiplexed I/O writes begin when both CS# and IOW# are asserted. The Address and Write data must meet the setup time, with respect to the Write transaction end. Data is written into the register or Packet buffer when CS# or IOW# is de-asserted. A new I/O Write transaction cannot start until the T15A recovery time expires, and a new I/O Read transaction cannot start until the T15B recovery time expires. ALE must be held low for Non-Multiplexed mode.

Note: *For 1.8, 2.5, and 3.3V VDDIO setup and T timing values, refer to Table 11-29, Table 11-39, and Table 11-49, respectively.*

Figure 4-1. Non-Multiplexed I/O Write Timing

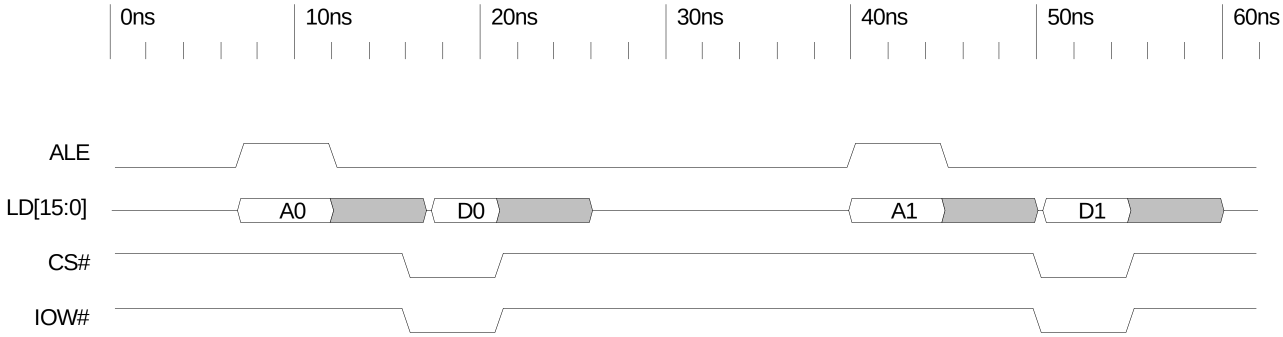


4.5.2 Multiplexed I/O Write

Multiplexed I/O Write transactions are started when the address is driven onto the lower bits of the Data bus, and ALE is pulsed. After the address is latched into the NET2272, the Data phase is initiated with CS# and IOW# assertion. The Write data must meet the setup time, with respect to the Write cycle end. Data is written into the register or Packet buffer when CS# or IOW# is de-asserted. A new I/O Write transaction cannot start until the T15A recovery time expires, and a new I/O Read transaction cannot start until the T15B recovery time expires.

Note: *For 1.8, 2.5, and 3.3V VDDIO setup and T timing values, refer to Table 11-30, Table 11-40, and Table 11-50, respectively.*

Figure 4-2. Multiplexed I/O Write Timing

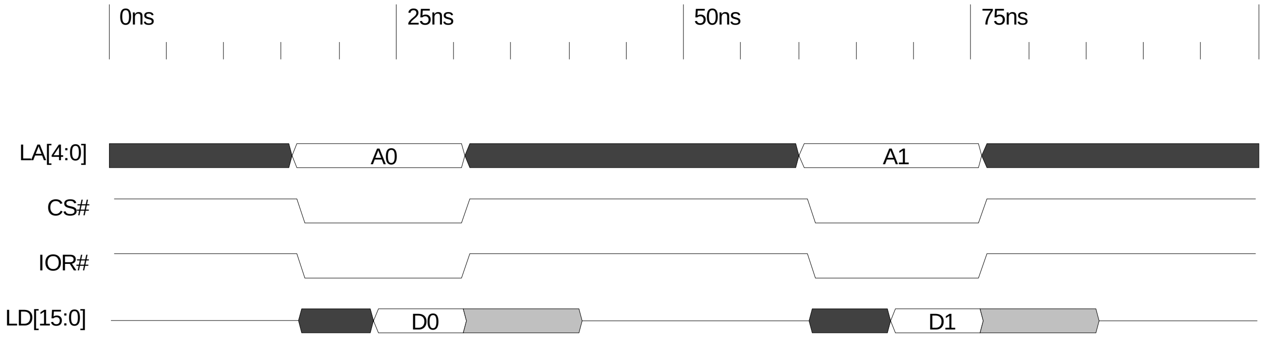


4.5.3 Non-Multiplexed I/O Read

Non-Multiplexed I/O Read transactions are started when both CS# and IOR# are asserted. The address must be valid T4 before CS# and IOR# are both asserted. Valid Read data is driven onto the Data bus within T6 after CS# and IOR# are both asserted. The Read transaction ends and the Data bus floats when CS# or IOR# is de-asserted. A new I/O Read transaction cannot start until the T8A recovery time expires, and a new I/O Write transaction cannot start until the T8B recovery time expires. ALE must be held low for Non-Multiplexed mode.

Note: *For 1.8, 2.5, and 3.3V VDDIO T timing values, refer to Table 11-31, Table 11-41, and Table 11-51, respectively.*

Figure 4-3. Non-Multiplexed I/O Read Timing

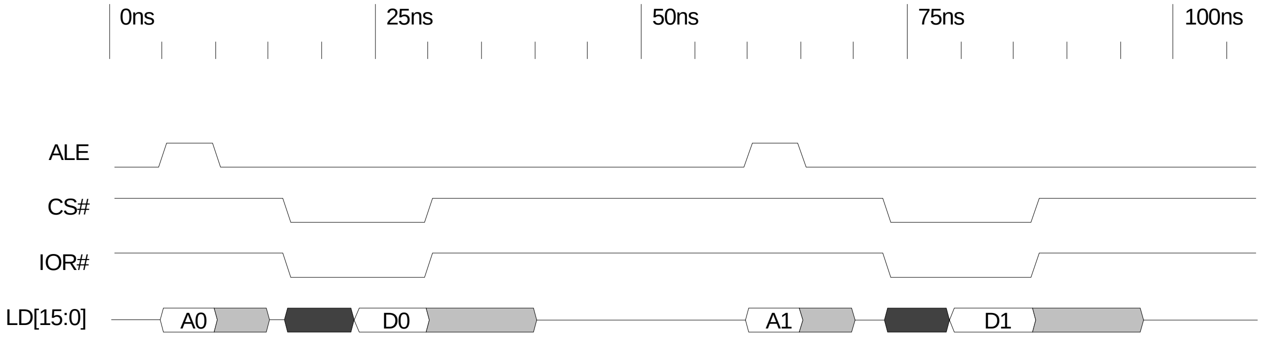


4.5.4 Multiplexed I/O Read

Multiplexed I/O Read transactions are started when the address is driven onto the lower bits of the Data bus, and ALE is pulsed. After the address is latched into the NET2272, the Data phase is initiated with CS# and IOR# assertion. To prevent Data bus contention, do ***not*** assert CS# and IOR# until the Local Bus master has three-stated the address. Valid Read data is driven onto the Data bus within T6 after CS# and IOR# are both asserted. The Read transaction ends and the Data bus floats when CS# or IOR# is de-asserted. A new I/O Read transaction cannot start until the T8A recovery time expires, and a new I/O Write transaction cannot start until the T8B recovery time expires.

Note: *For 1.8, 2.5, and 3.3V VDDIO T timing values, refer to Table 11-32, Table 11-42, and Table 11-52, respectively.*

Figure 4-4. Multiplexed I/O Read Timing



4.5.5 I/O Performance at 3.3V VDDIO

Table 4-1 delineates I/O performance specifications at 3.3V VDDIO.

Note: *For T timing values, refer to Table 11-49 through Table 11-52.*

Table 4-1. I/O Performance Specifications at 3.3V VDDIO

Transaction Type	Specification	Value
Non-Multiplexed Write	Write Width (T12)	5 ns
	Write to Write Recovery Time (T15A)	28 ns
	8-Bit Bus Maximum Performance	1/33 ns = 30 MB/s
	16-Bit Bus Maximum Performance	2/33 ns = 60 MB/s
Multiplexed Write	ALE Width (T3)	5 ns
	ALE to Write Command (T19)	1 ns minimum
	Write Width (T12)	5 ns
	Write to Write Recovery Time (T15A)	28 ns
	8-Bit Bus Maximum Performance	1/39 ns = 25 MB/s
	16-Bit Bus Maximum Performance	2/39 ns = 50 MB/s
Non-Multiplexed Read	Read Access Time (T6)	18 ns
	Read Recovery Time (T8)	19 ns
	8-Bit Bus Maximum Performance	1/37 ns = 27 MB/s
	16-Bit Bus Maximum Performance	2/37 ns = 54 MB/s
Multiplexed Read	ALE Width (T3)	5 ns
	ALE to Read Command (T19)	1 ns minimum
	Read Access Time (T6)	18 ns
	Read Recovery Time (T8)	19 ns
	8-Bit Bus Maximum Performance	1/43 ns = 23 MB/s
	16-Bit Bus Maximum Performance	2/43 ns = 46 MB/s

4.6 DMA Transactions

DMA transfers are those in which an external DMA controller transfers data between memory and one of the Packet buffers within the NET2272. DMA transfers can be configured only for Endpoint A or B. The Local CPU handles transfers to and from Endpoints 0 and C. The external DMA controller is programmed to perform Fly-By Demand mode transfers. In Fly-By Demand mode, transfers occur only when the NET2272 requests them. The data is transferred between the NET2272 and Local memory during the same Bus transaction.

During DMA transactions, the Endpoint buffer is determined by the **DMAREQ** register *DMA Endpoint Select* field. During CPU transactions, the Endpoint buffer is determined by the **PAGESEL** register *Page Select* field. In DMA Split Bus mode, CPU accesses to an Endpoint buffer can simultaneously occur with DMA accesses to another Endpoint buffer.

4.6.1 DMA Write

For IN transfers (NET2272-to-host), the Local and Host CPUs first arrange to transfer a block of data from Local memory to Host memory. The Local CPU programs the external DMA controller to transfer the needed number of bytes. The NET2272 **EP_TRANSFER*x*** registers are also programmed with the needed Transfer size (in bytes). The Transfer size programmed into the **EP_TRANSFER*x*** registers can span many packets, allowing a single DMA setup to transfer multiple packets. The DMA Request signal (DREQ) is asserted whenever Buffer space is available. The endpoint **Maximum Packet Size** register controls the maximum number of bytes transmitted to the host in a single packet. A Short packet is transmitted if bytes remain to transfer after all **EP_TRANSFER*x*** bytes are written, or when EOT is asserted during the last DMA cycle and the **DMAREQ** register *DMA Buffer Valid* bit is set.

The DREQ, DACK, IOW#, and EOT signals control transactions between the external DMA controller and NET2272. DREQ and DACK are minimally needed to exchange data with the NET2272, because the direction (write) is established by the **EP_CFG** register *Endpoint Direction* bit. The operation mode is set by the **DMAREQ** register *DMA Control DACK* bit. If the *DMA Control DACK* bit is high, the NET2272 needs DACK and IOW# for a DMA write. If the bit is low, only DACK is needed for a DMA write.

The Local CPU programs the NET2272 **DMAREQ** register to associate the DMA DREQ and DACK signals with a NET2272 endpoint (Endpoint A or B). Transfers occur only when the NET2272 requests them, after the **DMAREQ** register *DMA Request Enable* bit is set.

If space is available in the selected Endpoint buffer, and bytes remaining to transfer, the NET2272 requests DMA transfers by asserting DREQ. The external DMA controller then requests the Local Bus from the Local CPU. After the DMA controller is granted the bus, the controller drives DACK, IOW# (optional), and external DMA controller MEMR# output (to memory) to transfer one byte from memory to the Endpoint buffer. For DMA Slow mode, the NET2272 de-asserts DREQ within T20 after the transaction starts. For DMA Fast mode, the NET2272 de-asserts DREQ when the transaction starts. If there is Buffer space available and remaining bytes to transfer, the NET2272 re-asserts DREQ.

The USB host transmits an IN token to the NET2272 and starts an IN Data transaction from the selected Endpoint buffer. DMA transfers continue until the number of bytes specified by **EP_TRANSFER*x*** are transferred, or EOT is asserted.

The **IRQSTAT0** register *DMA Done Interrupt* bit is set for the following conditions:

- EOT is asserted during the last DMA transfer
- **EP_TRANSFER*x*** Counter counts down to 0
- Local CPU writes a 0 to the **EP_TRANSFER*x*** register after the DMA transfer is complete

When DMA Burst mode is selected, DREQ is asserted when Buffer space is available and the *DMA Request Enable* bit is set. DREQ remains asserted until one of the following conditions is met:

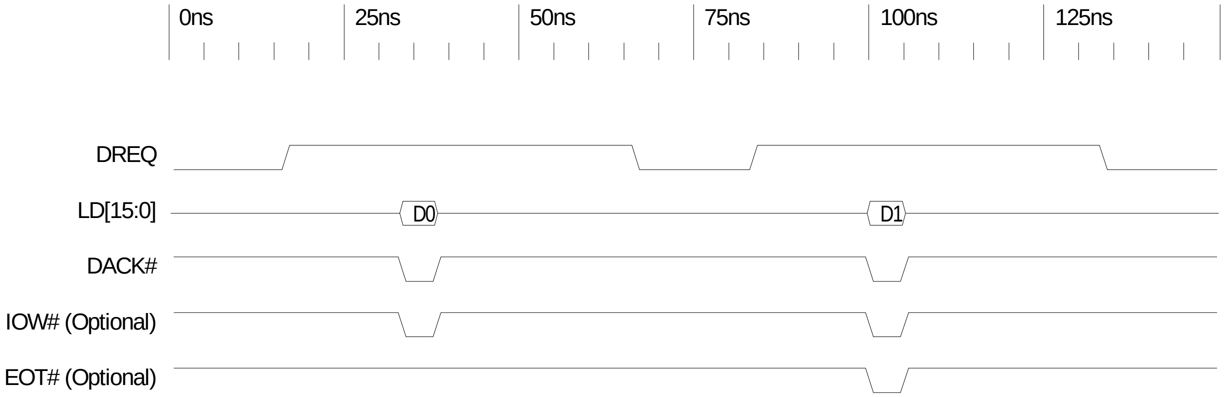
- Buffer becomes full
- *DMA Request Enable* bit is cleared
- **EP_TRANSFER*x*** Counter counts down to 0
- EOT is asserted

4.6.2 DMA Slow Mode Write Timing

For DMA Slow mode Write timing, DREQ is de-asserted within T20 after the Write transaction starts. DREQ is re-asserted within T21 after the Write transaction ends.

Note: *For 1.8, 2.5, and 3.3V VDDIO T timing values, refer to Table 11-33, Table 11-43, and Table 11-53, respectively.*

Figure 4-5. DMA Slow Mode Write Timing

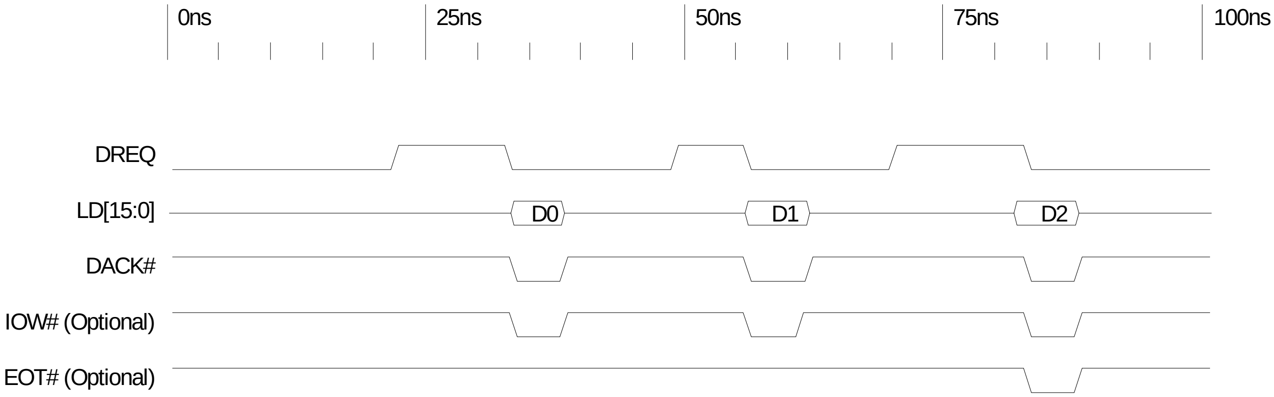


4.6.2.1 DMA Fast Mode Write Timing

In DMA Fast mode Write timing, DREQ is de-asserted when the Write transaction starts. DREQ is re-asserted within T21 after the Write transaction ends.

Note: *For 1.8, 2.5, and 3.3V VDDIO T timing values, refer to Table 11-34, Table 11-44, and Table 11-54, respectively.*

Figure 4-6. DMA Fast Mode Write Timing

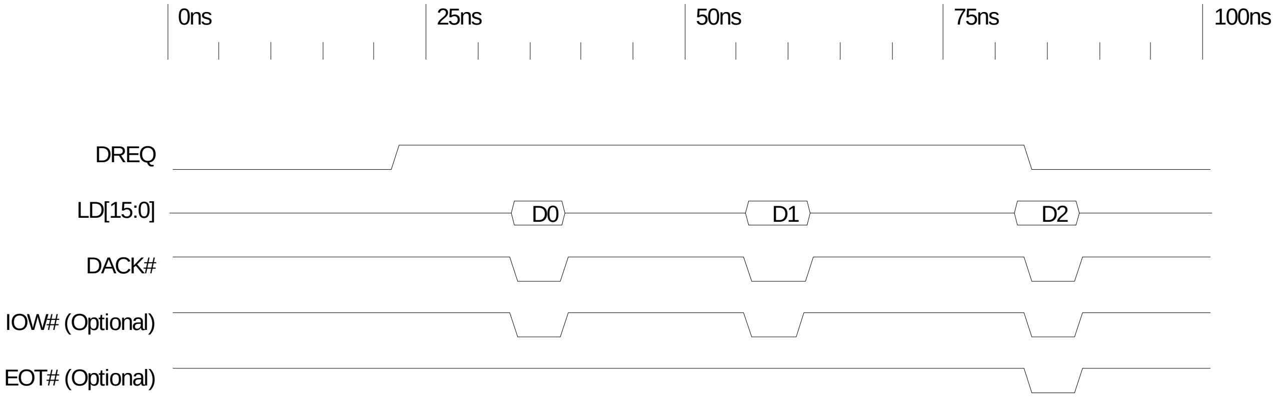


4.6.2.2 DMA Burst Mode Write Timing

In DMA Burst mode Write timing, DREQ remains asserted until the DMA transfer completes.

Note: *For 1.8, 2.5, and 3.3V VDDIO T timing values, refer to Table 11-35, Table 11-45, and Table 11-55, respectively.*

Figure 4-7. DMA Burst Mode Write Timing



4.6.3 DMA Read

For OUT transfers (host-to-NET2272), the Local and Host CPUs first arrange to transfer a block of data from Host memory to Local memory. The Local CPU programs the external DMA controller to transfer the needed number of bytes. The DREQ, DACK, IOR#, and EOT signals control transactions between the external DMA controller and the NET2272. DREQ and DACK are minimally needed to exchange data with the NET2272, because the direction (read) is established by the **EP_CFG** register *Endpoint Direction* bit. The operation mode is set by the **DMAREQ** register *DMA Control DACK* bit. If the *DMA Control DACK* bit is high, the NET2272 needs DACK and IOR# for a DMA read. If the bit is low, only DACK is needed for a DMA read.

The Local CPU programs the NET2272 **DMAREQ** register to associate the DMA DREQ and DACK signals with a NET2272 endpoint (Endpoint A or B). Transfers occur only when the NET2272 requests them, after the **DMAREQ** register *DMA Request Enable* bit is set.

When data is available in an Endpoint buffer, and that endpoint is assigned to a DMA channel, the DMA Request signal (DREQ) is asserted. The external DMA controller then requests the Local Bus from the Local Bus master. After the external DMA controller is granted the bus, the controller drives a valid Memory address and asserts DACK, IOR# (optional), and external DMA controller MEMW# output (to memory), thereby transferring one byte from an Endpoint buffer to Local memory. In DMA Slow mode, the NET2272 de-asserts DREQ within T20 after the transaction starts. In DMA Fast mode, the NET2272 de-asserts DREQ when the transaction starts. If data remains in the buffer, the NET2272 re-asserts DREQ. DMA transfers continue until the DMA Byte Count reaches 0, or EOT is asserted during the last DMA transfer. The **IRQSTAT0** register *DMA Done Interrupt* bit is set for the following conditions:

- EOT is asserted during the last DMA transfer
- Local CPU writes a 0 to the **EP_TRANSFER*x*** register after the DMA transfer completes
- Short packet is received and the Endpoint buffers are empty

When DMA Burst mode is selected, DREQ is asserted when there is data in the buffer and the *DMA Request Enable* bit is set. DREQ remains asserted until one of the following conditions is met:

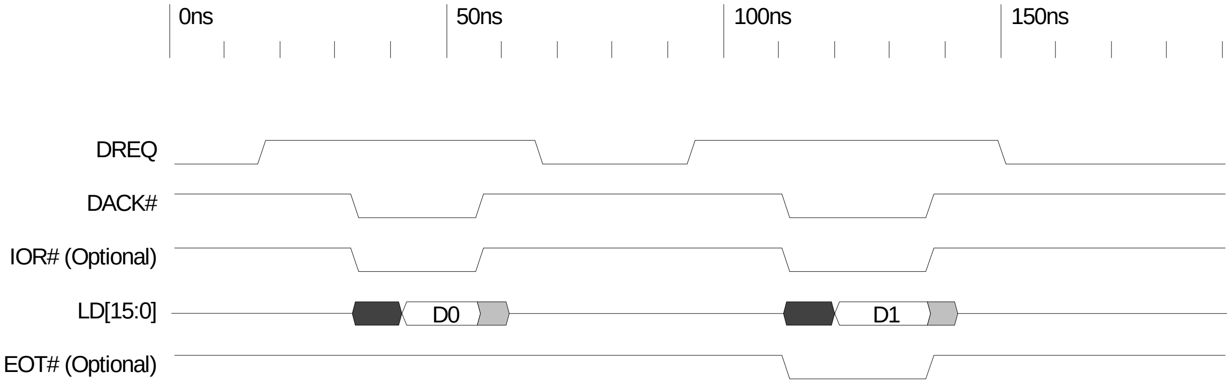
- Buffer becomes empty
- *DMA Request Enable* bit is cleared
- EOT is asserted

4.6.3.1 DMA Slow Mode Read Timing

In DMA Slow mode Read timing, DREQ is de-asserted within T20 after the Read transaction starts. DREQ is re-asserted within T21 after the Read transaction ends.

Note: *For 1.8, 2.5, and 3.3V VDDIO T timing values, refer to Table 11-36, Table 11-46, and Table 11-56, respectively.*

Figure 4-8. DMA Slow Mode Read Timing

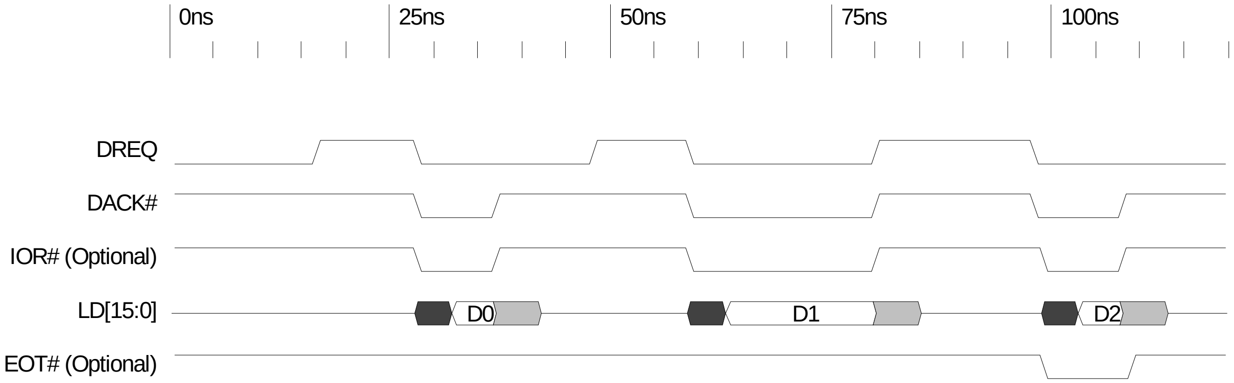


4.6.3.2 DMA Fast Mode Read Timing

In DMA Fast mode Read timing, DREQ is de-asserted when the Read transaction starts. DREQ is re-asserted, either when the Read transaction ends (if the Read Enable width is greater than T21), or at T21 after the Read transaction starts (if the Read Enable width is less than T21).

Note: *For 1.8, 2.5, and 3.3V VDDIO T timing values, refer to Table 11-37, Table 11-47, and Table 11-57, respectively.*

Figure 4-9. DMA Fast Mode Read Timing

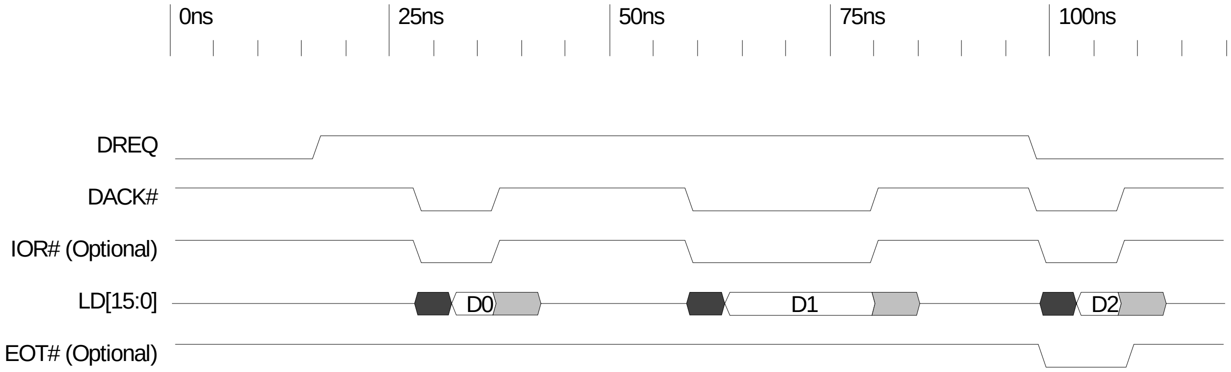


4.6.3.3 DMA Burst Mode Read Timing

In DMA Burst mode Read timing, DREQ remains asserted until the DMA transfer completes.

Note: *For 1.8, 2.5, and 3.3V VDDIO T timing values, refer to Table 11-38, Table 11-48, and Table 11-58, respectively.*

Figure 4-10. DMA Burst Mode Read Timing



4.6.4 DMA Split Bus Mode

In DMA Split Bus mode, the external DMA controller is connected to LD[15:8] of the Data bus, while the Local CPU is connected to LD[7:0] of the Data bus. The **LOCCTL** register *DMA Split Bus Mode* bit enables DMA Split Bus mode.

DMA Split Bus mode transactions are the same as normal DMA transactions, except that DMARD# and DMAWR# are used instead of IOR# and IOW#, respectively. While DMA transactions are proceeding using LD[15:8], the Local CPU can simultaneously access Configuration registers for any endpoint, and can access Endpoint buffers not involved with the DMA transfer.

4.6.5 Terminating DMA Transfers

The EOT signal is used to halt DMA transfers, and is typically provided by an external DMA controller. Assert EOT while DACK – and optionally IOR#, IOW#, DMARD#, or DMAWR# – are simultaneously active, to indicate that DMA activity has stopped. Although an EOT signal indicates that a DMA transfer has terminated, the USB transfer (in the case of an IN transaction) is not complete until the last byte is transferred from the Endpoint buffer to the USB. The EOT input resets the **DMAREQ** register *DMA Request Enable* bit. When EOT is detected asserted, the current Endpoint buffer is automatically validated, causing data remaining in the current packet to transmit to the host as a Short packet. If there is no data in the buffer when the current buffer is validated, then a Zero-Length packet is returned in response to the next IN token. The *DMA Request Enable* bit is also automatically cleared when the **EP_TRANSFER*x*** Counter reaches 0 for IN endpoints, or when the Local CPU writes a 0 to the **EP_TRANSFER*x*** Counter.

If the external DMA controller does not provide an EOT signal, the Local CPU can terminate the DMA transfer at any time by resetting the NET2272 *DMA Request Enable* bit. If the NET2272 *DMA Request Enable* bit is cleared during the middle of a DMA cycle (possible only when using DMA Split Bus mode), the current cycle completes before DMA requests are terminated. The Endpoint buffer is not automatically validated when the *DMA Request Enable* bit is cleared. In this case, the CPU must explicitly validate the packet by writing 0 to the **EP_TRANSFER*x*** register.

4.6.6 DMA Performance at 3.3V VDDIO

Table 4-2 delineates DMA performance specifications at 3.3V VDDIO.

Note: *For T timing values, refer to Table 11-53 through Table 11-58.*

Table 4-2. DMA Performance Specifications at 3.3V VDDIO

Transaction Type	Mode	Specification	Value
DMA Write	Slow*	DREQ to DACK (Depends on DMA Controller)	5 ns
		DACK Asserted to DREQ de-Asserted (T20)	50 ns
		DREQ De-Asserted to DREQ Asserted (T25)	25 ns
		8-Bit Bus Maximum Performance	1/80 ns = 12.5 MB/s
		16-Bit Bus Maximum Performance	2/80 ns = 25 MB/s
	Fast*	DREQ to DACK (Depends on DMA Controller)	5 ns
		Write Width (T26)	5 ns
		DACK De-Asserted to DREQ Asserted (T21)	45 ns
		8-Bit Bus Maximum Performance	1/55 ns = 18 MB/s
		16-Bit Bus Maximum Performance	2/55 ns = 36 MB/s
	Burst	Write Width (T26)	5 ns
		Write Recovery (T30)	36 ns
		8-Bit Bus Maximum Performance	1/41 ns = 24 MB/s
		16-Bit Bus Maximum Performance	2/41 ns = 48 MB/s
DMA Read	Slow*	DREQ to DACK (Depends on DMA Controller)	5 ns
		DACK Asserted to DREQ De-Asserted (T20)	50 ns
		DREQ De-Asserted to DREQ Asserted (T25)	25 ns
		8-Bit Bus Maximum Performance	1/80 ns = 12.5 MB/s
		16-Bit Bus Maximum Performance	2/80 ns = 25 MB/s
	Fast*	DREQ to DACK (Depends on DMA Controller)	5 ns
		Read Width (T22)	16 ns
		DACK De-Asserted to DREQ Asserted (T21)	35 ns
		8-Bit Bus Maximum Performance	1/56 ns = 17.8 MB/s
		16-Bit Bus Maximum Performance	2/56 ns = 35.7 MB/s
	Burst	DMA Read Cycle Time (T17)	35 ns
		8-Bit Bus Maximum Performance	1/35 ns = 28 MB/s
		16-Bit Bus Maximum Performance	2/35 ns = 57 MB/s

Note: ** Actual throughput in DMA Slow and Fast modes is reduced if the DMA controller DREQ-to-DACK delay is longer than 5 ns.*

Chapter 5 USB Functional Description

5.1 USB Interface

The NET2272 is a USB-function device, and as a result is a slave to the USB host. The bit- and packet-level protocols, as well as the NET2272 electrical interface, conform to the *USB r2.0*. The USB host initiates all USB Data transfers to and from the NET2272 USB port. The NET2272 is configured for up to three Physical and 30 Virtual endpoints, in addition to Endpoint 0. Endpoints can be of type Isochronous, Bulk, or Interrupt. The Configuration registers are used to program endpoint characteristics. The NET2272 operates in Full- (12 Mbps) or High-speed (480 Mbps) modes.

5.2 USB Protocol

The USB packet protocol consists of tokens, packets, transactions, and transfers.

5.2.1 Tokens

Tokens are a type of Packet Identifier (PID), and follow the Sync field at the beginning of a token. The four classic token types are OUT, IN, SOF, and SETUP. In High-Speed mode, the NET2272 also recognizes the PING token.

5.2.2 Packets

There are four types of packets – Start-of-Frame (SOF), Token, Data, and Handshake – which are transmitted and received in the order listed in Table 5-1. Each packet begins with a Sync field and a Packet Identifier (PID). The other fields vary, depending on the packet type. Table 5-1 delineates the four USB protocol packet types.

Table 5-1. USB Protocol Packets

Packet Type	Number of Bits							
	Sync Field	Packet Identifier (PID)	Frame Number	Address	Endpoint	Data	Cyclic Redundancy Checks (CRC)	*Total*
1. Start-of-Frame (SOF)	8	8	11	–	–	–	5	32
2. Token	8	8	–	7	4	–	5	32
3. Data	8	8	–	–	–	*N*	16	32 + *N*
4. Handshake	8	8	–	–	–	–	–	16

5.2.3 Transaction

A transaction consists of a Token packet, optional Data packet(s), and Handshake packet.

5.2.4 Transfer

A transfer consists of one or more transactions. Control transfers consist of a Setup transaction, optional Data transactions, and a Status transaction.

5.3 Automatic Retries

5.3.1 OUT Transactions

If an error occurs during an OUT transaction, the NET2272 reloads its Local Bus Buffer Read pointer to the beginning of the failed packet. The host then transmits another OUT token and re-transmits the packet. After the NET2272 successfully receives the packet, the *Data Packet Received Interrupt* bit is set. The NET2272 can handle any number of back-to-back Retries; however, the host determines the number of packet Retries.

5.3.2 IN Transactions

If an error occurs during an IN transaction, the NET2272 reloads its USB Buffer Read pointer to the beginning of the failed packet. The host then transmits another IN token and the NET2272 re-transmits the packet. After the host successfully receives the packet, the *Data Packet Transmitted Interrupt* bit is set.

5.4 PING Flow Control

When operating in High-Speed mode, the NET2272 supports PING protocol for Bulk OUT and Control endpoints. This protocol allows the NET2272 to indicate to the host that it cannot accept an OUT packet. The host then transmits PING tokens to query the NET2272. The NET2272 returns an ACK in response to the PING when it is able to accept a maximum-size packet. At this time, the host transmits an OUT token and Data packet. The NET2272 returns an ACK handshake if the packet is accepted, and there is sufficient space to receive an additional packet. The NET2272 returns a NYET handshake to the host if it can accept only the current packet. The host then starts transmitting PING tokens.

5.5 Packet Sizes

An endpoint Maximum Packet Size is determined by the corresponding **EP_MAXPKT*x*** register. For IN transactions, the NET2272 returns a maximum-size packet to the host if the number of “Maximum Packet” bytes exist in the buffer. If the buffer data is validated, a packet size less than the maximum is returned to the host in response to an IN token. Table 5-2 delineates the allowable Maximum Packet Sizes.

Table 5-2. Allowable Maximum Packet Sizes

Endpoint Type	Allowable Maximum Packet Size (Bytes)		
	Low-Speed Mode	Full-Speed Mode	High-Speed Mode
Bulk	N/A	8, 16, 32, 64	512
Control	8	8, 16, 32, 64	64
Interrupt	8	64 maximum	1,024 maximum
Isochronous	N/A	1,023 maximum	1,024 maximum

5.6 USB Endpoints

The NET2272 supports Control, Isochronous, Bulk, and Interrupt endpoints. All endpoints are uni-directional except for Control endpoints. Bi-directional Isochronous, Bulk, and Interrupt traffic requires two endpoints.

5.6.1 Control Endpoint (Endpoint 0)

The Control endpoint, Endpoint 0, is a ***reserved*** endpoint. The host uses this endpoint to configure and acquire information about the NET2272, its configurations, interfaces, and other endpoints. Control endpoints are bi-directional, and data delivery is guaranteed.

The host transmits 8-byte Setup packets to Endpoint 0, to which the NET2272 interprets and responds. The NET2272 consists of a set of registers dedicated to storing the Setup packet, and uses the Endpoint 0 Packet buffer for Control data. For Control writes, data flows through the Packet buffer from the USB-to-Local Bus. For Control reads, data flows through the Packet buffer from the Local Bus-to-USB.

When Endpoint 0 detects a Setup packet, the NET2272 sets status bits and interrupts the Local CPU. The CPU reads the Setup packet from NET2272 registers, and responds based on the contents. The Local CPU provides data to return to the host, including status and descriptors. Refer to Chapter 9, “USB Standard Device Requests,” for a description of the data that must be returned for each USB request. The host rejects descriptors that contain unexpected field values.

5.6.1.1 Control Write Transfer

A successful Control Write transfer to Control Endpoint 0 consists of the data delineated in Table 5-3.

Table 5-3. Control Write Transfer

Transaction	Stage	Packet Contents	Number of Bytes	Source
Setup	Setup Token	SETUP PID, address, endpoint, and CRC5	3	Host
	Data	DATA0 PID, 8 data bytes, and CRC16	11	Host
	Status	ACK	1	NET2272
Data (zero, one, or more packets)	OUT Token	OUT PID, address, endpoint, and CRC5	3	Host
	Data (1/0)	DATA PID, *N* data bytes, and CRC16	*N*+3	Host
	Status	ACK	1	NET2272
Status	IN Token	IN PID, address, endpoint, and CRC5	3	Host
	Data	DATA1 PID, Zero-Length packet, and CRC16	3	NET2272
	Status	ACK	1	Host

During the Setup transaction, the NET2272 stores the Data Stage packet in its Setup registers. The NET2272 returns an ACK handshake to the host after all eight bytes are received. A *Setup Packet Interrupt* bit is set to notify the Local CPU that a Setup packet was received. The Local CPU reads and interprets the 8-byte Data packet. A Setup transaction cannot be stalled or NAKed; however, if the data is corrupt, the NET2272 does not return an ACK to the host.

During the optional Data transaction, zero, one, or more Data packets are written into the Endpoint 0 buffer. For each packet:

1. Interrupt bits are set and can interrupt the Local CPU.
2. Local CPU reads the buffer.
3. NET2272 returns an ACK if no error occurred.

For a successful Status transaction, the NET2272 returns a Zero-Length Data packet. A NAK or STALL handshake is returned if an error occurs.

5.6.1.2 Control Write Transfer Details

For Control Write transfers, the host first transmits eight bytes of setup information. The Setup bytes are stored into an 8-byte register bank that is accessed by the Local CPU. After the eight bytes are stored into the Setup registers, the *Setup Packet Interrupt* bit is set. The Local CPU then reads the 8-byte Setup packet and prepares to respond to the optional Data transaction. The number of bytes to transfer in the Data transactions is specified in the Setup packet. When the Setup packet is received, the *Control Status Stage Handshake* bit is automatically set, in anticipation of the Control Status transaction. While this bit is set, the Control Status transaction is acknowledged with a NAK, allowing the Local CPU to prepare its handshake response (ACK or STALL). After the *Control Status Stage Handshake* bit is cleared and the OUT buffer is empty, an ACK or STALL handshake is returned to the host. Waiting for the OUT buffer to become empty prevents another Control Write from corrupting the current Packet data in the buffer.

During a Control Write operation, optional Data transactions can follow the Setup transaction. The *Data Out Token Interrupt* bit is set at the beginning of each Data transaction. The bytes corresponding to the Data transaction are stored into the Endpoint 0 buffer. If the buffer fills and the host transfers another byte, the NET2272 returns a NAK handshake to the host, signaling that the data cannot be accepted.

If a packet is not successfully received (NAK or Timeout status), the *Data Packet Received Interrupt* bit is not set, and the data is automatically flushed from the buffer. The host later re-transmits the same packet. This process is transparent to the Local CPU.

If the Local CPU stalled this endpoint by setting the *Endpoint Halt* bit, the NET2272 does not store data into the buffer, and responds with a STALL acknowledge to the host. There is no Status transaction in this case.

The Local CPU can poll the *Data Packet Received Interrupt* bit, or enable the bit as an interrupt, and then read the packet from the buffer. If the host tries to write more data than indicated in the Setup packet, the Local CPU sets the *Endpoint Halt* bit for Endpoint 0. In this case, there is no Status transaction from the host.

After all optional Data Stage packets are received, the host transmits an IN token, signifying the Status transaction. The *Control Status Interrupt* bit is set after the Status transaction IN token is received. Until the Local CPU clears the *Control Status Stage Handshake* bit and the OUT buffer is empty, the NET2272 responds with NAKs, indicating that the NET2272 is processing the Setup command. When the Local CPU clears the *Control Status Stage Handshake* bit and firmware has emptied the OUT buffer data, the NET2272 responds with a Zero-Length Data packet (transfer OK) or STALL (error encountered).

5.6.1.3 Control Read Transfer

A successful Control Read transfer from Control Endpoint 0 consists of the data delineated in Table 5-4.

Table 5-4. Control Read Transfer

Transaction	Stage	Packet Contents	Number of Bytes	Source
Setup	Setup Token	SETUP PID, address, endpoint, and CRC5	3	Host
	Data	DATA0 PID, 8 data bytes, and CRC16	11	Host
	Status	ACK	1	NET2272
Data (zero, one, or more packets)	IN Token	IN PID, address, endpoint, and CRC5	3	Host
	Data (1/0)	DATA PID, *N* data bytes, and CRC16	*N*+3	NET2272
	Status	ACK	1	Host
Status	OUT Token	OUT PID, address, endpoint, and CRC5	3	Host
	Data	DATA1 PID, Zero-Length packet, and CRC5	3	Host
	Status	ACK	1	NET2272

The Setup transaction is processed with the same method as Control Write transfers. (Refer to Section 5.6.1.2.) During the optional Data transaction, zero, one, or more Data packets are read from the Endpoint 0 buffer. For each packet:

1. Interrupt bits are set and can interrupt the Local CPU.
2. Local CPU writes data to the buffer.
3. If there is no data in the buffer, a NAK or Zero-Length packet is returned to the host.
4. Host returns an ACK to the NET2272 if no error occurred.

For a successful Status transaction, the Host transmits a Zero-Length Data packet, and the NET2272 responds with an ACK. A NAK or STALL is returned if an error occurred.

5.6.1.4 Control Read Transfer Details

For Control Read transfers, the host first transmits eight bytes of setup information. The Setup bytes are stored into an 8-byte register bank, accessed from the Local CPU. After the eight bytes are stored into the Setup registers, the *Setup Packet Interrupt* bit is set. The Local CPU then reads the 8-Byte Setup packet and prepares to respond to the optional Data transaction. The number of bytes to transfer in the Data transaction is specified in the Setup packet. When the Setup packet is received, the *Control Status Stage Handshake* bit is automatically set. While this bit is set, the Control Status transaction is acknowledged with a NAK, allowing the Local CPU to prepare its handshake response (ACK or STALL). After the *Control Status Stage Handshake* bit is cleared, an ACK or STALL handshake is returned to the host.

During a Control Read operation, optional Data transactions can follow the Setup transaction. After the Setup transaction, the Local CPU can start writing the first byte of Packet data into the Endpoint 0 buffer, in anticipation of the Data transaction. The *Data In Token Interrupt* bit is set at the beginning of each Data transaction. If there is data in the Endpoint 0 buffer, the data is returned to the host. If Endpoint 0 has no data to return, the endpoint returns a Zero-Length packet (signaling that no further data is available) or NAK handshake (the data is not available). The NET2272 responds to the Data transaction IN token, according to the data delineated in Table 5-5.

Table 5-5. Control Read Transfer Details

Packet Validated	Amount of Data in Buffer	Action
0	< Maximum Packet Size	NAK to host
X	≥ Maximum Packet Size	Return data to host
1	Empty	Zero-Length packet to host
1	> 0	Return data to host

***Note:** "X" is "Don't Care."*

After each packet is transmitted to the host, the *Data Packet Transmitted Interrupt* bit is set.

If a packet is not successfully transmitted (*Timeout* status bit set), the *Data Packet Transmitted Interrupt* bit is not set, and this packet is transmitted to the host when another IN token is received. The Retry operation is transparent to the Local CPU.

If the host tries to read more data than requested in the Setup packet, the Local CPU sets the *STALL* bit for the endpoint.

After all optional Data Stage packets are transmitted, the host transmits an OUT token, followed by a Zero-Length Data packet, signifying the Status transaction. The *Control Status Interrupt* bit is set after the Status transaction OUT token is received. Until the Local CPU clears the *Control Status Stage Handshake* bit, the NET2272 responds with NAKs, indicating that the NET2272 is processing the command specified by the Setup transaction. When the Local CPU clears the *Control Status Stage Handshake* bit, the NET2272 responds with an ACK (transfer OK) or STALL (Endpoint 0 is stalled).

5.6.2 Isochronous Endpoints

Isochronous endpoints are used for time-critical Data transfers. Isochronous transfers do not support handshaking nor error-checking protocol, and are guaranteed a certain amount of bandwidth during each frame. The NET2272 SIE ignores Cyclic Redundancy Checks (CRC) and Bit-Stuffing errors during Isochronous transfers; however, the engine sets the **EP_STAT*x*** register handshaking status bits the same as it sets for Non-Isochronous packets, enabling the Local CPU to detect the errors. Isochronous endpoints are uni-directional, with the direction defined by the Endpoint Configuration registers.

For Isochronous endpoints, the Packet Buffer Size must be equal to or greater than the Maximum Packet Size. The Maximum Packet Size for an Isochronous endpoint ranges from 1 to 1,024 bytes.

For an Isochronous OUT endpoint, the Local CPU or DMA controller can read data from the Endpoint buffer after an entire packet is received. If the Endpoint buffer is the same size as the Maximum Packet Size, then the ISO bandwidth must be set so the buffer is emptied before the next ISO packet arrives.

For an Isochronous IN endpoint, the Local CPU or DMA controller can write data to one Endpoint buffer at the same time that data is being transmitted to the USB from the other Endpoint buffer (Double-Buffered mode only).

5.6.2.1 Isochronous OUT Transactions

Isochronous OUT endpoints transfer data from a USB host to the NET2272 Local Bus. An Isochronous OUT transaction consists of the data delineated in Table 5-6.

Table 5-6. Isochronous OUT Transactions

Stage	Packet Contents	Number of Bytes	Source
OUT Token	OUT PID, address, endpoint, and CRC5	3	Host
Data	DATA0 PID, *N* data bytes, and CRC16	*N*+3	Host

The USB host initiates an Isochronous OUT transaction by transmitting an OUT token to an Isochronous OUT endpoint. The *Data OUT Token Interrupt* bit is set when the OUT token is recognized. The bytes corresponding to the Data stage are stored into the Endpoint buffer. Isochronous transactions are not Retried; therefore, if the buffer is full when the host transfers a packet (or the *NAK OUT Packets* bit is set), the packet is discarded. No Handshake packets are returned to the host; however, the *USB OUT ACK Transmitted* and *Timeout* status bits are set to indicate transaction status. If a CRC error is detected, the packet is accepted, and the *Timeout* status bit is set. After all Data packets are received, the Local CPU samples these status bits to determine whether the NET2272 successfully received the packet.

By definition, Isochronous endpoints do not utilize handshaking with the host. Because there is no method to return a stalled handshake from an Isochronous endpoint to the host, data that is transmitted to a stalled Isochronous endpoint is received normally. The Maximum Packet Size must be less than or equal to the Buffer size.

The Local CPU must wait for the *Data Packet Received Interrupt* bit to be set before reading the data from the buffer. If the endpoint is programmed for single-buffering, then the host is programmed to allow the Local CPU sufficient time to unload the buffer before the next packet is transmitted. If the endpoint is programmed for double-buffering, then the Local Bus can unload one packet while the next packet is being received.

5.6.2.2 High-Bandwidth Isochronous OUT Transactions

The host transmits high-bandwidth OUT PID sequences for each microframe, depending on the Endpoint descriptor *Additional Transaction Opportunities* field, as delineated in Table 5-7.

Table 5-7. High-Bandwidth Isochronous OUT Transactions

Additional Transaction Opportunities	PID Sequence
0	DATA0 (normal ISO)
1	MDATA, DATA1 (one additional transaction)
2	MDATA, MDATA, DATA2 (two additional transactions)

The NET2272 accepts data (unless the Endpoint buffer is full), and records the PID in the **EP_HBW** register *High-Bandwidth OUT Transaction PID* field. This allows firmware to track PIDs as they arrive and determine whether the data sequence is complete. (Refer to Table 5-8.)

Table 5-8. High-Bandwidth Isochronous OUT PID Values

High-Bandwidth OUT Transaction PID Field	PID Received
00b	DATA0
01b	DATA1
10b	DATA2
11b	MDATA

5.6.2.3 Isochronous IN Transactions

Isochronous IN endpoints transfer data from the NET2272 Local Bus to a USB host. An Isochronous IN transaction consists of the data delineated in Table 5-9.

Table 5-9. Isochronous IN Transactions

Stage	Packet Contents	Number of Bytes	Source
IN Token	IN PID, address, endpoint, and CRC5	3	Host
Data	DATA0 PID, *N* data bytes, and CRC16	*N*+3	NET2272

The USB host initiates an Isochronous IN transaction by transmitting an IN token to an Isochronous IN endpoint. The *Data IN Token Interrupt* bit is set when the IN token is recognized. If there is data in the Endpoint buffer, the data is returned to the host. If the endpoint has no data to return, a Zero-Length packet is returned to the host. The NET2272 responds to the IN token, according to the data delineated in Table 5-10.

Table 5-10. IN Token Response

Packet Validated	Amount of Data in Buffer	Action
0	< Maximum Packet Size	Zero-Length packet to host; *USB IN NAK Transmitted* status bit set
X	≥ Maximum Packet Size	Return data to host
1	Empty	Zero-Length packet to host
1	> 0	Return data to host

Note: *"X" is "Don't Care."*

After the packet is transmitted to the host, the *Data Packet Transmitted Interrupt* bit is set. If an IN token arrives and there is no valid packet in the Endpoint buffer, the NET2272 returns a Zero-Length packet. No Handshake packets are returned to the host; however, the *USB IN ACK Transmitted* and *Timeout* status bits are set to indicate transaction status. After all Data packets are transmitted, the Local CPU samples these status bits to determine whether the packets were successfully transmitted to the host.

By definition, Isochronous endpoints do not utilize handshaking with the host. Because there is no method to return a stalled handshake from an Isochronous endpoint to the host, data that is requested from a stalled Isochronous endpoint is transmitted normally.

5.6.2.4 High-Bandwidth Isochronous IN Transactions

A USB device that provides High-Bandwidth ISO IN endpoints is required to transmit ISO PID sequences for each microframe, according to the Endpoint descriptor *Additional Transaction Opportunities* field in the corresponding **EP_MAXPKT*x*** register, as delineated in Table 5-11.

Table 5-11. High-Bandwidth Isochronous IN Transactions

Additional Transaction Opportunities	PID Sequence
0	DATA0 (normal ISO)
1	DATA1, DATA0 (one additional transaction)
2	DATA2, DATA1, DATA0 (two additional transactions)

When the first microframe IN token arrives, the NET2272 copies the **EP_MAXPKT*x*** register *Additional Transaction Opportunities* field to determine the initial PID. On each succeeding microframe IN token, the PID advances to the next token.

5.6.3 Bulk Endpoints

Bulk endpoints are used for guaranteed error-free delivery of large amounts of data between a host and device. Bulk endpoints are uni-directional, with the direction defined by the **Endpoint Configuration** registers.

5.6.3.1 Bulk OUT Transactions

Bulk OUT endpoints transfer data from a USB host to the NET2272 Local Bus. A Bulk OUT transaction to a Bulk OUT endpoint consists of the data delineated in Table 5-12.

Table 5-12. Bulk OUT Transactions

Stage	Packet Contents	Number of Bytes	Source
OUT Token	OUT PID, address, endpoint, and CRC5	3	Host
Data (1/0)	DATA PID, *N* data bytes, and CRC16	*N*+3	Host
Status	ACK, NAK, or STALL	1	NET2272

The USB host initiates a Bulk OUT transaction by transmitting an OUT token to a Bulk OUT endpoint. The *Data OUT Token Interrupt* bit is set when the OUT token is recognized. The bytes corresponding to the Data stage are stored into the Endpoint buffer. If the buffer is full when the host transfers another packet, the packet is discarded and the *USB OUT NAK Transmitted* status bit is set. At packet completion, a NAK handshake is returned to the host, indicating that the packet cannot be accepted.

All USB data passes through Endpoint buffers to the Local Bus. The CPU waits until the *Data Packet Received Interrupt* occurs before reading the Buffer data.

If a packet is not successfully received (*USB OUT NAK Transmitted* or *Timeout* status bits set), the *Data Packet Received Interrupt* bit is not set, and the data is automatically flushed from the buffer. The host later re-transmits the same packet. This process is transparent to the Local CPU.

If the Local CPU stalled this endpoint by setting the *Endpoint Halt* bit, the NET2272 does not store data into the buffer, and responds with a STALL handshake to the host.

5.6.3.2 Bulk IN Endpoints

Bulk IN endpoints transfer data from the NET2272 Local Bus to a USB host. A Bulk IN transaction from a Bulk IN endpoint consists of the data delineated in Table 5-13.

Table 5-13. Bulk IN Endpoints

Stage	Packet Contents	Number of Bytes	Source
IN Token	IN PID, address, endpoint, and CRC5	3	Host
Data (1/0)	DATA PID, *N* data bytes, and CRC16, or NAK or STALL	*N*+3	NET2272
Status	ACK	1	Host

The USB host initiates a Bulk IN transaction by transmitting an IN token to a Bulk IN endpoint. The *Data IN Token Interrupt* bit is set when the IN token is recognized. If there is validated data in the Endpoint buffer, the data is returned to the host. If the endpoint has no data to return, a Zero-Length packet (signaling that no further data is available) or NAK handshake (the data is not available yet) is returned. The NET2272 responds to the IN token, according to the data delineated in Table 5-14.

Table 5-14. Bulk IN Endpoints Packet Validation

Packet Validated	Amount of Data in Buffer	Action
0	< Maximum Packet Size	NAK to host
X	≥ Maximum Packet Size	Return data to host
1	Empty	Zero-Length packet to host
1	> 0	Return data to host

Note: *"X" is "Don't Care."*

After the packet is transmitted to the host, the *Data Packet Transmitted Interrupt* bit is set.

If a packet is not successfully transmitted (*Timeout* status bit set), the *Data Packet Transmitted Interrupt* bit is not set, and the same packet is transmitted to the host when another IN token is received. The Retry operation is transparent to the Local CPU.

If the Local CPU stalled this endpoint by setting the *Endpoint Halt* bit, the NET2272 responds to the IN token with a STALL handshake to the host.

5.6.4 Interrupt Endpoints

Interrupt endpoints are used for transmitting or receiving small amounts of data to the host with a bounded service period.

5.6.4.1 Interrupt OUT Transactions

Interrupt OUT endpoints transfer data from a USB host to the NET2272 Local Bus. An Interrupt OUT transaction to an Interrupt OUT endpoint consists of the data delineated in Table 5-15.

Table 5-15. Interrupt OUT Transactions

Stage	Packet Contents	Number of Bytes	Source
OUT Token	OUT PID, address, endpoint, and CRC5	3	Host
Data (1/0)	DATA PID, *N* data bytes, and CRC16	*N*+3	Host
Status	ACK, NAK, or STALL	1	NET2272

The behavior of an Interrupt OUT endpoint is essentially the same as a Bulk OUT endpoint, except for the *Endpoint Toggle* bit. If the **EP_RSPCLR/EP_RSPSET** register *Interrupt Mode* bit is cleared, the Interrupt OUT *Endpoint Toggle* bit is initialized to 0 (DATA0 PID), and behaves the same as a Bulk OUT endpoint. If the *Interrupt Mode* bit is set, the Interrupt OUT *Endpoint Toggle* bit changes after each Data packet is received from the host, without regard to the Status stage. PING protocol is not allowed for Interrupt OUT endpoints, as per the *USB r2.0*.

5.6.4.2 Interrupt IN Endpoints

An Interrupt IN endpoint is polled at a rate specified in the Endpoint descriptor. An Interrupt transaction from an Interrupt IN endpoint consists of the data delineated in Table 5-16.

Table 5-16. Interrupt IN Endpoints

Stage	Packet Contents	Number of Bytes	Source
IN Token	IN PID, address, endpoint, and CRC5	3	Host
Data (1/0)	DATA PID, *N* data bytes, and CRC16	*N*+3	NET2272
Status	ACK	1	Host

The behavior of an Interrupt IN endpoint is the same as a Bulk IN endpoint, except for the toggle bit. If the **EP_RSPCLR/EP_RSPSET** register *Interrupt Mode* bit is cleared, the Interrupt IN *Endpoint Toggle* bit is initialized to 0 (DATA0 PID), and behaves the same as a Bulk IN endpoint. Use Interrupt endpoints to communicate rate feedback information for certain Isochronous functions. To support this mode, the *Interrupt Mode* bit is set, and the Interrupt IN *Endpoint Toggle* bit changes after each Data packet is transmitted to the host, without regard to the Status stage.

5.6.4.3 High-Bandwidth Interrupt Endpoints

From the USB device point of view, high-bandwidth Interrupt endpoints are the same as Bulk endpoints, except that the MAXPKT is a value from 1 to 1,024. Normal Interrupt endpoints in Full-Speed mode can set MAXPKT from 1 to 64.

5.7 PLX Virtual Endpoints

PLX's Virtual Endpoint hardware support allows a USB device to utilize the full potential of USB endpoints, by providing the capability to expose any number of endpoints to the USB host. Firmware can track and assign the available Physical endpoints to the dynamically required Logical endpoints, with full flexibility. Any number of endpoints required by a host driver (including USB Class drivers) are supported.

5.7.1 Overview

The NET2272 features PLX Virtual Endpoint hardware support, which enables firmware to implement any number of USB device endpoints (up to a maximum of 15 per direction), excluding Endpoint 0. (Refer to the *USB r2.0*, Sections 8.3.2.2 and 9.6.6.) Hardware support for endpoint virtualization consists of the **Virtual Interrupt** registers – **VIRTOUT0**, **VIRTOUT1**, **VIRTIN0**, and **VIRTIN1** – and the *Virtual Endpoint Interrupt* status bit.

In this section, "Logical endpoint" refers to the Endpoint Number from the host point of view (embedded in the OUT, IN, and SETUP tokens), "Physical endpoint" refers to the NET2272 hardware (Endpoints 0, A, B, or C), and "Unassigned endpoint" refers to a Logical Endpoint Number that is not currently assigned (by way of the **EP_CFG** register *Endpoint Address* field) to a Physical endpoint.

When the **USBCTL1** register *Virtual Endpoint Enable* bit is low (default), the NET2272 responds to a Host request to an Unassigned endpoint with a timeout. The host considers a timeout response as a fatal error. The host Retries transactions with fatal errors; however, after 256 consecutive Retries, the host shuts down the pipe (endpoint).

When the *Virtual Endpoint Enable* bit is high, the NET2272 responds to all requests on Unassigned endpoints with a NAK. This causes the host to Retry the request until an ACK is returned.

In addition to NAKing, the NET2272 sets the bit in the **Virtual Interrupt** register that corresponds to the requesting Logical Endpoint Number. *For example*, if the host requests an IN token on Endpoint 3 when none of the NET2272 Physical endpoints are assigned to Address 3 with direction IN, **VIRTIN0** register, bit 3, is set (and the NET2272 NAKs the IN request).

When **Virtual Endpoint** register bits are set, the corresponding *Virtual Endpoint Interrupt* status bit is also set. If this interrupt is enabled, firmware is notified that the host tried to access an Unassigned endpoint. Firmware can then re-assign one of the Physical endpoints to the new Logical endpoint, allowing the Data transaction to proceed.

Virtual endpoint participation is completely flexible – all Physical endpoints (excluding Endpoint 0) can participate in Virtual endpoint re-assignment, or some Physical endpoints are dedicated to specific high-usage Logical endpoints.

5.7.2 Endpoint Virtualization

Virtualization relies on the firmware's ability to capture, preserve, and restore the complete Endpoint state as the firmware switches the available Physical endpoint resources between a larger number of Logical endpoints (similar to a CPU context switch). PLX Virtual Endpoint hardware support makes all the Endpoint state information available to the firmware.

When re-assigning endpoints, firmware must ensure that USB traffic is not disturbed. Specifically, an endpoint should not be reprogrammed or flushed while the endpoint is enabled. PLX Virtual Endpoint hardware support includes logic to prevent an endpoint's Enable state from changing while a USB transaction to the endpoint is in progress. Therefore, firmware should first disable the endpoint, and then check that the enable succeeded (there can be a delay while a pending USB transaction completes).

To re-assign an IN endpoint, firmware should proceed using one of the following two methods.

Re-Assign an IN Endpoint – Method A

1. Stop loading data into the endpoint.
2. Wait until there is no data waiting to be transmitted to the host (Buffer states are available in the **EP_BUFF_STATE** register). Because of the USB-inherent problem discussed in the *USB r2.0*, Section 8.5.3.3, waiting for the endpoint to empty can potentially lead to deadlock.
3. Write 0 to the **EP_CFG** register *Endpoint Enable* bit.
4. Verify that the *Endpoint Enable* bit is clear.
5. Save the Endpoint registers – **EP_CFG**, **EP_IRQENB**, **EP_TRANSFER*x***, **EP_RSPSET**, and **EP_MAXPKT*x***.
6. Re-assign the endpoint.

Re-Assign an IN Endpoint – Method B

1. Stop loading data into the endpoint.
2. Write 0 to the **EP_CFG** register *Endpoint Enable* bit.
3. Verify that the *Endpoint Enable* bit is clear.
4. Read the **EP_AVAIL*x*** register.
5. Read out and store packets remaining loaded in the Endpoint buffers. Reading is accomplished by clearing the **EP_CFG** register *Endpoint Direction* bit, so the buffer data is available to **EP_DATA** and the Count is available in **EP_AVAIL*x***. Zero-Length packets should also be detected, read out, and stored. Zero-Length packets can be detected from the **EP_BUFF_STATE** register.
6. Save the Endpoint registers – **EP_CFG**, **EP_IRQENB**, **EP_TRANSFER*x***, **EP_RSPSET**, and **EP_MAXPKT*x***.
7. Re-assign the endpoint.

Re-assign an OUT Endpoint

1. Write 0 to the **EP_CFG** register *Endpoint Enable* bit.
2. Verify that the *Endpoint Enable* bit is clear.
3. If data is available in the Endpoint buffer, read out the data. The host can transmit a packet after firmware clears the *Endpoint Enable* bit, before the endpoint is disabled.
4. Save the Endpoint registers – **EP_CFG**, **EP_IRQENB**, **EP_TRANSFER*x***, **EP_RSPSET**, and **EP_MAXPKT*x***.
5. Re-assign the endpoint.

Re-assigning a Physical endpoint consists of programming the direction, type, and Logical Endpoint Number and setting the **EP_CFG** register *Endpoint Enable* bit (these operations can be executed in a single register Write operation), and loading the **EP_IRQENB**, **EP_TRANSFER*x***, **EP_RSPSET**, and **EP_MAXPKT*x*** registers. Flush the endpoint before loading data or enabling the endpoint.

Restoring **EP_RSPSET** and **EP_RSPCLR** requires two register writes, one to **EP_RSPCLR** with the logical NOT of the saved **EP_RSPSET** value, and a second write to **EP_RSPSET** with the saved **EP_RSPSET** value. Clearing the *Endpoint HALT* bit also clears the *Endpoint Toggle* bit, so the write to **EP_RSPCLR** should occur first, followed by the write to **EP_RSPSET**.

When restoring an IN endpoint with stored buffer data (refer to Re-Assign an IN Endpoint – Method B), care should be taken to correctly restore **EP_TRANSFER*x***, using one of the following methods:

- Load **EP_TRANSFER*x*** with the stored **EP_TRANSFER*x*** value, plus the count of stored data before the stored data is loaded
- Load the stored data first (and packets validated by writing 0 to **EP_TRANSFER*x***), followed by restoring **EP_TRANSFER*x*** to the stored value

5.7.3 Efficiency Considerations

Depending on the number of Virtual endpoints and the host-controller requirements, the firmware may need to prioritize Virtual endpoint re-assignment. *For example*, a simple scheme of scheduling the lowest Virtual Endpoint request each time can starve higher Logical Endpoint addresses. *Round-Robin priority* is one method available to ensure that data travels on all endpoints.

Interrupt endpoints can require special care, because the polling interval between accesses can be excessive. It is not efficient to detect the Endpoint access on an Interrupt endpoint (which was NAKed) – switch to that endpoint, then detect and switch to a different endpoint before the next Interrupt polling interval arrives. A possible solution is to “lock down” the Physical endpoint (prevent further endpoint switching) until a minimum number of packets pass through the endpoint.

5.7.4 Deadlock Considerations

Usually, the USB host controller Retries NAKed Bulk transactions in a Round-Robin priority; therefore, deadlocks do not normally occur. However, some host drivers can be susceptible to deadlocks. Evaluate NET2272 and host driver implementations, specifically for deadlock exposure.

For example, step 2 of Method A, re-assigning an IN endpoint, requires firmware to wait until the Physical endpoint is empty. (Refer to Re-Assign an IN Endpoint – Method A.) Deadlock occurs when the host is not requesting data on the (old) Logical endpoint; instead, it is waiting for a transaction on the new Logical endpoint (the Logical Endpoint firmware is trying to switch to the new Logical endpoint; however, it is prevented from doing so because the Physical endpoint is not empty).

This particular deadlock is broken by flushing the IN endpoint if the NET2272 is prepared to reload the packet(s), or if the data is discardable. Alternatively, deadlock is avoided by not loading data into the IN Logical endpoint until the host transmits an IN Token request, or by using Method B, steps 2 through 4, instead. (Refer to Re-Assign an IN Endpoint – Method B.)

Another potential deadlock source is the USB-inherent problem discussed in the *USB r2.0*, Section 8.5.3.3. In this situation, the endpoint is unable to flush the final packet of a transfer because the endpoint does not know that the host correctly received the packet, and the host cannot transmit another IN Token request on the same endpoint for an indeterminate length of time.

5.7.5 Buffer Control

The NET2272 buffers are read and written from the Local Bus. This feature is used for chip diagnostics (power-on tests) and during Virtual endpoint context switching.

When the **EP_CFG** register *Endpoint Direction* bit (bit 4) of an endpoint is set, the endpoint is an IN endpoint and **EP_DATA** is a Write-Only FIFO-style register. After data is loaded into the endpoint, the packet is validated by writing 0 to **EP_TRANSFER*x*** (as a convenience, if the upper two bytes of **EP_TRANSFER*x*** are 0, writing 0 to **EP_TRANSFER0** validates the packet with a single register write).

After a packet is validated (or both packets have been validated if the endpoint is configured to be double-buffered), the *Endpoint Direction* bit is switched to OUT (by clearing the **EP_CFG** register *Endpoint Direction* bit) and the Buffer data can be read out. The **EP_AVAIL*x*** registers indicate the number of bytes available in the buffer.

Zero-Length packets behave the same as OUT Zero-Length packets transmitted by the host, and are similarly flushed by a dummy read from the **EP_DATA** register. The presence of a Zero-Length packet is indicated by the **EP_STAT1** register *Local OUT ZLP* bit (bit 6) or by the **EP_BUFF_STATES** Buffer states.

Data is available for reading from the OUT **EP_DATA** register only after the packet is validated and while the endpoint is configured for IN. Written, unvalidated data is lost when the direction is switched to OUT.

The endpoint should be disabled before switching directions, and the disable operation checked by reading bit 7 of the **EP_CFG** register, *Endpoint Enable*. This ensures that the USB host does not read the IN packet before local firmware reads out the packet.

EP_DATA buffers can be read or written while the endpoint is disabled.

5.8 Packet Buffers

The NET2272 contains one 3-KB memory bank, allocated to Endpoint Packet buffers. Endpoint A and B buffer configuration is selected by the **LOCCTL** Configuration register *Buffer Configuration* field. Available configurations for Endpoints A and B are as follows:

- 512 bytes, double-buffered (total of 1 KB)
- 1,024 bytes, single-buffered
- 1,024 bytes, double-buffered

The combined size of all Endpoint A and B buffers cannot exceed 2 KB. Endpoint C has a 1-KB buffer, configured as two 512-byte buffers. Endpoint 0 has a 128-byte buffer, configured as two 64-byte buffers. Data is stored into the buffers in 32-bit Dwords, allowing each entry to contain between 1 and 4 bytes.

If a write to a full buffer is detected, the data is ignored. If a read from an empty buffer is detected, undefined data is presented on the Data bus.

5.8.1 OUT Endpoint Buffers

When receiving data, the NET2272 NAKs the host (indicating that the NET2272 cannot accept the data) when one of the following conditions is met:

- Buffer is full
- *NAK OUT Packets Mode* and *NAK OUT Packets* bits are set

If the packets received are of maximum size, then additional packets are received, independent of the *NAK OUT Packets Mode* bit state. This bit causes additional OUT packets to be NAKed if the last packet received was a Short packet. If the *NAK OUT Packets Mode* bit is true (Blocking mode), USB OUT transfers can overlap with the Local CPU, unloading the data in the following sequence:

1. Local CPU responds to the *Data Packet Received Interrupt* bit and reads the **EP_AVAIL*x*** register to determine the number of bytes in the current packet.
2. Local CPU clears the *Data Packet Received Interrupt* and *NAK OUT Packets* bits, allowing the next packet to be received.
3. Local CPU can now unload data from the buffer during the next USB OUT transaction.

If the *NAK OUT Packets Mode* bit is false (Non-Blocking mode), the NET2272 accepts packets as long as there is space for the complete packet in the next available buffer. There are no indications of packet boundaries when buffers in a double-buffered configuration consist of multiple packets.

5.8.2 IN Endpoint Buffers

IN packet data is written by the Local CPU or DMA controller into one of the IN Endpoint buffers. After the buffer data is validated, the data is returned to the USB host in response to an IN token. The NET2272 does not transmit more than the specified number of **EP_MAXPKT*x*** bytes (quantity indicated by the register value) per packet. After a packet is written into a double-buffered buffer and validated, the Local CPU can continue loading data for the next packet. The NET2272 automatically divides the data flow into individual packets, with a Maximum Packet Size determined by the associated **EP_MAXPKT*x*** register. This allows USB transactions to overlap with loading of data from the Local Bus.

If the buffer data is not validated, the NET2272 responds to an IN token with a NAK handshake. There are several methods for validating IN buffer data:

- For large amounts of data, the Local Bus controller can write data to the buffer as long as Buffer space is available. When there are at least the specified number of **EP_MAXPKT*x*** bytes in the buffer, the NET2272 responds to an IN token with a Data packet. If the entire Data transfer is a multiple of the specified number of **EP_MAXPKT*x*** bytes, no further action is required to validate the Buffer data. If a Zero-Length packet must be transmitted to the host, the Local CPU can write 0 to the **EP_TRANSFER*x*** register without writing additional data to the buffer.
- For moderate amounts of data (between **EP_MAXPKT*x*** and 16 MB), the **EP_TRANSFER*x*** Counter is used. This counter is initialized to the total Transfer Byte Count before data is written to the buffer. Also, the counter decrements as data is written to the buffer. When the counter reaches 0, data remaining in the buffer is validated. If 16-bit transfers are being utilized on the Local Bus, excess bytes in the last word are automatically ignored. If the last packet of a transfer contains **EP_MAXPKT*x*** bytes, the NET2272 responds to the next IN token with a Zero-Length packet.
- For small amounts of data (character-oriented applications), the data is first written to the buffer. A 0 is then written to the **EP_TRANSFER*x*** register, thereby causing the data to be validated.
- For a DMA write terminated with an EOT, the Buffer data is validated if the **DMAREQ** register *DMA Buffer Valid* bit is set, which causes a Short- or Zero-Length packet to transmit in response to the next IN token.

Note: *Up to two Short packets (fewer than the specified number of* ***EP_MAXPKTx*** *bytes) can be stored in a double-buffered Packet buffer.*

5.8.2.1 16-Bit Post-Validation

Post-validation is a technique in which data is written to the buffer before the data is validated. To post-validate an Odd-Length packet (size 2*n*+1) when operating in 16-Bit mode, use the following sequence:

1. Write 2*n* bytes, using *n* 16-bit transactions to the Endpoint buffer, by way of **EP_DATA**.
2. Change to an 8-bit bus by clearing the **LOCCTL** register *Data Width* bit.
3. Write the last byte to the Endpoint buffer by way of **EP_DATA**.
4. Post-validate the buffer by writing 0 to **EP_TRANSFER0**.
5. Return to a 16-bit bus by setting the **LOCCTL** register *Data Width* bit.

5.9 USB Test Modes

Use the **XCVRDIAG** register *Force Full Speed* and *Force High Speed* bits to force the NET2272 into Full- and High-Speed modes, respectively. These bits **must not be used in normal operation**, as they are for testing purposes only. In normal operation, the NET2272 automatically performs *USB r2.0*-Chirp Protocol negotiation with the host to determine the correct operating speed.

USB r2.0 Test mode support is provided by way of the **USBTEST** register *USB Test Mode Select* field. This field selects the appropriate USB Test mode settings. (Refer to the *USB r2.0,* Section 9.4.9, for further details.) Normally, the host transmits a SET_FEATURE request with the Test Selector in the upper byte of `wIndex`. To select the correct Test mode, copy the Test Selector into the **USBTEST** register *USB Test Mode Select* field.

USB Test mode settings are functional only when the NET2272 is in High-Speed mode. Also, if the NET2272 is operating in High-Speed mode, and the *USB Test Mode Select* field is set to non-zero, the NET2272 is prevented from switching out of High-Speed mode. Normal USB Suspend and Reset, as well as the *Force Full Speed* and *Force High Speed* bits, are ignored for test purposes.

The NET2272 can be forced into High-Speed mode (using the *Force High Speed* bit), regardless of whether the NET2272 is connected to a host controller. After choosing High-Speed mode, USB Test modes can be selected.

Most USB Test modes require no further support from the NET2272 firmware. However, the Test_Packet (100b) Test Selector requires the NET2272 to return a specific packet.

Firmware performs the following sequence to activate the test mode:

1. Sets the *USB Test Mode Select* field to 100b.
2. Flushes Endpoint 0.
3. Loads the following 53 (35h) hex byte packets into Endpoint 0:

   00 00 00 00 00 00 00 00 - 00 AA AA AA AA AA AA AA AA EE EE EE EE EE EE -

   EE EE FE FF FF FF FF FF FF FF FF FF FF FF 7F BF DF -

   EF F7 FB FD FC 7E BF DF EF F7 FB FD 7E

Validate the packet with a normal validation method, *such as* pre-validating by writing 35h into **EP_TRANSFER0** before loading the bytes, or writing 0 to **EP_TRANSFER0** after loading the bytes.

Chapter 6 Interrupt and Status Register Operation

6.1 Overview

The purpose of this chapter is to describe, in general terms, how Interrupt and Status registers operate. For in-depth details of all Interrupt bits, refer to Chapter 8, "Configuration Registers."

6.2 Interrupt Status Registers (IRQSTAT0 and IRQSTAT1)

Bits [3:0] of the **IRQSTAT0** register, *Endpoint Interrupt*, indicate whether an interrupt is pending on Endpoint C, B, A, or 0, respectively. These bits cannot be written, and can cause a Local interrupt if the corresponding **IRQENB0** register *Interrupt Enable* bits are set. Bit 7 is automatically set when a Start-of-Frame (SOF) token is received, and cleared by writing 1. Bit 7 can cause a Local interrupt if the corresponding **IRQENB0** register *Interrupt Enable* bit is set. *Interrupt* bits can be set without the corresponding *Interrupt Enable* bit being set. This allows the Local CPU to operate in a Polled and/or Interrupt-Driven environment.

IRQSTAT0 register bits [6:4] and **IRQSTAT1** register bits [7:4 and 2:1] are set when a particular event occurs in the NET2272, and cleared by writing 1 to the corresponding bit. These bits can cause a Local interrupt if the corresponding **IRQENB0** and **IRQENB1** register *Interrupt Enable* bits are set.

IRQSTAT1 register bit 3 is set when the host transmits a Suspend request; however, the bit is not typically enabled to generate an interrupt. Writing 1 clears this bit and causes the NET2272 to enter the Low-Power Suspend state.

6.3 Endpoint Response Registers (EP_RSPCLR and EP_RSPSET)

Each endpoint maintains of a pair of Endpoint Response registers. These registers determine how the NET2272 responds to various situations during USB transactions. Writing 1 to any **EP_RSPCLR** register bit clears the corresponding bit. Writing 1 to any **EP_RSPSET** register bit sets the corresponding bit. Reading either of these registers returns their current bit states.

6.4 Endpoint Status Register (EP_STAT0 and EP_STAT1)

Each endpoint maintains a pair of Endpoint Status registers. The **EP_STAT*x*** register bits are set when a particular endpoint event occurs, and cleared by writing 1 to the corresponding bit. A Local interrupt can be generated if the corresponding **EP_IRQENB** register *Interrupt Enable* bits are set. Reading the **EP_STAT*x*** registers returns their current bit states.

Chapter 7 Power Management

7.1 Overview

The NET2272 supports the USB Power Management requirements detailed in the *USB r2.0*.

7.2 USB Low-Power Suspend State

When there is a 3 ms period of USB inactivity, the *USB r2.0* requires bus-powered devices to enter into the Low-Power Suspend state. During this state, bus-powered devices can draw no more current than the maximum value listed in Table 7-1. To facilitate this, the NET2272 provides the **IRQSTAT1** register *Suspend Request Change Interrupt* and *Suspend Request Interrupt* bits. Additionally, the NET2272 allows Local Bus hardware to initiate a Device-Remote Wakeup to the USB.

Table 7-1. Bus-Powered Device Maximum Currents in Low-Power Suspend State

Device	Maximum Current
Low Powered	500 µA
High Powered	2.5 mA

7.2.1 Suspend Sequence

The typical Suspend sequence is as follows:

1. During device configuration, the Local CPU enables the **IRQSTAT1** register *Suspend Request Change Interrupt* bit to generate a Local interrupt.
2. When the USB is idle for 3 ms, the NET2272 sets the *Suspend Request Change Interrupt* bit, generating an interrupt to the Local CPU. This interrupt also occurs when the NET2272 is not connected to a host and the USB data lines are pulled to the Idle state (DP high, DM low), or the VBUS input is low.
3. The Local CPU accepts this interrupt by clearing the *Suspend Request Change Interrupt* bit, and performs the required tasks, ensuring that no more than 500 mA of current is drawn from the USB power bus.
4. The Local CPU writes a 1 to the **IRQSTAT1** register *Suspend Request Interrupt* bit to initiate the transition to the Low-Power Suspend state.
5. LCLKO continues to operate for 500 µs before the NET2272 enters the Low-Power Suspend state. This allows time for a Local CPU using LCLKO to power down.
6. A Device-Remote Wakeup event is not recognized during the 500 µs suspend delay period.

In the Low-Power Suspend state, the NET2272 oscillator shuts down, and most Output pins/balls are three-stated to conserve power. (Refer to Chapter 2, "Pin/Ball Description.") Do ***not*** allow the NET2272 Input pins/balls to float during this state. The NET2272 exits the Low-Power Suspend state by detecting a Host-Initiated or Device-Remote Wakeup.

If the NET2272 is self-powered, it can ignore the USB Suspend request and never write 1 to the *Suspend Request Interrupt* bit.

Note: *Configuration registers cannot be accessed when the NET2272 is in the Low-Power Suspend state.*

7.2.2 Host-Initiated Wakeup

The host can wake up the NET2272 by driving a Non-Idle state on the USB. The NET2272 detects the host's Wakeup request, and restarts its internal oscillator. The Host-Initiated Wakeup is recognized only when the VBUS input pin/ball is high, and the **USBCTL0** register *USB Detect Enable* and *USB Root-Port Wakeup Enable* bits are set.

7.2.3 Device-Remote Wakeup

The NET2272 hardware signals a USB Device-Remote Wakeup by driving CS# input pin/ball low. If the **USBCTL0** register *I/O Wakeup Enable* bit is set, the NET2272 restarts its Local oscillator. Two milliseconds after CS# is asserted, the Local CPU must write to the **USBCTL1** register *Generate Resume* bit. This transmits a 10-ms Wakeup signal to the USB host.

7.2.4 Resume Interrupt

When the NET2272 starts a Device-Remote or Host-Initiated Wakeup, the NET2272 can also be programmed to generate a Resume interrupt. The **IRQSTAT1** register *Resume Interrupt* bit is set when a resume is detected, and enabled to generate an interrupt with the *Resume Interrupt Enable* bit.

7.3 NET2272 Power Configuration

The *USB r2.0* defines both bus-powered and self-powered devices. A *bus-powered* device is a peripheral that derives its power from the upstream USB connector. A *self-powered* device derives its power from an external power supply.

The most significant consideration when deciding whether to build a bus- or self-powered device is power consumption. The *USB r2.0* dictates the following requirements for maximum current draw:

- Device not host-configured can draw only 100 mA from the USB connector Power pins.
- Device can draw no more than 500 mA from the USB connector Power pins.
- In the Low-Power Suspend state, however, the device can draw no more than 500 µA (or 2.5 mA for a high-power device) from the USB connector Power pins.

Warning: ***If these power considerations can be met without using an external power supply, it is recommended that the NET2272 be bus-powered; otherwise, implement a self-powered design. Harm, such as ground shorting, can occur to the NET2272 if the proper power design is not used.***

7.3.1 Self-Powered Device

Generally, a device with higher power requirements is self-powered. In a self-powered device, the NET2272 VDDC and VDDIO pins/balls are powered by the local power supply. This allows the Local Bus to continue accessing the NET2272, although the NET2272 cannot be connected to the USB. The USB connector Power pin is directly connected to the NET2272 VBUS pin/ball, and is used only to detect whether the device is connected to the USB host.

While the device is connected to the USB host, the NET2272 automatically requests the Low-Power Suspend state when appropriate. Do ***not*** power-down the NET2272 when its Local Bus is connected to a powered-up device. There are ESD protection circuits in the NET2272 that short VDDC and VDDIO pins/balls to Ground. If the VDDC and VDDIO pins/balls are not powered, they sink excessive current from the board.

7.4 NET2272 Low-Power Modes

7.4.1 USB Suspend (Unplugged from USB)

The NET2272 can draw a small amount of power when disconnected from the USB. Disconnecting from the USB is accomplished in two ways:

- Unplug the USB cable
- Clear the **USBCTL0** register *USB Detect Enable* bit

In power-sensitive applications, the Local CPU can force the NET2272 to enter the Low-Power Suspend state when disconnected from the USB by writing 1 to the *Suspend Request Interrupt* bit. The NET2272 automatically wakes up when the peripheral device is re-connected to the USB (cable is plugged in and *USB Detect Enable* bit is set).

Caution: ***Do not force the Low-Power Suspend state, unless the peripheral device is disconnected from the USB host. When the NET2272 is connected to the USB, it is a violation of the*** *USB r2.0* ***to enter this state unless the upstream port is idle for at least 3 ms.***

USB Suspend is the preferred method of suspending the NET2272, because a USB re-connection automatically causes the NET2272 to wakeup and set the *Resume Interrupt* bit.

7.4.2 Power-On Standby

The Local CPU can prevent the NET2272 from starting its oscillator upon power-up by driving a LOW into the CS# pin/ball while RESET# is asserted (LOW). In the Low-Power Suspend state, the NET2272 requires only a minor quiescent standby current.

When the peripheral device requires starting of the oscillator, the Local CPU releases the CS# pin/ball and continues to assert RESET# for a minimum of 2 ms. While the oscillator is stopped, the NET2272 cannot respond to USB requests; therefore, the oscillator must be allowed to start when the peripheral device detects a USB Connection event. The Local CPU is responsible for detecting the connection, and ending the Standby condition.

Power-On Standby is appropriate when the device’s power budget does not allow the NET2272 to be active a sufficient amount of time to shut itself down by setting the *Suspend Request Interrupt* bit.

Chapter 8 Configuration Registers

8.1 Register Description

The NET2272 occupies a 32-byte Address space that can be accessed by a Local CPU. Registers can be directly or indirectly accessed through a Pointer register. Directly accessing the registers provides higher performance, while accessing the registers through the Pointer register requires fewer physical Address pins/balls. The most commonly used registers are located at the lowest addresses, thereby providing the highest performance when using fewer than five Address bits.

Each Configuration register is organized as an 8-bit register, while the Endpoint Packet buffers can be accessed as an 8- or 16-bit port, depending on the **LOCCTL** register *Data Width* bit value.

After the NET2272 is powered-up or reset, the registers are set to their default values. Writes to unused registers are ignored, and reads from unused registers return a value of 0.

Always write register ***reserved*** bits with a 0, to maintain compatibility with future revisions.

Note: *Configuration registers cannot be accessed when the NET2272 is in the Low-Power Suspend state.*

This chapter describes the three register groups:

- Main Control
- USB Control
- Endpoint

8.2 Register Address Mapping

Table 8-1. Main Control Register Address Mapping

Address	Register Name	Register Description	Page
00h	REGADDRPTR	Indirect Register Address Pointer	62
01h	REGDATA	Indirect Register Data	62
02h	IRQSTAT0	Interrupt Status (Low Byte)	63
03h	IRQSTAT1	Interrupt Status (High Byte)	64
04h	PAGESEL	Endpoint Page Select	64
1Ch	DMAREQ	DMA Request Control	65
1Dh	SCRATCH	Scratchpad	65
20h	IRQENB0	Interrupt Enable (Low Byte)	66
21h	IRQENB1	Interrupt Enable (High Byte)	67
22h	LOCCTL	Local Bus Control	68
23h	CHIPREV_LEGACY	Legacy Silicon Revision	68
24h	LOCCTL1	Local Bus Control 1	69
25h	CHIPREV_2272	NET2272 Silicon Revision	69

Table 8-2. USB Control Register Address Mapping

Address	Register Name	Register Description	Page
18h	USBCTL0	USB Control (Low Byte)	70
19h	USBCTL1	USB Control (High Byte)	70
1Ah	FRAME0	Frame Counter (Low Byte)	71
1Bh	FRAME1	Frame Counter (High Byte)	71
30h	OURADDR	Our Current USB Address	71
31h	USBDIAG	USB Diagnostic	71
32h	USBTEST	USB Test Modes	72
33h	XCVRDIAG	Transceiver Diagnostic	72
34h	VIRTOUT0	Virtual OUT Interrupt 0	73
35h	VIRTOUT1	Virtual OUT Interrupt 1	73
36h	VIRTIN0	Virtual IN Interrupt 0	73
37h	VIRTIN1	Virtual IN Interrupt 1	73
40h	SETUP0	Setup Byte 0	74
41h	SETUP1	Setup Byte 1	74
42h	SETUP2	Setup Byte 2	74
43h	SETUP3	Setup Byte 3	74
44h	SETUP4	Setup Byte 4	75
45h	SETUP5	Setup Byte 5	75
46h	SETUP6	Setup Byte 6	75
47h	SETUP7	Setup Byte 7	75

Table 8-3. Endpoint Register Address Mapping

Address	Register Name	Register Description	Page
05h	EP_DATA	Endpoint Data	76
06h	EP_STAT0	Endpoint Status (Low Byte)	77
07h	EP_STAT1	Endpoint Status (High Byte)	78
08h	EP_TRANSFER0	IN Endpoint Transfer Count (Byte 0)	78
09h	EP_TRANSFER1	IN Endpoint Transfer Count (Byte 1)	79
0Ah	EP_TRANSFER2	IN Endpoint Transfer Count (Byte 2)	79
0Bh	EP_IRQENB	Endpoint Interrupt Enable	79
0Ch	EP_AVAIL0	Endpoint Available Count (Low Byte)	80
0Dh	EP_AVAIL1	Endpoint Available Count (High Byte)	80
0Eh	EP_RSPCLR	Endpoint Response Clear	81
0Fh	EP_RSPSET	Endpoint Response Set	82
28h	EP_MAXPKT0	Endpoint Maximum Packet Size (Low Byte)	82
29h	EP_MAXPKT1	Endpoint Maximum Packet Size (High Byte)	82
2Ah	EP_CFG	Endpoint Configuration	83
2Bh	EP_HBW	Endpoint High Bandwidth	83
2Ch	EP_BUFF_STATES	Endpoint Buffer States	84

Note: *There is a set of Endpoint registers for each endpoint. The* ***PAGESEL*** *register Page Select field selects which register set is active when the addresses in Table 11-3 are accessed.*

8.3 Numeric Register Listing

Table 8-4 lists the quantity of Address bits required to directly access a register. If only four Address bits are supplied to the NET2272, then registers that require five Address bits must be indirectly addressed, using the **REGADDRPTR** and **REGDATA** registers.

Addresses marked with (P) are paged registers, selected by the **PAGESEL** register *Page Select* field.

Table 8-4. Numeric Register Listing

Address	Required Address Bit Quantity	LD[15:8]	LD[7:0]	Register Description
00h	1		REGADDRPTR	Register Address pointer for indirect register addressing.
01h		REGDATA1	REGDATA	Register Data port for indirect register addressing.
02h	2		IRQSTAT0	Interrupt Status (low byte).
03h			IRQSTAT1	Interrupt Status (high byte).
04h	3		PAGESEL	Page Select. Select current Endpoint.
05h (P)		EP_DATA1	EP_DATA	Endpoint Data.
06h (P)			EP_STAT0	Endpoint Main Status.
07h (P)			EP_STAT1	
08h (P)	4		EP_TRANSFER0	For IN endpoint, number of bytes to transmit to the host.
09h (P)			EP_TRANSFER1	
0Ah (P)			EP_TRANSFER2	
0Bh (P)			EP_IRQENB	Endpoint Interrupt Enable.
0Ch (P)			EP_AVAIL0	For IN endpoints, number of available spaces in buffer.
0Dh (P)			EP_AVAIL1	For OUT endpoints, number of bytes in buffer.
0Eh (P)			EP_RSPCLR	Endpoint Response Clear.
0Fh (P)			EP_RSPSET	Endpoint Response Set.
10h - 17h	5		***Reserved***	
18h			USBCTL0	USB Control.
19h			USBCTL1	
1Ah			FRAME0	Frame Counter.
1Bh			FRAME1	
1Ch			DMAREQ	DMA Request Control.
1Dh			SCRATCH	General-Purpose Scratchpad.
1Eh - 1Fh			***Reserved***	

Table 8-4. Numeric Register Listing *(Cont.)*

Address	Required Address Bit Quantity	LD[15:8]	LD[7:0]	Register Description
20h	Indirect Only		IRQENB0	Interrupt Enable.
21h			IRQENB1	
22h			LOCCTL	Local Bus Control.
23h			CHIPREV_LEGACY	Legacy Chip Revision Number.
24h			LOCCTL1	Local Bus Control 1.
25h			CHIPREV_2272	NET2272 Chip Revision Number.
26h - 27h			***Reserved***	
28h (P)			EP_MAXPKT0	Endpoint Maximum Packet Size.
29h (P)			EP_MAXPKT1	
2Ah (P)			EP_CFG	Endpoint Configuration.
2Bh (P)			EP_HBW	Endpoint High Bandwidth.
2Ch (P)			EP_BUFF_STATES	Endpoint Buffer States.
2Dh - 2Fh			***Reserved***	
30h			OURADDR	Our USB Address.
31h			USBDIAG	*USB r2.0* Diagnostic.
32h			USBTEST	*USB r2.0* Test Control.
33h			XCVRDIAG	Transceiver Diagnostic.
34h			VIRTOUT0	Virtual OUT Interrupt 0.
35h			VIRTOUT1	Virtual OUT Interrupt 1.
36h			VIRTIN0	Virtual IN Interrupt 0.
37h			VIRTIN1	Virtual IN Interrupt 1.
38h - 3Fh			***Reserved***	
40h			SETUP0	Setup Byte 0.
41h			SETUP1	Setup Byte 1.
42h			SETUP2	Setup Byte 2.
43h			SETUP3	Setup Byte 3.
44h			SETUP4	Setup Byte 4.
45h			SETUP5	Setup Byte 5.
46h			SETUP6	Setup Byte 6.
47h			SETUP7	Setup Byte 7.

8.4 Main Control Registers

Register 11-1. (Address 00h; REGADDRPTR) Indirect Register Address Pointer

Bits	Description	Read	Write	Default
6:0	**Register Address** Register Address pointer used for indirect register addressing.	Yes	Yes	0h
7	***Reserved***	Yes	No	0

Register 11-2. (Address 01h; REGDATA) Indirect Register Data

Bits	Description	Read	Write	Default
15:0	**Register Data** Register Data port for indirect register addressing. For 8-bit bus widths, data is transferred on bits [7:0]. For 16-bit Buffer accesses, data is transferred on bits [15:0].	Yes	Yes	0h

Register 11-3. (Address 02h; IRQSTAT0) Interrupt Status (Low Byte)

Bits	Description	Read	Write	Default
0	**Endpoint 0 Interrupt** Conveys interrupt status for Endpoint 0. When set, the Endpoint 0 Interrupt Status register is read to determine the interrupt cause. Endpoint 0 Interrupt is set independently of the *Interrupt Enable* bit.	Yes	No	0
1	**Endpoint A Interrupt** Conveys interrupt status for Endpoint A. When set, the Endpoint A Interrupt Status register is read to determine the interrupt cause. Endpoint A Interrupt is set independently of the *Interrupt Enable* bit.	Yes	No	0
2	**Endpoint B Interrupt** Conveys interrupt status for Endpoint B. When set, the Endpoint B Interrupt Status register is read to determine the interrupt cause. Endpoint B Interrupt is set independently of the *Interrupt Enable* bit.	Yes	No	0
3	**Endpoint C Interrupt** Conveys interrupt status for Endpoint C. When set, the Endpoint C Interrupt Status register is read to determine the interrupt cause. Endpoint C Interrupt is set independently of the *Interrupt Enable* bit.	Yes	No	0
4	**Virtual Endpoint Interrupt** Set when a *Virtual Endpoint Interrupt* is set.	Yes	No	0
5	**Setup Packet Interrupt** Set when a Setup packet is received from the host. Writing 1 clears this bit.	Yes	Yes/CLR	0
6	**DMA Done Interrupt** For IN endpoints, indicates that EOT was asserted, the **EP_TRANSFER*x*** Counter reached 0 during a DMA transfer, or the corresponding **EP_TRANSFER*x*** Counter was loaded with a 0. For OUT endpoints, indicates one of the following: • EOT is asserted during the last DMA transfer • Local CPU writes a 0 to the **EP_TRANSFER*x*** register after the DMA transfer completes • Short packet is received and the Endpoint buffers are empty Writing 1 clears this bit. *DMA Done Interrupt* is set independently of the corresponding *Interrupt Enable* bit.	Yes	Yes/CLR	0
7	**SOF Interrupt** Indicates when the NET2272 receives a Start-of-Frame (SOF) packet. Writing 1 clears this bit. Set every 1 ms for Full-Speed connections, and every 125 µs for High-Speed connections.	Yes	Yes/CLR	0

Register 11-4. (Address 03h; IRQSTAT1) Interrupt Status (High Byte)

Bits	Description	Read	Write	Default
0	***Reserved***	Yes	No	0
1	**Control Status Interrupt** Set when an IN or OUT token indicates a Control Status transaction was received. Writing 1 clears this bit.	Yes	Yes/CLR	0
2	**VBUS Interrupt** When set, indicates a change occurred on the VBUS input pin/ball. Read the **USBCTL1** register for the VBUS input pin/ball's current state. Writing 1 clears this bit.	Yes	Yes/CLR	0
3	**Suspend Request Interrupt** Set when the NET2272 detects a USB Suspend request from the host. The Suspend Request state cannot be set nor cleared by writing this bit. Instead, writing 1 to this bit places the NET2272 into the Low-Power Suspend state. (Refer to Section 7.2.1.)	Yes	Yes/Suspend	0
4	**Suspend Request Change Interrupt** Set when there is a change in the *Suspend Request Interrupt* state (bit 3 of this register). Writing 1 clears this bit.	Yes	Yes/CLR	0
5	**Resume Interrupt** When set, indicates that a device resume occurred. Writing 1 clears this bit.	Yes	Yes/CLR	0
6	**Root-Port Reset Interrupt** Indicates a change in the Root-Port Reset Detector state. Writing 1 clears this bit.	Yes	Yes/CLR	0
7	**Reset Status** When set, indicates that RESET# is asserted, or a USB Root-Port Reset is currently active.	Yes	No	0

Register 11-5. (Address 04h; PAGESEL) Endpoint Page Select

Bits	Description	Read	Write	Default
1:0	**Page Select** The NET2272 uses a paged architecture for accessing the Endpoint registers. Selects which Endpoint register set can be accessed. Values: 00b = Endpoint 0 01b = Endpoint A 10b = Endpoint B 11b = Endpoint C	Yes	Yes	00b
7:2	***Reserved***	Yes	No	0h

Register 11-6. (Address 1Ch; DMAREQ) DMA Request Control

Bits	Description	Read	Write	Default
0	**DMA Endpoint Select** Determines which endpoint is accessed during a DMA Channel transfer. Values: 0 = Endpoint A 1 = Endpoint B	Yes	Yes	0
1	**DREQ Polarity** When clear, DREQ output pin/ball is active low. When set, DREQ output pin/ball is active high.	Yes	Yes	1
2	**DACK Polarity** When clear, DACK input pin/ball is active low. When set, DACK input pin/ball is active high.	Yes	Yes	0
3	**EOT Polarity** When clear, EOT input pin/ball is active low. When set, EOT input pin/ball is active high.	Yes	Yes	0
4	**DMA Control DACK** When clear, only DACK is used to perform DMA Read or Write transactions. When set, the IOR# and IOW# (or DMARD# and DMAWR# for DMA Split Bus mode) Control signals are used with DACK to perform DMA Read or Write transactions.	Yes	Yes	0
5	**DMA Request Enable** Writing 1 enables the NET2272 to request DMA cycles from a Local Bus DMA controller. If EOT input is asserted, the **EP_TRANSFER*x*** Counter reaches 0, or a Short OUT packet is received and the Endpoint buffer is empty, *DMA Request Enable* is automatically reset. A Local Bus CPU can also explicitly reset this bit to terminate a DMA transfer. If the CPU writes a 0 to the **EP_TRANSFER0** register of the endpoint selected by *Page Select*, this bit is cleared. *DMA Request Enable* can be read to determine whether a DMA transfer remains in progress.	Yes	Yes	0
6	**DMA Request** Reflects DREQ output pin/ball state, and allows a Local Bus CPU to monitor DMA transfers.	Yes	No	0
7	**DMA Buffer Valid** When clear, the buffer is not automatically validated at the end of a DMA transfer. When set, the buffer is automatically validated at the end of a DMA transfer if EOT is asserted. Applies only to IN endpoints.	Yes	Yes	0

Register 11-7. (Address 1Dh; SCRATCH) Scratchpad

Bits	Description	Read	Write	Default
7:0	**Scratch** General-purpose scratchpad register.	Yes	Yes	5Ah

Register 11-8. (Address 20h; IRQENB0) Interrupt Enable (Low Byte)

Bits	Description	Read	Write	Default
0	**Endpoint 0 Interrupt Enable** When set, enables a Local interrupt to be set when an interrupt is active on Endpoint 0.	Yes	Yes	0
1	**Endpoint A Interrupt Enable** When set, enables a Local interrupt to be set when an interrupt is active on Endpoint A.	Yes	Yes	0
2	**Endpoint B Interrupt Enable** When set, enables a Local interrupt to be set when an interrupt is active on Endpoint B.	Yes	Yes	0
3	**Endpoint C Interrupt Enable** When set, enables a Local interrupt to be set when an interrupt is active on Endpoint C.	Yes	Yes	0
4	**Virtual Endpoint Interrupt Enable** When set, enables a Local interrupt to be generated when an IN or OUT token to a Virtual endpoint is detected.	Yes	Yes	0
5	**Setup Packet Interrupt Enable** When set, enables a Local interrupt to be generated when a Setup packet is received from the host.	Yes	Yes	0
6	**DMA Done Interrupt Enable** When set, enables a Local interrupt to be generated when the **IRQSTAT0** register *DMA Done Interrupt* status bit is set.	Yes	Yes	0
7	**SOF Interrupt Enable** When set, enables a Local interrupt to be generated when the NET2272 receives a Start-of-Frame (SOF) packet.	Yes	Yes	0

Register 11-9. (Address 21h; IRQENB1) Interrupt Enable (High Byte)

Bits	Description	Read	Write	Default
0	***Reserved***	Yes	No	0
1	**Control Status Interrupt Enable** When set, enables a Local interrupt to be generated when an IN or OUT token indicates a Control Status transaction was received.	Yes	Yes	0
2	**VBUS Interrupt Enable** When set, enables a Local interrupt to be generated when a change is detected on the VBUS pin/ball.	Yes	Yes	0
3	**Suspend Request Interrupt Enable** When set, enables a Local interrupt to be generated when a USB Suspend request from the host is detected.	Yes	Yes	0
4	**Suspend Request Change Interrupt Enable** When set, enables a Local interrupt to be generated when a change in the Suspend Request Interrupt state is detected.	Yes	Yes	0
5	**Resume Interrupt Enable** When set, enables a Local interrupt to be generated when a device resume is detected.	Yes	Yes	0
6	**Root-Port Reset Interrupt Enable** When set, enables a Local interrupt to be generated when a Root-Port Reset is detected.	Yes	Yes	0
7	***Reserved***	Yes	No	0

Register 11-10. (Address 22h; LOCCTL) Local Bus Control

Bits	Description	Read	Write	Default
0	**Data Width** Controls the Local Data bus width for **EP_DATA** accesses to Endpoint buffers. To switch to 16-Bit mode, write to this register using the lower eight Data bus bits. Data Width does not affect accesses to other registers. Values: 0 = 8 bits 1 = 16 bits	Yes	Yes	0
3:1	**Local Clock Output** Controls LCLKO pin/ball frequency. Values: 000b = 0 (off) 001b = 3.75 MHz 010b = 7.5 MHz (default) 011b = 15 MHz 100b = 30 MHz 101b = 60 MHz 110b, 111b = ***Reserved***	Yes	Yes	010b
4	**DMA Split Bus Mode** When clear, I/O and DMA accesses share the same Data bus. When set, I/O accesses to Configuration registers or buffers use LD[7:0], and DMA accesses to buffers use LD[15:8], thereby splitting the Data bus for CPU and DMA accesses.	Yes	Yes	0
5	**Byte Swap** When clear, Local Data bus LD[15:0] is connected to the Endpoint buffer with no byte swapping. When set, two bytes of a 16-bit Data bus are swapped before connecting to the Endpoint buffer.	Yes	Yes	0
7:6	**Buffer Configuration** Endpoints A and B buffer configuration. *For example*, if value 01b is selected, Endpoint A is allocated a single buffer of 1,024 bytes, while Endpoint B is allocated a double buffer, two buffers of 512 bytes each. Actual packets transmitted and/or received can be less than or equal to the **EP_MAXPKT*x*** size for the selected endpoint. The **EP_MAXPKT*x*** size must be less than or equal to the allocated Buffer size. Values (in bytes): **Value** — **Endpoint A** — **Endpoint B** 00b — 512 db * — 512 db 01b — 1,024 — 512 db 10b — 1,024 — 1,024 11b — 1,024 db — Disabled * db is “double-buffered.”	Yes	Yes	00b

Register 11-11. (Address 23h; CHIPREV_LEGACY) Legacy Silicon Revision

Bits	Description	Read	Write	Default
7:0	**Legacy Chip Revision** This register returns a legacy Silicon Revision number, for use by the NET2270 firmware. ***Note:*** *The chip revision is encoded as a 2-digit Binary-Coded Decimal (BCD) value. The most significant digit is the Major Revision number, and the least significant digit is the Minor Revision number.*	Yes	No	40h

Register 11-12. (Address 24h; LOCCTL1) Local Bus Control 1

Bits	Description	Read	Write	Default
1:0	**DMA Mode** Determines DREQ behavior during DMA transactions. Values: 00b = **Slow Mode**. DREQ is de-asserted several clock periods after the DMA transaction starts, and re-asserted several clock periods after the DMA transaction is complete. This mode is compatible with the NET2270. 01b = **Fast Mode.** DREQ is de-asserted when a DMA transaction starts, and re-asserted soon after the DMA transaction is complete. This mode provides higher DMA performance. 10b = **Burst Mode.** DREQ is asserted when the *DMA Request Enable* bit is set and there is space and/or data available in the Endpoint buffer. DREQ remains asserted until the buffer becomes empty or full, the *DMA Request Enable* bit is cleared, EOT is asserted, or **EP_TRANSFER*x*** counts to 0 for an IN endpoint. 11b = ***Reserved***	Yes	Yes	00b
2	**DMA DACK Enable** When clear, the NET2272 does not recognize the DACK input pin/ball. When set, the DACK input pin/ball is enabled. *DMA DACK Enable* is automatically set when the **DMAREQ** register *DMA Request Enable* bit is set. In DMA Split Bus mode, do ***not*** clear this bit if there is a possibility that DACK can be asserted.	Yes	Yes	0
7:3	***Reserved***	Yes	No	0h

Register 11-13. (Address 25h; CHIPREV_2272) NET2272 Silicon Revision

Bits	Description	Read	Write	Default
7:0	**Chip Revision** This register returns the NET2272 Silicon Revision number. ***Note:*** *The chip revision is encoded as a 2-digit BCD value. The most significant digit is the Major Revision number, and the least significant digit is the Minor Revision number.*	Yes	No	11h

8.5 USB Control Registers

Register 11-14. (Address 18h; USBCTL0) USB Control (Low Byte)

Bits	Description	Read	Write	Default
0	***Reserved***	Yes	No	0
1	**I/O Wakeup Enable** When clear, asserting CS# does not cause a Device-Remote Wakeup. When set, *I/O Wakeup Enable* enables CS# assertion to initiate a Device-Remote Wakeup.	Yes	Yes	0
2	***Reserved***	Yes	No	0
3	**USB Detect Enable** When clear, the NET2272 does not appear to be connected to the USB host. When set, the NET2272 appears to be connected to the USB host. Do ***not*** set *USB Detect Enable* until the Configuration registers are programmed. When operating as a bus-powered device, program the registers and promptly set *USB Detect Enable* after VBUS is detected.	Yes	Yes	0
4	***Reserved***	Yes	No	0
5	**USB Root-Port Wakeup Enable** When clear, the Wakeup condition is not detected. When set, the Root-Port Wakeup condition is detected when activity is detected on the USB.	Yes	Yes	1
7:6	***Reserved***	Yes	No	00b

Register 11-15. (Address 19h; USBCTL1) USB Control (High Byte)

Bits	Description	Read	Write	Default
0	**VBUS Pin/Ball** Indicates the VBUS input pin/ball state. When set, indicates that the NET2272 is connected to the USB.	Yes	No	0
1	**USB Full Speed** When set, indicates that the transceiver is operating in Full-Speed mode (12 Mbps).	Yes	No	0
2	**USB High Speed** When set, indicates that the transceiver is operating in High-Speed mode (480 Mbps).	Yes	No	0
3	**Generate Resume** Writing 1 causes a Resume sequence to be initiated to the host if Device-Remote Wakeup is enabled. Write *Generate Resume* after a Device-Remote Wakeup is generated (CS# asserted). *Generate Resume* is self-clearing, and reading always returns a 0.	No	Yes/Resume	0
4	**Virtual Endpoint Enable** When set, enables the NET2272 Virtual Endpoint feature. A NAK is returned to the USB host if an IN token is received for a Virtual endpoint.	Yes	Yes	0
7:5	***Reserved***	Yes	No	000b

Register 11-16. (Address 1Ah; FRAME0) Frame Counter (Low Byte)

Bits	Description	Read	Write	Default
7:0	**FRAME[7:0]** Contains the Frame Counter from the most recent SOF packet.	Yes	No	0h

Register 11-17. (Address 1Bh; FRAME1) Frame Counter (High Byte)

Bits	Description	Read	Write	Default
2:0	**FRAME[10:8]** Contains the Frame Counter from the most recent SOF packet.	Yes	No	000b
7:3	***Reserved***	Yes	No	0h

Register 11-18. (Address 30h; OURADDR) Our Current USB Address

Bits	Description	Read	Write	Default
6:0	**Our Address** Contains the current device USB address. *Our Address* is cleared when a Root-Port Reset is detected. After written, the register does not update until the corresponding Control Write Transfer Status transaction successfully completes. This allows the firmware to write this field when the Setup packet is received, rather than waiting for a successful Status transaction. Refer to the *USB r2.0*, Sections 9.2.6.3 and 9.4.6.	Yes	Yes	0h
7	**Force Immediate** If *Force Immediate* is set when this register is written, the NET2272 USB address is immediately updated, without waiting for a valid Status transaction from the USB host.	No	Yes/Force	0

Register 11-19. (Address 31h; USBDIAG) USB Diagnostic

Bits	Description	Read	Write	Default
0	**Force Transmit CRC Error** When set, a CRC error is forced on the next transmitted Data packet. Inverting the most significant bit of the calculated CRC generates the CRC error. *Force Transmit CRC Error* is automatically cleared at the end of the next packet.	Yes	Yes/Set	0
1	**Prevent Transmit Bit-Stuff** When set, normal bit-stuffing is suppressed during the next transmitted Data packet. This causes a Bit-Stuffing error when six or more consecutive bits of 1 are in the data stream. Automatically cleared at the end of the next packet.	Yes	Yes/Set	0
2	**Force Receive Error** When set, an error is forced on the next received Data packet. As a result, the packet is not acknowledged. Automatically cleared at the end of the next packet.	Yes	Yes/Set	0
3	***Reserved***	Yes	No	0
4	**Fast Times** When set, the Frame Counter operates at a fast speed, for factory chip-testing purposes only.	Yes	Yes	0
7:5	***Reserved***	Yes	No	000b

Register 11-20. (Address 32h; USBTEST) USB Test Modes

Bits	Description	Read	Write	Default
2:0	**USB Test Mode Select** Valid only in High-Speed mode. Refer to the *USB r2.0*, Sections 7.1.20 and 9.4.9, for further details regarding *USB Test Mode Select*. Values: 000b = Normal Operation 001b = Test_J 010b = Test_K 011b = Test_SE0_NAK 100b = Test_Packet 101b = Test_Force_Enable 110b, 111b = ***Reserved*** ***Note:*** *If the NET2272 is operating in High-Speed mode, and the USB Test Mode Select field is set to non-zero, the NET2272 is prevented from switching out of High-Speed mode.*	Yes	Yes	000b
7:3	***Reserved***	Yes	No	0h

Register 11-21. (Address 33h; XCVRDIAG) Transceiver Diagnostic

Bits	Description	Read	Write	Default
1:0	***Reserved***	Yes	No	–
2	**Force Full Speed** When high, the NET2272 is forced into Full-Speed mode (12 Mbps).	Yes	Yes	0
3	**Force High Speed** When high, the NET2272 is forced into High-Speed mode (480 Mbps).	Yes	Yes	0
5:4	**Opmode** Indicates the transceiver operational state. Values: 00b = Normal Operation 01b = Non-Driving 10b = Disable bit-stuffing and NRZI encoding 11b = ***Reserved***	Yes	No	–
7:6	**Linestate** Indicates the DM (Linestate[1]) and DP (Linestate[0]) USB Data signal states. DM and DP values, respectively: 00b = SE0 (Single-ended zero) 01b = J state (Idle) 10b = K state (Resume) 11b = SE1 (Single-ended one)	Yes	No	–

Register 11-22. (Address 34h; VIRTOUT0) Virtual OUT Interrupt 0

Bits	Description	Read	Write	Default
0	***Reserved***	Yes	No	0
7:1	**Virtual OUT Interrupts** Set when the *Virtual Endpoint Enable* bit is set, and a Virtual endpoint received an OUT token that is not mapped to a Physical endpoint. Bit 1 corresponds to OUT Endpoint Number 1, and bit 7 corresponds to OUT Endpoint Number 7. Writing 1 to a bit clears that bit.	Yes	Yes/Clr	0h

Register 11-23. (Address 35h; VIRTOUT1) Virtual OUT Interrupt 1

Bits	Description	Read	Write	Default
7:0	**Virtual OUT Interrupts** Set when the *Virtual Endpoint Enable* bit is set, and a Virtual endpoint received an OUT token that is not mapped to a Physical endpoint. Bit 0 corresponds to OUT Endpoint Number 8, and bit 7 corresponds to OUT Endpoint Number 15. Writing 1 to a bit clears that bit.	Yes	Yes/Clr	0h

Register 11-24. (Address 36h; VIRTIN0) Virtual IN Interrupt 0

Bits	Description	Read	Write	Default
0	***Reserved***	Yes	No	0
7:1	**Virtual IN Interrupts** Set when the *Virtual Endpoint Enable* bit is set, and a Virtual endpoint received an IN token that is not mapped to a Physical endpoint. Bit 1 corresponds to IN Endpoint Number 1, and bit 7 corresponds to IN Endpoint Number 7. Writing 1 to a bit clears that bit.	Yes	Yes/Clr	0h

Register 11-25. (Address 37h; VIRTIN1) Virtual IN Interrupt 1

Bits	Description	Read	Write	Default
7:0	**Virtual IN Interrupts** Set when the *Virtual Endpoint Enable* bit is set, and a Virtual endpoint received an IN token that is not mapped to a Physical endpoint. Bit 0 corresponds to IN Endpoint Number 8, and bit 7 corresponds to IN Endpoint Number 15. Writing 1 to a bit clears that bit.	Yes	Yes/Clr	0h

Register 11-26. (Address 40h; SETUP0) Setup Byte 0

Bits	Description	Read	Write	Default
7:0	**Setup Byte 0** Provides byte 0 of the last Setup packet received. For a Standard Device request, the following `bmRequestType` information is returned. Refer to the *USB r2.0*, Section 9.3. Values: **Bit** **Description** 7 Direction: 0 = Host-to-Device; 1 = Device to Host 6:5 Type: 00b = Standard, 01b = Class, 10b = Vendor, 11b = ***Reserved*** 4:0 Recipient: 00000b = Device, 00001b = Interface, 00010b = Endpoint, 00011b = Other, 04h - 31h = ***Reserved***	Yes	No	0h

Register 11-27. (Address 41h; SETUP1) Setup Byte 1

Bits	Description	Read	Write	Default
7:0	**Setup Byte 1** Provides byte 1 of the last Setup packet received. For a Standard Device request, the following `bRequest` code information is returned. Refer to the *USB r2.0*, Section 9.4. Values: 00h = Get Status 01h = Clear Feature 02h = ***Reserved*** 03h = Set Feature 04h = ***Reserved*** 05h = Set Address 06h = Get Descriptor 07h = Set Descriptor 08h = Get Configuration 09h = Set Configuration 0Ah = Get Interface 0Bh = Set Interface 0Ch = Synch Frame	Yes	No	00h

Register 11-28. (Address 42h; SETUP2) Setup Byte 2

Bits	Description	Read	Write	Default
7:0	**Setup Byte 2** Provides byte 2 of the last Setup packet received. For a Standard Device request, the least significant byte of the `wValue` field is returned. Refer to the *USB r2.0*, Section 9.3.	Yes	No	0h

Register 11-29. (Address 43h; SETUP3) Setup Byte 3

Bits	Description	Read	Write	Default
7:0	**Setup Byte 3** Provides byte 3 of the last Setup packet received. For a Standard Device request, the most significant byte of the `wValue` field is returned. Refer to the *USB r2.0*, Section 9.3.3.	Yes	No	0h

Register 11-30. (Address 44h; SETUP4) Setup Byte 4

Bits	Description	Read	Write	Default
7:0	**Setup Byte 4** Provides byte 4 of the last Setup packet received. For a Standard Device request, the least significant byte of the `wIndex` field is returned. Refer to the *USB r2.0*, Section 9.3.4.	Yes	No	0h

Register 11-31. (Address 45h; SETUP5) Setup Byte 5

Bits	Description	Read	Write	Default
7:0	**Setup Byte 5** Provides byte 5 of the last Setup packet received. For a Standard Device request, the most significant byte of the `wIndex` field is returned. Refer to the *USB r2.0*, Section 9.3.4.	Yes	No	0h

Register 11-32. (Address 46h; SETUP6) Setup Byte 6

Bits	Description	Read	Write	Default
7:0	**Setup Byte 6** Provides byte 6 of the last Setup packet received. For a Standard Device request, the least significant byte of the `wLength` field is returned. Refer to the *USB r2.0*, Section 9.3.3.	Yes	No	0h

Register 11-33. (Address 47h; SETUP7) Setup Byte 7

Bits	Description	Read	Write	Default
7:0	**Setup Byte 7** Provides byte 7 of the last Setup packet received. For a Standard Device request, the most significant byte of the `wLength` field is returned. Refer to the *USB r2.0*, Section 9.3.3.	Yes	No	0h

8.6 Endpoint Registers

There are four sets of Endpoint registers, one for each endpoint (0, A, B, and C). To access an endpoint, set the **PAGESEL** register *Page Select* field to the needed endpoint, then read or write to an Endpoint register as defined in the following register tables. Status bits associated with an Endpoint Packet buffer (Buffer Full, Buffer Empty, and so forth), are valid only for the currently visible buffer (*that is*, the buffer that is currently being written to or read from the Local Bus).

Register 11-34. (Address 05h; EP_DATA) Endpoint Data

Bits	Description	Read	Write	Default
7:0	**Endpoint Data (Low-Order Byte)** When operating with an 8-bit bus width, bits [7:0] of this register provide the Buffer Transaction data (Read or Write). When operating with a 16-bit bus width, bits [7:0] of this register provide the low-order byte.	Yes	Yes	0h
15:8	**Endpoint Data (High-Order Byte)** When operating with a 16-bit bus width, bits [15:8] of this register provide the high-order byte.	Yes	Yes	0h

Note: *If DMA Request is enabled, then Register 11-34 accesses the Endpoint buffer selected by DMA Endpoint Select, rather than Page Select.*

Register 11-35. (Address 06h; EP_STAT0) Endpoint Status (Low Byte)

Bits	Description	Read	Write	Default
0	**Data IN Token Interrupt** Set when a Data IN token is received from the host. Writing 1 clears this bit.	Yes	Yes/Clr	0
1	**Data OUT Token Interrupt** Set when a Data OUT token is received from the host. Also set by PING tokens (High-Speed mode only). Writing 1 clears this bit.	Yes	Yes/Clr	0
2	**Data Packet Transmitted Interrupt** Set when a Data packet is transmitted from the endpoint to the host. Writing a 1 clears this bit.	Yes	Yes/Clr	0
3	**Data Packet Received Interrupt** Set when a Data packet is received from the host. Writing 1 clears this bit.	Yes	Yes/Clr	0
4	**Short Packet Transferred Interrupt** Set when the length of the last packet was less than the Maximum Packet Size (**EP_MAXPKT*x***). Writing 1 clears this bit.	Yes	Yes/Clr	0
5	**NAK OUT Packets** Set when a Short Data packet is received from the host, and the **EP_RSPSET** register *NAK OUT Packets Mode* bit is set. Writing 1 clears this bit. While this bit is set, subsequent OUT packets are acknowledged with a NAK handshake. *NAK OUT Packets* can also be controlled by the **EP_RSPCLR** and **EP_RSPSET** registers.	Yes	Yes/Clr	0
6	**Buffer Empty** For an IN endpoint, a buffer is available to the Local Bus for writing up to MAXPKT bytes. For an OUT endpoint, the currently selected buffer has a count of 0, or no buffer is available on the Local Bus (nothing to read).	Yes	No	1
7	**Buffer Full** Set when the Endpoint Packet buffer is full. For an IN endpoint, the currently selected buffer has a count of MAXPKT bytes, or no buffer is available to the Local Bus for writing (no space to write). For an OUT endpoint, there is a buffer available on the Local Bus, and there are MAXPKT bytes available to read (entire packet is available for reading).	Yes	No	0

Notes: *If DMA Request is enabled, then the Buffer Full and Buffer Empty bits correspond to the Endpoint buffer selected by DMA Endpoint Select, rather than Page Select.*

Buffer Full and Buffer Empty bits take up to 100 ns to become valid after an Endpoint buffer is written or read.

Register 11-36. (Address 07h; EP_STAT1) Endpoint Status (High Byte)

Bits	Description	Read	Write	Default
0	**Timeout** For an IN endpoint, the last USB packet transmitted was not acknowledged by the Host PC, indicating a Bus error. The Host PC expects the same packet to be re-transmitted in response to the next IN token. For an OUT endpoint, the last USB packet received contained a CRC or Bit-Stuffing error, and was not acknowledged by the NET2272. The Host PC re-transmits the packet. Writing 1 clears this bit.	Yes	Yes/Clr	0
1	**USB OUT ACK Transmitted** The last USB OUT Data packet transferred was successfully acknowledged with an ACK to the host. Writing 1 clears this bit.	Yes	Yes/Clr	0
2	**USB OUT NAK Transmitted** The last USB OUT Data packet was not accepted, but was acknowledged with a NAK to the host. Writing 1 clears this bit.	Yes	Yes/Clr	0
3	**USB IN ACK Received** The last USB IN Data packet transferred was successfully acknowledged with an ACK from the host. Writing 1 clears this bit.	Yes	Yes/Clr	0
4	**USB IN NAK Transmitted** The last USB IN packet was not provided, but was acknowledged with a NAK. Writing 1 clears this bit.	Yes	Yes/Clr	0
5	**USB STALL Transmitted** The last USB packet was not accepted or provided because the endpoint was stalled, but was acknowledged with a STALL. Writing 1 clears this bit.	Yes	Yes/Clr	0
6	**Local OUT ZLP** When set, indicates that the current Local buffer contains a Zero-Length packet.	Yes	No	0
7	**Buffer Flush** Writing 1 to this bit causes the Packet buffer to be flushed and the corresponding **EP_AVAIL*x*** register to be cleared. *Buffer Flush* is self-clearing and always written after an Endpoint configuration (*such as* direction, address) is changed. Do ***not*** assert during DMA Split Bus mode if the *Page Select* field is selecting another endpoint.	No	Yes/Flush	0

Register 11-37. (Address 08h; EP_TRANSFER0) IN Endpoint Transfer Count (Byte 0)

Bits	Description	Read	Write	Default
7:0	**EP_TRANSFER[7:0]** For IN endpoints, determines the number of bytes to transmit to the host. Write before Packet data is written to the buffer. When the count reaches 0, remaining buffer data is validated. Writing 0 to **EP_TRANSFER0** when **EP_TRANSFER1** and **EP_TRANSFER2** have a value of 0 validates the buffer contents, regardless of the *Auto Validate* bit state. If the buffer is empty, writing 0 to **EP_TRANSFER0** validates a Zero-Length packet. ***Note:*** *Validation takes approximately 100 ns.* For OUT endpoints, counter is cleared when the *NAK OUT Packets* bit is cleared (**EP_RSPCLR**, bit 7). This counter is incremented for every byte read from the Packet buffer. If 16-Bit mode is selected, and only one of the two bytes is valid, the counter increments only by 1.	Yes	Yes	0h

Register 11-38. (Address 09h; EP_TRANSFER1) IN Endpoint Transfer Count (Byte 1)

Bits	Description	Read	Write	Default
7:0	**EP_TRANSFER[15:8]** Refer to Register 11-37 for description.	Yes	Yes	0h

Register 11-39. (Address 0Ah; EP_TRANSFER2) IN Endpoint Transfer Count (Byte 2)

Bits	Description	Read	Write	Default
7:0	**EP_TRANSFER[23:16]** Refer to Register 11-37 for description.	Yes	Yes	0h

Register 11-40. (Address 0Bh; EP_IRQENB) Endpoint Interrupt Enable

Bits	Description	Read	Write	Default
0	**Data IN Token Interrupt Enable** When set, enables a Local interrupt to be set when a Data IN token is received from the host.	Yes	Yes	0
1	**Data OUT Token Interrupt Enable** When set, enables a Local interrupt to be set when a Data OUT token is received from the host.	Yes	Yes	0
2	**Data Packet Transmitted Interrupt Enable** When set, enables a Local interrupt to be set when a Data packet is transmitted to the host.	Yes	Yes	0
3	**Data Packet Received Interrupt Enable** When set, enables a Local interrupt to be set when a Data packet is received from the host.	Yes	Yes	0
4	**Short Packet Transferred Interrupt Enable** When set, enables a Local interrupt to be set when a Short Data packet is transferred to and/or from the host.	Yes	Yes	0
7:5	***Reserved***	Yes	No	000b

Register 11-41. (Address 0Ch; EP_AVAIL0) Endpoint Available Count (Low Byte)

Bits	Description	Read	Write	Default
7:0	**EP_AVAIL[7:0]** For IN endpoints, returns the number of empty bytes in the Packet buffer. Values range from 0 (full) to 1,024 (empty) bytes. *EP_AVAIL[7:0]* is updated after 2 bytes are written to the buffer, or when the buffer is validated. For OUT endpoints, returns the number of valid bytes in the Endpoint Packet buffer. Values range from 0 (empty) to 1,024 (full) bytes. If only the low byte of *EP_AVAIL[7:0]* is read, the entire 11-bit field is frozen until the upper byte is read.	Yes	No	0h

Note: *If DMA Request is enabled, then the value in Register 11-41 corresponds to the Endpoint buffer selected by DMA Endpoint Select, rather than Page Select.*

Register 11-42. (Address 0Dh; EP_AVAIL1) Endpoint Available Count (High Byte)

Bits	Description	Read	Write	Default
2:0	**EP_AVAIL[10:8]** Refer to Register 11-41 for description.	Yes	No	000b
7:3	***Reserved***	Yes	No	0h

Register 11-43. (Address 0Eh; EP_RSPCLR) Endpoint Response Clear

Bits	Description	Read	Write	Default
0	**Endpoint Halt** Used to clear the *Endpoint Stall* bit. When an Endpoint Set Feature Standard request to the *Halt* bit is detected by the Local CPU, the Local CPU must write 1 to *Endpoint Halt.* Reading this bit returns the *Endpoint Halt* bit current state. For Endpoint 0, the *Halt* bit is automatically cleared when another Setup packet is received. For Endpoint 0, the *Control Status Stage Handshake* bit should be cleared when the *Endpoint Halt* bit is set. Writing a 1 to this bit also clears the *Endpoint Toggle* bit.	Yes	Yes/Clr	0
1	**Endpoint Toggle** Used to clear the *Endpoint Data Toggle* bit. (Refer to the *USB r2.0*, Section 8.6.) Reading *Endpoint Toggle* returns the *Endpoint Data Toggle* bit current state. Under normal operation, the toggle bit is automatically controlled; therefore, the Local CPU need not use this bit. *Endpoint Toggle* is also cleared when the *Endpoint Halt* bit is cleared.	Yes	Yes/Clr	0
2	**NAK OUT Packets Mode** Used only for OUT endpoints. When *NAK OUT Packets Mode* is true, the *NAK OUT Packets* bit is set whenever *NAK OUT Packets Mode* receives a Short packet.	Yes	Yes/Clr	1
3	**Control Status Stage Handshake** Used only for Endpoint 0. Automatically set when a Setup packet is detected. While the bit is set, a Control Status stage is acknowledged with a NAK. When cleared, returns the proper response to the host (ACK for Control Reads, and Zero-Length packets for Control Writes).	Yes	Yes/Clr	0
4	**Interrupt Mode** Used for Interrupt endpoints. For normal interrupt data, set to 0 and follow standard data toggle protocol. When *Interrupt Mode* is used for isochronous rate feedback information, set high. In this mode, the *Data Toggle* bit is changed after each packet is transmitted to the host without regard to handshaking. (Refer to Section 5.6.4.1 and Section 5.6.4.2.) No packet Retries are performed in Rate Feedback mode.	Yes	Yes/Clr	0
5	**Auto Validate** When set, allows automatic validation of maximum-size packets. *That is*, there are **EP_MAXPKT*x*** bytes in the Endpoint buffer, and the data is returned to the USB host in response to the next IN token without being manually validated by the Local CPU. This is the normal operation mode for Endpoint transactions and the default state for this bit. When cleared, packets must be manually validated.	Yes	Yes/Clr	1
6	**Hide Status Stage** When set, the following bits are ***not*** set for Status Stage packets: *Data IN Token Interrupt, Data OUT Token Interrupt, Data Packet Received Interrupt*, *Data Packet Transmitted Interrupt*, *Short Packet Transferred Interrupt, USB OUT ACK Transmitted, USB OUT NAK Transmitted, USB IN ACK Received,* and *USB IN NAK Transmitted.* *Hide Status Stage* is not used for normal operation – it is intended for special applications, as determined by the designer.	Yes	Yes/Clr	0
7	**Alt NAK OUT Packets** Set when a Short Data packet is received from the host by this endpoint, and *NAK OUT Packets Mode* bit is set. If *Alt NAK OUT Packets* is set and another OUT token is received, a NAK is returned to the host if another OUT packet is transmitted to this endpoint. Can also be cleared by the **EP_STAT0** register *NAK OUT Packets* bit.	Yes	Yes/Clr	0

Note: *Writing 1 to bits [7:0] clears the corresponding register bits.*

Register 11-44. (Address 0Fh; EP_RSPSET) Endpoint Response Set

Bits	Description	Read	Write	Default
0	**Endpoint Halt**	Yes	Yes/Set	0
1	**Endpoint Toggle**	Yes	Yes/Set	0
2	**NAK OUT Packets Mode**	Yes	Yes/Set	1
3	**Control Status Stage Handshake**	Yes	Yes/Set	0
4	**Interrupt Mode**	Yes	Yes/Set	0
5	**Auto Validate**	Yes	Yes/Set	1
6	**Hide Status Stage**	Yes	Yes/Set	0
7	**Alt NAK OUT Packets**	Yes	Yes/Set	0

Notes: *Writing 1 to bits [7:0] sets the corresponding register bits.*

Refer to Register 11-43 for bit descriptions.

Register 11-45. (Address 28h; EP_MAXPKT0) Endpoint Maximum Packet Size (Low Byte)

Bits	Description	Read	Write	Default

7:0	**EP_MAXPKT[7:0]**
	Determines Endpoint Maximum Packet Size.
		Yes	Yes	EP0 = 64
				EPA = 512
				EPB = 512
				EPC = 512

Register 11-46. (Address 29h; EP_MAXPKT1) Endpoint Maximum Packet Size (High Byte)

Bits	Description	Read	Write	Default

2:0	**EP_MAXPKT[10:8]**
	Determines Endpoint Maximum Packet Size.
		Yes	Yes	EP0 = 64
				EPA = 512
				EPB = 512
				EPC = 512

4:3	**Additional Transaction Opportunities** Determines the number of additional transaction opportunities per microframe for High-Speed Isochronous and Interrupt endpoints. Values: 00b = None (One transaction per microframe) 01b = One additional (Two per microframe) 10b = Two additional (Three per microframe) 11b = ***Reserved***	Yes	Yes	00b
7:5	***Reserved***	Yes	No	000b

Register 11-47. (Address 2Ah; EP_CFG) Endpoint Configuration

Bits	Description	Read	Write	Default
3:0	**Endpoint Number** Selects the Endpoint Number. Valid numbers are 0 to 15. *Endpoint Number* has no effect on Endpoint 0, which always retains an Endpoint Number of 0.	Yes	Yes	0h
4	**Endpoint Direction** Chooses the Endpoint direction selected by the *Page Select* field. EP_DIR = 0 defines Host OUT to Device, while EP_DIR = 1 defines Host IN from Device. Endpoint 0 is bi-directional, and uses *Endpoint Direction* for a test mode. When set, Endpoint Packet buffers can be read for diagnostics. For Endpoint 0, *Endpoint Direction* is dynamic, and depends on the direction bit in the last Setup packet. ***Note:*** *A maximum of one OUT and IN endpoint is allowed for each Endpoint Number.*	Yes	Yes	0
6:5	**Endpoint Type** Selects endpoint type. Endpoint 0 is forced to a Control type. Values: 00b = ***Reserved*** 01b = Isochronous 10b = Bulk 11b = Interrupt	Yes	Yes	00b
7	**Endpoint Enable** When set, enables endpoint. Has no effect on Endpoint 0, which is always enabled. When cleared, does not read as a 0 until all pending USB transactions on the endpoint are complete.	Yes	Yes	0

Note: *For Endpoint 0, all fields in Register 11-47, except Endpoint Direction, are assigned to fixed values, and are* ***Reserved****.*

Register 11-48. (Address 2Bh; EP_HBW) Endpoint High Bandwidth

Bits	Description	Read	Write	Default
1:0	**High-Bandwidth OUT Transaction PID** Provides the PID of the last high bandwidth OUT packet received. This field is stable when the *Data Packet Received Interrupt* bit is set, and remains stable until another OUT packet is received, which is based on the currently active buffer. Values: 00b = DATA0 01b = DATA1 10b = DATA2 11b = MDATA	Yes	No	00b
7:2	***Reserved***	Yes	No	0h

Register 11-49. (Address 2Ch; EP_BUFF_STATES) Endpoint Buffer States

Bits	Description	Read	Write	Default
1:0	**Buffer A State** Provides the current Endpoint Buffer A state. Values: 00b = Buff_Free – Buffer empty, free to assign 01b = Buff_Valid – Buffer has valid packet, waiting to move to Local Bus or USB host 10b = Buff_Lcl – Buffer assigned to Local Bus 11b = Buff_Usb – Buffer assigned to USB host	Yes	No	–
3:2	**Buffer B State** Provides the current Endpoint Buffer B state. Values: 00b = Buff_Free – Buffer is empty, free to assign 01b = Buff_Valid – Buffer has valid packet, waiting to move to Local Bus or USB host 10b = Buff_Lcl – Buffer assigned to Local Bus 11b = Buff_Usb – Buffer assigned to USB host	Yes	No	–
7:4	***Reserved***	Yes	No	0h

Notes: *IN packets move from Lcl → Valid → Usb → Free*

OUT packets move from Usb → Valid → Lcl → Free

If an endpoint is double-buffered, the buffers are never in the same state. If an endpoint is single-buffered, the unused (B) buffer is locked into the Buff_Free state.

8.7 NET2272 Register Differences from NET2270 Registers

Table 8-5 lists the NET2272 register additions and removal from the NET2270 registers.

Table 8-5. NET2272 Register Differences from NET2270 Registers

Bits	Registers
Additions	
• **EP_STAT1**[6] *Local OUT ZLP* status • **IRQENB0**[4] *Virtual Endpoint Interrupt Enable* • **IRQSTAT0**[4] *Virtual Endpoint Interrupt* status • **LOCCTL1**[1:0] *DMA Mode* • **USBCTL1**[4] *Virtual Endpoint Enable*	• **EP_BUFF_STATES** Endpoint Buffer States • **EP_HBW** Endpoint High Bandwidth • **LOCCTL1** Local Bus Control 1 • **VIRTIN0** Virtual IN Interrupt 0 • **VIRTIN1** Virtual IN Interrupt 1 • **VIRTOUT0** Virtual OUT Interrupt 0 • **VIRTOUT1** Virtual OUT Interrupt 1
Removal	
• **USBDIAG**[5] *Force Bi-directional to Inputs*	–

Chapter 9 USB Standard Device Requests

9.1 Overview

Standard Device requests must be supported by Endpoint 0. (Refer to the *USB r2.0*, Chapter 9.) The Local Bus CPU decodes the Setup packets for Endpoint 0 and generates a response, based on the information in the following tables.

Table 9-1. Standard Request Codes

`bRequest`	Value
Get_Status	0h
Clear_Feature	1h
Reserved	2h
Set_Feature	3h
Reserved	4h
Set_Address	5h
Get_Descriptor	6h
Set_Descriptor	7h
Get_Configuration	8h
Set_Configuration	9h
Get_Interface	Ah
Set_Interface	Bh
Synch_Frame	Ch

Table 9-2. Descriptor Types

Descriptor Type	Value
Device	1h
Configuration	2h
String	3h
Interface	4h
Endpoint	5h
Device Qualifier	6h
Other_Speed_Configuration	7h
Interface Power	8h

9.2 Control Write Transfers

Note: *The Local CPU must write the new device address into the* ***USBADDR*** *Configuration register.*

Table 9-3. Set Address (0 Bytes)

Offset	Number of Bytes	Description	Recommended Value
–	0	Sets USB address of device. Values: `wValue` = Device address `wIndex` = 0 `wLength`= 0	–

Note: *The Local CPU must keep track of the Configuration value.*

Table 9-4. Set Configuration (0 Bytes)

Offset	Number of Bytes	Description	Recommended Value
–	0	Sets device configuration. **Values:** `wValue` = Configuration value `wIndex` = 0 `wLength`= 0	–

Note: *The Local CPU must keep track of the Interface value.*

Table 9-5. Set Interface (0 Bytes)

Offset	Number of Bytes	Description	Recommended Value
–	0	Selects alternate setting for specified interface. **Values:** `wValue` = Alternate setting `wIndex` = Specified interface `wLength`= 0	–

Note: *The Local CPU must keep track of the Device-Remote Wakeup Enable state.*

Table 9-6. Device Clear Feature (0 Bytes)

Offset	Number of Bytes	Description	Recommended Value

–	0	Clear the selected device feature.
		Values:
		wValue = Feature selector
		wIndex = 0
		wLength= 0
		FS = 1 → Device-Remote Wakeup (disable)
			–

Table 9-7. Device Set Feature (0 Bytes)

Offset	Number of Bytes	Description	Recommended Value

–	0	Set the selected device feature.
		Values:
		wValue = Feature selector
		wLength= 0
		wIndex Values:
		FS = 1 → Device-Remote Wakeup (enable), wIndex = 0
		FS = 2 → Test mode, wIndex = Specifies Test mode
			–

Table 9-8. Endpoint Set Feature (0 Bytes)

Offset	Number of Bytes	Description	Recommended Value

–	0	Set the selected endpoint feature.
		Values:
		wValue = Feature selector
		wIndex = Endpoint Number
		wLength= 0
		FS = 0 → Endpoint halt (sets *Endpoint Halt* bit)
		Note: *The Local CPU must set the Endpoint Halt bit by writing to the* ***EP_RSPSET*** *register Endpoint Halt bit.*
			–

Table 9-9. Endpoint Clear Feature (0 Bytes)

Offset	Number of Bytes	Description	Recommended Value

–	0	Clear the selected endpoint feature.
		Values:
		wValue = Feature selector
		wIndex = Endpoint Number
		wLength= 0
		FS = 0 → Endpoint halt (clears *Endpoint Halt* bit)
		Note: *The Local CPU must clear the Endpoint Halt bit by writing to the* ***EP_RSPCLR*** *register Endpoint Halt bit.*
			–

9.3 Control Read Transfers

Table 9-10. Get Device Status (2 Bytes)

Offset	Number of Bytes	Description	Recommended Value
0h	2	**Bits** **Description** 0 = Power supply good in Self-Powered mode 1 = Device-Remote Wakeup enabled [15:2] = ***Reserved***	Determined by Local CPU

Table 9-11. Get Interface Status (2 Bytes)

Offset	Number of Bytes	Description	Recommended Value
0h	2	**Bits [15:0] =** ***Reserved***	0000h

Table 9-12. Get Endpoint Status (2 Bytes)

Offset	Number of Bytes	Description	Recommended Value
0h	2	**Bits** **Description** 0 = Endpoint halted [15:1] = ***Reserved***	Determined by Local CPU

Table 9-13. Get Device Descriptor (18 Bytes)

Offset	Number of Bytes	Description	Recommended Value
0h	1	Length	12h
1h	1	Type (device)	01h
2h	2	*USB r2.0* Release Number	0200h
4h	1	Class Code	FFh
5h	1	Sub Class Code	00h
6h	1	Protocol	00h
7h	1	Maximum Endpoint 0 Packet Size	40h
8h	2	Vendor ID	0525h
10h	2	Product ID	2272h
12h	2	Device Release Number	0110h
14h	1	Index of string descriptor describing manufacturer	01h
15h	1	Index of string descriptor describing product	02h
16h	1	Index of string descriptor describing serial number	00h (not enabled)
17h	1	Number of configurations	Determined by Local CPU

Table 9-14. Get Device Qualifier (10 Bytes)

Offset	Number of Bytes	Description	Recommended Value
0h	1	Length	0Ah
1h	1	Type (device qualifier)	06h
2h	2	*USB r2.0* Release Number	0200h
4h	1	Class Code	FFh
5h	1	Sub Class Code	00h
6h	1	Protocol	00h
7h	1	Maximum Endpoint 0 Packet Size for other speed	40h
8h	1	Number of other-speed configurations	Determined by Local CPU
9h	1	***Reserved***	00h

9.3.1 Get Other_Speed_Configuration Descriptor

The structure of Other_Speed_Configuration is identical to a Configuration descriptor, except that the `bDescriptorType` is 7 instead of 2.

9.3.2 Get Configuration Descriptor

The NET2272 supports a variety of configurations, interfaces, and endpoints, each of which is defined by the descriptor data returned to the USB host. The Local CPU is responsible for providing this data to the NET2272 when requested by the host.

The following example contains one configuration, and two interfaces:

- The first interface defines one Bulk OUT endpoint at Address 1 with a Full-Speed Maximum Packet Size of 512 bytes and one Interrupt IN endpoint at Address 82h (Endpoint Number = 2) with a Maximum Packet Size of 8 bytes.
- The second interface defines one Bulk OUT endpoint at Address 3 with a Maximum Packet Size of 512 bytes.

Note: *All interface and Endpoint descriptors are returned in response to a Get Configuration descriptor request, and for this example, 48 bytes are returned.*

Table 9-15. Get Configuration Descriptor

Offset	Number of Bytes	Description	Recommended Value
Configuration Descriptor			
0h	1	Length	09h
1h	1	Type (configuration)	02h
2h	2	Total length returned for this configuration	0030h
4h	1	Number of Interfaces	02h
5h	1	Number of this configuration	01h
6h	1	Index of string descriptor describing this configuration	00h
7h	1	Attributes **Bits** **Description** [4:0] = ***Reserved*** 5 = Device-Remote Wakeup 6 = Self-Powered 7 = 1	Determined by Local CPU
8h	1	Maximum USB power required (in 2 mA units)	Determined by Local CPU
Interface 0 Descriptor			
0h	1	Descriptor Size, in bytes	09h
1h	1	Type (interface)	04h
2h	1	Number of this interface	00h
3h	1	Alternate Interface	00h
4h	1	Number of endpoints this interface uses (excluding Endpoint 0)	02h
5h	1	Class Code	FFh
6h	1	Sub Class Code	00h
7h	1	Device Protocol	00h
8h	1	Index of string descriptor describing this interface	00h

Continued …

Table 9-15. Get Configuration Descriptor *(Cont.)*

Offset	Number of Bytes	Description	Recommended Value
Bulk OUT Endpoint 1 Descriptor			
0h	1	Descriptor Size	07h
1h	1	Descriptor Type (endpoint)	05h
2h	1	Endpoint Address **Bits** **Description** [3:0] = Endpoint Number [6:4] = ***Reserved*** 7 = Direction (1 = IN, 0 = OUT)	01h
3h	1	Endpoint Attributes **Bits** **Description** [1:0] Values: 00b = Control 01b = Isochronous 10b = Bulk 11b = Interrupt [7:2] = ***Reserved***	02h
4h	2	**Bits [10:0]** = Maximum Endpoint Packet Size	0200h (HS)
6h	1	Maximum Endpoint NAK rate	Determined by Local CPU
Interrupt IN Endpoint 2 Descriptor			
0h	1	Descriptor Size	07h
1h	1	Descriptor Type (endpoint)	05h
2h	1	Endpoint Address **Bits** **Description** [3:0] = Endpoint Number [6:4] = ***Reserved*** 7 = Direction (1 = IN, 0 = OUT)	82h
3h	1	Endpoint Attributes **Bits** **Description** [1:0] Values: 00b = Control 01b = Isochronous 10b = Bulk 11b = Interrupt [7:2] = ***Reserved***	03h
4h	2	Bits [10:0] = Maximum Endpoint Packet Size	0008h
6h	1	Interval for polling endpoint	Determined by Local CPU

Continued …

Table 9-15. Get Configuration Descriptor *(Cont.)*

Offset	Number of Bytes	Description	Recommended Value
Interface 1 Descriptor			
0h	1	Descriptor Size, in bytes	09h
1h	1	Type (interface)	04h
2h	1	Number of this interface	01h
3h	1	Alternate Interface	00h
4h	1	Number of endpoints this interface uses (excluding Endpoint 0)	01h
5h	1	Class Code	FFh
6h	1	Sub Class Code	00h
7h	1	Device Protocol	00h
8h	1	Index of string descriptor describing this interface	00h
Bulk OUT Endpoint 3 Descriptor			
0h	1	Descriptor Size	07h
1h	1	Descriptor Type (endpoint)	05h
2h	1	Endpoint Address **Bits** **Description** [3:0] = Endpoint Number [6:4] = ***Reserved*** 7 = Direction (1 = IN, 0 = OUT)	03h
3h	1	Endpoint Attributes **Bits** **Description** [1:0] Values: 00b = Control 01b = Isochronous 10b = Bulk 11b = Interrupt [7:2] = ***Reserved***	02h
4h	2	Bits [10:0] = Maximum Endpoint Packet Size	0200h
6h	1	Maximum Endpoint NAK rate	Determined by Local CPU

Table 9-16. Get String Descriptor 0 (4 Bytes)

Offset	Number of Bytes	Description	Recommended Value
0h	1	Descriptor Size, in bytes	04h
1h	1	Descriptor Type (string)	03h
2h	2	Language ID (English = 09h, U.S. = 04h)	0409h

Table 9-17. Get String Descriptor 1 (30 Bytes)

Offset	Number of Bytes	Description	Recommended Value
0h	1	Descriptor Size, in bytes	1Eh
1h	1	Descriptor Type (string)	03h
2h	28	Manufacturer Descriptor (text string is encoded in UNICODE)	“PLX Technology”

Table 9-18. Get String Descriptor 2 (66 Bytes)

Offset	Number of Bytes	Description	Recommended Value
0h	1	Descriptor Size, in bytes	42h
1h	1	Descriptor Type (string)	03h
2h	64	Product Descriptor (text string is encoded in UNICODE)	“NET2272 USB Peripheral Controller”

Table 9-19. Get String Descriptor 3 (10 Bytes)

Offset	Number of Bytes	Description	Recommended Value
0h	1	Descriptor Size, in bytes	0Ah
1h	1	Descriptor Type (string)	03h
2h	8	Serial Number Descriptor (text string is encoded in UNICODE)	1001h

Table 9-20. Get Configuration (1 Byte)

Offset	Number of Bytes	Description	Recommended Value
0h	1	Returns current device configuration	00h or currently selected configuration

Table 9-21. Get Interface (1 Byte)

Offset	Number of Bytes	Description	Recommended Value
0h	1	Returns current alternate setting for the specified interface	00h or currently selected interface

Chapter 10 NAND Tree Test

10.1 NAND Tree

The VBUS pin/ball is the input to the NAND tree, and the output is observed on the LCLKO pin/ball. The VBUS pin/ball is inverted into the NAND tree; therefore, hold VBUS low while the other pins/balls are being tested. The NAND gates are tied together, with each output feeding the input of the next. A *walking zeros* pattern is then applied to the NET2272 pins/balls. All pins/balls are initially set to logic one (except VBUS), and a logic zero is then *walked* (asserted) on each input in succession, while all other inputs remain at logic one. The net result of this pattern is a series of alternating ones and zeros that are observed on the LCLKO pin/ball. Failure to observe the alternating pattern indicates a connectivity failure between the Input pin/ball and ASIC bonding pad for the corresponding pattern location.

10.2 NAND Tree Connections

Enable the NAND tree by setting the TEST, LA3, and LA4 pins/balls high. This also forces all bi-directional pins/balls to the input direction. The NAND Tree Connection sequence is as follows:

1. VBUS (input, inverted)
2. CS#, IOW#, IOR#, RESET#, ALE, EOT, DACK, DMARD#, LD15, LD14, LD13, LD12, LD11, LD10, LD9, LD8, LD7, LD6, LD5, LD4, LD3, LD2, LD1, LD0, DMAWR#, LA0, LA1, LA2
3. LCLKO (output)

Chapter 11 Electrical Specifications

11.1 Absolute Maximum Ratings

Warning: ***Conditions that exceed the Absolute Maximum limits can destroy the device.***

Table 11-1. Absolute Maximum Ratings

Symbol	Parameter	Conditions	Min	Max	Unit
AVDD, PVDD, VDD25, VDDC	2.5V Supply Voltages	With respect to Ground	-0.5	+3.6	V
VDD33	3.3V Supply Voltage	With respect to Ground	-0.5	+4.6	V
VDDIO	I/O Supply Voltage	With respect to Ground	-0.5	+4.6	V
V_I	DC Input Voltage	3.3V buffer	-0.5	+4.6	V
		5V tolerant buffer	-0.5	+6.6	V
I_{OUT}	DC Output Current (Per Pin/Ball)	3 mA buffer	-10	+10	mA
		6 mA buffer	-20	+20	mA
		12 mA buffer	-40	+40	mA
T_{STG}	Storage Temperature	No bias	-65	+150	°C
T_{AMB}	Ambient Temperature	Under bias	0	+85	°C
V_{ESD}	ESD Rating	R = 1.5K Ohms, C = 100 pF	–	2	KV

11.2 Recommended Operating Conditions

Warning: ***Conditions that exceed the Operating limits can cause the NET2272 to incorrectly function.***

Table 11-2. Recommended Operating Conditions

Symbol	Parameter	Conditions	Min	Max	Unit
AVDD, PVDD, VDD25, VDDC	2.5V Supply Voltages		2.2	2.6	V
VDD33	3.3V Supply Voltage		3.2	3.5	V
VDDIO	I/O Supply Voltage		1.7	3.5	V
V_N	Negative Trigger Voltage	3.3V buffer	0.8	1.7	V
		5V tolerant buffer	0.8	1.7	V
V_P	Positive Trigger Voltage	3.3V buffer	1.3	2.4	V
		5V tolerant buffer	1.3	2.4	V
V_{IL}	Low-Level Input Voltage	3.3V buffer	0	0.7	V
		5V tolerant buffer	0	0.8	V
V_{IH}	High-Level Input Voltage	3.3V buffer	0.5 * VDDIO	VDDIO	V
		5V tolerant buffer	2.0	5.5	V
I_{OL}	Low-Level Output Current	3 mA buffer (V_{OL} = 0.4)	–	3	mA
		6 mA buffer (V_{OL} = 0.4)	–	6	mA
		12 mA buffer (V_{OL} = 0.4)	–	12	mA
I_{OH}	High-Level Output Current	3 mA buffer (V_{OH} = 2.4)	–	-3	mA
		6 mA buffer (V_{OH}= 2.4)	–	-6	mA
		12 mA buffer (V_{OH} = 2.4)	–	-12	mA
T_A	Operating Temperature	Commercial temperature	0	85	°C
t_R	Input Rise Times	Normal input	0	200	ns
t_F	Input Fall Time		0	200	ns
t_R	Input Rise Times	Schmitt Trigger input	0	10	ms
t_F	Input Fall Time		0	10	ms

11.3 DC Specifications

11.3.1 1.8V VDDIO Core DC Specifications

Operating Conditions – VDDC = 2.37 to 2.63V, VDDIO = 1.71 to 1.89V, VDD33 = 3.13 to 3.47V, T_A = 0 to 85°C

Typical Values – VDDC = 2.5V, VDDIO = 1.8V, VDD33 = 3.3V, T_A = 25°C

Table 11-3. 1.8V VDDIO – Disconnected from USB (NET2272 Still Active)

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
I_{VDDIO}	1.8V I/O Supply Current	VDDIO = 1.8V		467	760	µA
I_{VDDC}, I_{VDD25}, I_{PVDD}, I_{AVDD}	2.5V Core Supply Current	VDDC, VDD25, PVDD, AVDD = 2.5V		37	45	mA
I_{VDD33}	3.3V Supply Current	VDD33 = 3.3V		0.08	0.1	µA

Table 11-4. 1.8V VDDIO – Connected to USB (High Speed)

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
I_{VDDIO}	1.8V I/O Supply Current	VDDIO = 1.8V		467	760	µA
I_{VDDC}, I_{VDD25}, I_{PVDD}, I_{AVDD}	2.5V Core Supply Current	VDDC, VDD25, PVDD, AVDD = 2.5V		52.3	58	mA
I_{VDD33}	3.3V Supply Current	VDD33 = 3.3V		1.7	2	mA

Table 11-5. 1.8V VDDIO – Active (High Speed)

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
I_{VDDIO}	1.8V I/O Supply Current	VDDIO = 1.8V		700	1200	µA
I_{VDDC}, I_{VDD25}, I_{PVDD}, I_{AVDD}	2.5V Core Supply Current	VDDC, VDD25, PVDD, AVDD = 2.5V		53.7	61	mA
I_{VDD33}	3.3V Supply Current	VDD33 = 3.3V		5.46	6	mA

Table 11-6. 1.8V VDDIO – Connected to USB (Full Speed)

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
I_{VDDIO}	1.8V I/O Supply Current	VDDIO = 1.8V		467	760	µA
I_{VDDC}, I_{VDD25}, I_{PVDD}, I_{AVDD}	2.5V Core Supply Current	VDDC, VDD25, PVDD, AVDD = 2.5V		35.6	45	mA
I_{VDD33}	3.3V Supply Current	VDD33 = 3.3V		1.5	2	mA

Table 11-7. 1.8V VDDIO – Active (Full Speed)

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
I_{VDDIO}	1.8V I/O Supply Current	VDDIO = 1.8V		508	810	µA
I_{VDDC}, I_{VDD25}, I_{PVDD}, I_{AVDD}	2.5V Core Supply Current	VDDC, VDD25, PVDD, AVDD = 2.5V		36.3	47	mA
I_{VDD33}	3.3V Supply Current	VDD33 = 3.3V		5.32	6	mA

Table 11-8. 1.8V VDDIO – Suspended (Connected to USB)[1]

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
I_{VDDIO}	1.8V I/O Supply Current	VDDIO = 1.8V		0.07	0.1	µA
I_{VDDC}, I_{VDD25}, I_{PVDD}, I_{AVDD}	2.5V Core Supply Current	VDDC, VDD25, PVDD, AVDD = 2.5V		0.16	31.05	µA
I_{VDD33}	3.3V Supply Current	VDD33 = 3.3V		201.9	202	µA

[1] *To prevent leakage current, 16-bit Data bus (LD[15:0]) should not float when suspended.*

Table 11-9. 1.8V VDDIO – Suspended (Disconnected from USB)[1]

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
I_{VDDIO}	1.8V I/O Supply Current	VDDIO = 1.8V		0.07	0.1	µA
I_{VDDC}, I_{VDD25}, I_{PVDD}, I_{AVDD}	2.5V Core Supply Current	VDDC, VDD25, PVDD, AVDD = 2.5V		0.16	37.26	µA
I_{VDD33}	3.3V Supply Current	VDD33 = 3.3V		0.08	0.1	µA

[1] *To prevent leakage current, 16-bit Data bus (LD[15:0]) should not float when suspended.*

11.3.2 2.5V VDDIO Power Supply Requirements

Operating Conditions – VDDC = 2.37 to 2.63V, VDDIO = 2.37 to 2.63V, VDD33 = 3.13 to 3.47V, T_A = 0 to 85°C

Typical Values – VDDC = 2.5V, VDDIO = 2.5V, VDD33 = 3.3V, T_A = 25°C

Table 11-10. 2.5V VDDIO – Disconnected from USB (NET2272 Still Active)

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
I_{VDDIO}	2.5V I/O Supply Current	VDDIO = 2.5V		678	900	µA
I_{VDDC}, I_{VDD25}, I_{PVDD}, I_{AVDD}	2.5V Core Supply Current	VDDC, VDD25, PVDD, AVDD = 2.5V		37	45	mA
I_{VDD33}	3.3V Supply Current	VDD33 = 3.3V		0.08	0.1	µA

Table 11-11. 2.5V VDDIO – Connected to USB (High Speed)

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
I_{VDDIO}	2.5V I/O Supply Current	VDDIO = 2.5V		678	900	µA
I_{VDDC}, I_{VDD25}, I_{PVDD}, I_{AVDD}	2.5V Core Supply Current	VDDC, VDD25, PVDD, AVDD = 2.5V		52.3	58	mA
I_{VDD33}	3.3V Supply Current	VDD33 = 3.3V		1.7	2	mA

Table 11-12. 2.5V VDDIO – Active (High Speed)

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
I_{VDDIO}	2.5V I/O Supply Current	VDDIO = 2.5V		1.2	2	mA
I_{VDDC}, I_{VDD25}, I_{PVDD}, I_{AVDD}	2.5V Core Supply Current	VDDC, VDD25, PVDD, AVDD = 2.5V		54.3	62	mA
I_{VDD33}	3.3V Supply Current	VDD33 = 3.3V		5.46	6	mA

Table 11-13. 2.5V VDDIO – Connected to USB (Full Speed)

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
I_{VDDIO}	2.5V I/O Supply Current	VDDIO = 2.5V		403	900	µA
I_{VDDC}, I_{VDD25}, I_{PVDD}, I_{AVDD}	2.5V Core Supply Current	VDDC, VDD25, PVDD, AVDD = 2.5V		35.6	45	mA
I_{VDD33}	3.3V Supply Current	VDD33 = 3.3V		1.5	2	mA

Table 11-14. 2.5V VDDIO – Active (Full Speed)

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
I_{VDDIO}	2.5V I/O Supply Current	VDDIO = 2.5V		870	1250	µA
I_{VDDC}, I_{VDD25}, I_{PVDD}, I_{AVDD}	2.5V Core Supply Current	VDDC, VDD25, PVDD, AVDD = 2.5V		36.3	47	mA
I_{VDD33}	3.3V Supply Current	VDD33 = 3.3V		5.6	6	mA

Table 11-15. 2.5V VDDIO – Suspended (Connected to USB)[1]

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
I_{VDDIO}	2.5V I/O Supply Current	VDDIO = 2.5V		0.07	0.1	µA
I_{VDDC}, I_{VDD25}, I_{PVDD}, I_{AVDD}	2.5V Core Supply Current	VDDC, VDD25, PVDD, AVDD = 2.5V		0.13	1	µA
I_{VDD33}	3.3V Supply Current	VDD33 = 3.3V		201.9	202	µA

[1] *To prevent leakage current, 16-bit Data bus (LD[15:0]) should not float when suspended.*

Table 11-16. 2.5V VDDIO – Suspended (Disconnected from USB)[1]

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
I_{VDDIO}	2.5V I/O Supply Current	VDDIO = 2.5V		0.07	0.1	µA
I_{VDDC}, I_{VDD25}, I_{PVDD}, I_{AVDD}	2.5V Core Supply Current	VDDC, VDD25, PVDD, AVDD = 2.5V		0.13	1	µA
I_{VDD33}	3.3V Supply Current	VDD33 = 3.3V		0.08	0.1	µA

[1] *To prevent leakage current, 16-bit Data bus (LD[15:0]) should not float when suspended.*

11.3.3 3.3V VDDIO Power Supply Requirements

Operating Conditions – VDDC = 2.37 to 2.63V, VDDIO = 3.2 to 3.5V, VDD33 = 3.13 to 3.47V, T_A = 0 to 85°C

Typical Values – VDDC = 2.5V, VDDIO = 3.3V, VDD33 = 3.3V, T_A = 25°C

Table 11-17. 3.3V VDDIO – Disconnected from USB (NET2272 Still Active)

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
I_{VDDIO}	3.3V I/O Supply Current	VDDIO = 3.3V		1.16	1.6	mA
I_{VDDC}, I_{VDD25}, I_{PVDD}, I_{AVDD}	2.5V Core Supply Current	VDDC, VDD25, PVDD, AVDD = 2.5V		37	45	mA
I_{VDD33}	3.3V Supply Current	VDD33 = 3.3V		0.08	0.1	µA

Table 11-18. 3.3V VDDIO – Connected to USB (High Speed)

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
I_{VDDIO}	3.3V I/O Supply Current	VDDIO = 3.3V		1.12	2.8	mA
I_{VDDC}, I_{VDD25}, I_{PVDD}, I_{AVDD}	2.5V Core Supply Current	VDDC, VDD25, PVDD, AVDD = 2.5V		52.3	58	mA
I_{VDD33}	3.3V Supply Current	VDD33 = 3.3V		1.7	2	mA

Table 11-19. 3.3V VDDIO – Active (High Speed)

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
I_{VDDIO}	3.3V I/O Supply Current	VDDIO = 3.3V		1.5	4.1	mA
I_{VDDC}, I_{VDD25}, I_{PVDD}, I_{AVDD}	2.5V Core Supply Current	VDDC, VDD25, PVDD, AVDD = 2.5V		53.7	62	mA
I_{VDD33}	3.3V Supply Current	VDD33 = 3.3V		3.75	5.2	mA

Table 11-20. 3.3V VDDIO – Connected to USB (Full Speed)

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
I_{VDDIO}	3.3V I/O Supply Current	VDDIO = 3.3V		1.12	2.8	mA
I_{VDDC}, I_{VDD25}, I_{PVDD}, I_{AVDD}	2.5V Core Supply Current	VDDC, VDD25, PVDD, AVDD = 2.5V		35.6	45	mA
I_{VDD33}	3.3V Supply Current	VDD33 = 3.3V		1.5	2	mA

Table 11-21. 3.3V VDDIO – Active (Full Speed)

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
I_{VDDIO}	3.3V I/O Supply Current	VDDIO = 3.3V		1.24	3.9	mA
I_{VDDC}, I_{VDD25}, I_{PVDD}, I_{AVDD}	2.5V Core Supply Current	VDDC, VDD25, PVDD, AVDD = 2.5V		36.3	47	mA
I_{VDD33}	3.3V Supply Current	VDD33 = 3.3V		4.96	5.8	mA

Table 11-22. 3..3V VDDIO – Suspended (Connected to USB)[1]

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
I_{VDDIO}	3.3V I/O Supply Current	VDDIO = 3.3V		1.48	1.6	µA
I_{VDDC}, I_{VDD25}, I_{PVDD}, I_{AVDD}	2.5V Core Supply Current	VDDC, VDD25, PVDD, AVDD = 2.5V		0.13	1	µA
I_{VDD33}	3.3V Supply Current	VDD33 = 3.3V		201.9	202	µA

[1] *To prevent leakage current, 16-bit Data bus (LD[15:0]) should not float when suspended.*

Table 11-23. 3.3V VDDIO – Suspended (Disconnected from USB)[1]

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
I_{VDDIO}	3.3V I/O Supply Current	VDDIO = 3.3V		1.48	1.6	µA
I_{VDDC}, I_{VDD25}, I_{PVDD}, I_{AVDD}	2.5V Core Supply Current	VDDC, VDD25, PVDD, AVDD = 2.5V		0.13	1	µA
I_{VDD33}	3.3V Supply Current	VDD33 = 3.3V		0.08	0.1	µA

[1] *To prevent leakage current, 16-bit Data bus (LD[15:0]) should not float when suspended.*

11.3.4 USB Full- and High-Speed DC Specifications

Operating Conditions – VDDC = 2.2 to 2.6V, VDDIO = 1.7 to 3.5V, T_A = 0 to 85°C

Typical Values – VDDC = 2.5V, VDDIO = 3.3V, T_A = 25°C

Table 11-24. USB Full-Speed DC Specifications

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{IH}	Input High Level (Driven)	Note 4	2.0	–	–	V
V_{IHZ}	Input High Level (Floating)	Note 4	2.7	–	3.6	V
V_{IL}	Input Low Level	Note 4	–	–	0.8	V
V_{DI}	Differential Input Sensitivity	| (D+) - (D-) |	0.2	–	–	V
V_{CM}	Differential Common Mode Range	Includes V_{DI} range	0.8	–	2.5	V
V_{OL}	Output Low Level	Notes 4, 5	0.0	–	0.3	V
V_{OH}	Output High Level (Driven)	Notes 4, 6	2.8	–	3.6	V
V_{SE1}	Single-Ended One		0.8	–	–	V
V_{CRS}	Output Signal Crossover Voltage	Note 10	1.3	–	2.0	V
C_{IO}	I/O Capacitance	Pin/Ball to Ground	–	–	20	pF

Table 11-25. USB High-Speed DC Specifications

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{HSSQ}	High-Speed Squelch Detection Threshold (Differential Signal Amplitude)		100	–	150	mV
V_{HSDSC}	High-Speed Disconnect Detection Threshold (Differential Signal Amplitude)		525	–	625	mV
–	High-Speed Differential Input Signaling Levels	Specified by eye patterns	–	–	–	–
V_{HSCM}	High-Speed Data Signaling Common Mode Voltage Range		-50	–	+500	mV
V_{HSOI}	High-Speed Idle Level		-10	–	+10	mV
V_{HSOH}	High-Speed Data Signaling High		360	–	440	mV
V_{HSOL}	High-Speed Data Signaling Low		-10	–	+10	mV
V_{CHIRPJ}	Chirp J Level (Differential Voltage)		700	–	1100	mV
V_{CHIRPK}	Chirp K Level (Differential Voltage)		-900	–	-500	mV
C_{IO}	I/O Capacitance	Pin/Ball to Ground	–	–	20	pF

11.3.5 Local Bus DC Specifications

Operating Conditions – VDDC = 2.2 to 2.6V, VDDIO = 1.7 to 3.5V, T_A = 0 to 85°C

Typical Values – VDDC = 2.5V, VDDIO = 3.3V, T_A = 25°C

Table 11-26. Local Bus DC Specifications

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{IH}	5.0V Tolerant Input High Voltage		2.0	–	5.5	V
V_{IL}	5.0V Tolerant Input Low Voltage		0	–	0	V
I_{IL}	Input Leakage	0V < V_{IN} < 5.5V	-10	–	+10	µA
I_{OZ}	Hi-Z State Data Line Leakage	0V < V_{IN} < 5.5V	–	–	10	µA
V_{OH}	5.0V Tolerant Input High Voltage	I_{OUT} = -12 mA	2.4	–	–	V
V_{OL}	5.0V Tolerant Input Low Voltage	I_{OUT} = +12 mA	–	–	0.4	V
C_{IN}	Input Capacitance	Pin/Ball to Ground	–	–	10	pF
C_{CLK}	CLK Pin/Ball Capacitance	Pin/Ball to Ground	5	–	12	pF
C_{IDSEL}	IDSEL Pin/Ball Capacitance	Pin/Ball to Ground	–	–	8	pF

11.4 AC Specifications

11.4.1 USB Full- and High-Speed Port AC Specifications

Operating Conditions – VDDC = 2.2 to 2.6V, VDDIO = 1.7 to 3.5V, T_A = 0 to 85°C

Typical values – VDDC = 2.5V, VDDIO = 3.3V, T_A = 25°C

Table 11-27. USB Full-Speed Port AC Specifications

Symbol	Parameter	Conditions	Waveform	Min	Typ	Max	Unit
T_{FR}	Full-Speed Rise Time	C_L = 50 pF, Note 10	Figure 11-2	4	–	20	ns
T_{FF}	Full-Speed Fall Time		Figure 11-1	4	–	20	ns
T_{FRFM}	Rise and Fall Time Matching	(T_{FR}/T_{FF}), Note 10	Figure 11-1 Figure 11-2	90	–	110	%
Z_{DRV}	Driver Output Resistance	Steady State Drive	–	10	–	15	Ohms
$T_{FDRATHS}$	Full-Speed Data Rate		–	11.994	12	12.006	Mbps
T_{DJ1}	Source Differential Driver Jitter to Next Transition	Notes 7, 8, 10, 12	Figure 11-3	-2	0	2	ns
T_{DJ2}	Source Differential Driver Jitter for Paired Transitions	Notes 7, 8, 10, 12	Figure 11-3	-1	0	+1	ns
T_{FDEOP}	Source Jitter for Differential Transition to SE0 Transition	Notes 8, 11	Figure 11-4	-2	0	+5	ns
T_{JR1}	Receiver Data Jitter Tolerance to Next Transition	Note 8	Figure 11-5	-18.5	0	+18.5	ns
T_{JR2}	Receiver Data Jitter Tolerance for Paired Transitions	Note 8	Figure 11-5	-9	0	+9	ns
T_{FEOPT}	Source SE0 Interval of EOP		Figure 11-4	160	167	175	ns
T_{FEOPR}	Receiver SE0 Interval of EOP	Note 13	Figure 11-4	82	–	–	ns
T_{FST}	Width of SE0 Interval during Differential Transition		–	14	–	–	ns

Table 11-28. USB High-Speed Port AC Specifications

Symbol	Parameter	Conditions	Waveform	Min	Typ	Max	Unit
T_{HSR}	High-Speed Rise Time	Note 16	–	500	–	–	ps
T_{HSF}	High-Speed Fall Time		–	500	–	–	ps
Z_{DRV}	Driver Output Resistance	Steady State Drive	–	10	–	15	Ohms
T_{HSDRV}	High-Speed Data Rate		–	479.760	480	480.240	Mbps
–	Data Source Jitter	Note 17	–	–	–	–	–
–	Receiver Jitter Tolerance	Note 17	–	–	–	–	–

11.4.2 USB Full-Speed Port AC Waveforms

Figure 11-1. Data Signal Rise and Fall Time

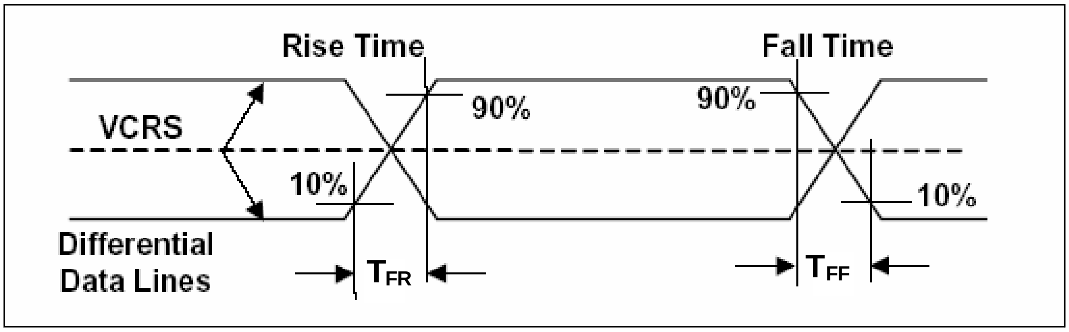


Figure 11-2. Full-Speed Load

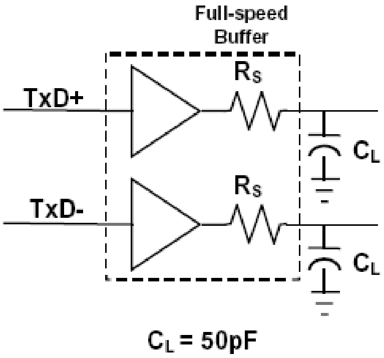


Figure 11-3. Source Differential Driver Jitter

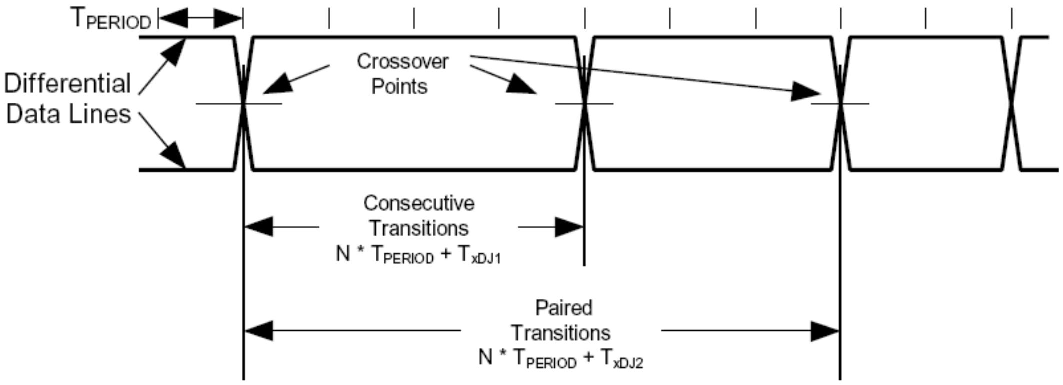


Figure 11-4. Differential to EOP Transition Skew and EOP Width

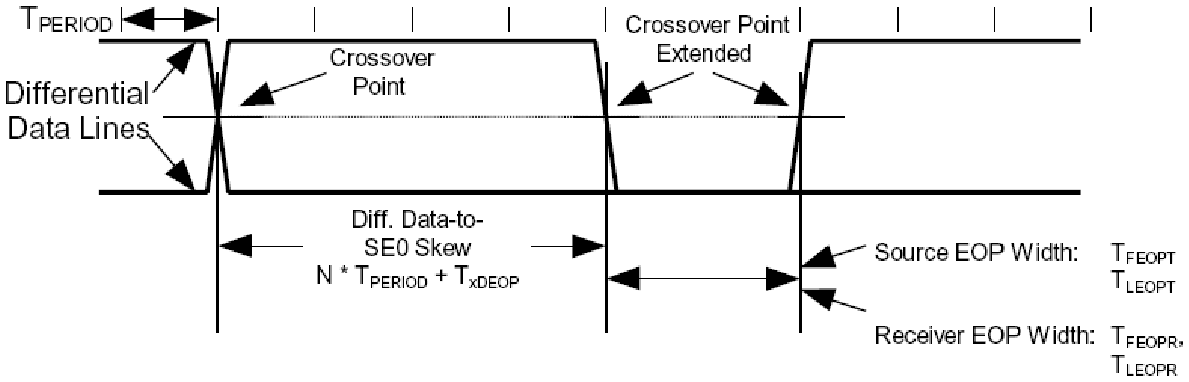
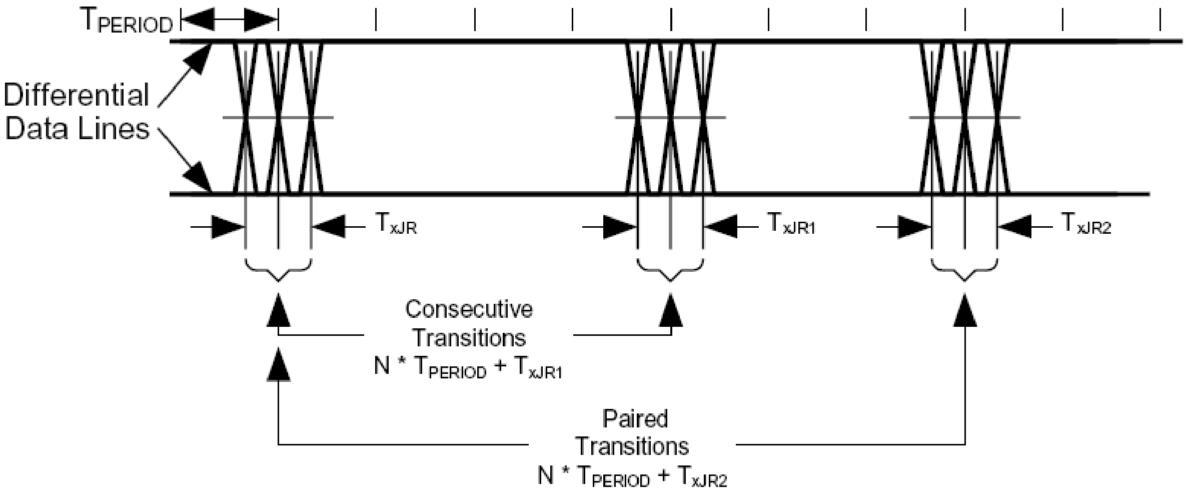


Figure 11-5. Receiver Jitter Tolerance



11.4.3 USB Port AC/DC Specification Notes

1. Measured at A plug.
2. Measured at A receptacle.
3. Measured at B receptacle.
4. Measured at A or B connector.
5. Measured with RL of 1.425K Ohms to 3.6V.
6. Measured with RL of 14.25K Ohms to Ground.
7. Timing difference between the differential Data signals.
8. Measured at crossover point of differential Data signals.
9. The maximum load specification is the maximum effective capacitive load allowed that meets the Target hub VBUS drop of 330 mV.
10. Excluding the first transition from the Idle state.
11. The two transitions should be a (nominal) bit time apart.
12. For both transitions of differential signaling.
13. Must accept as valid EOP.
14. Single-ended capacitance of D+ or D- is the capacitance of D+/D- to all other conductors and, if present, shield in the cable. *That is*, to measure the single-ended capacitance of D+, short D-, VBUS, Ground, and the shield line together and measure the capacitance of D+ to the other conductors.
15. For high-power devices (non-hubs) when enabled for Device-Remote Wakeup.
16. Measured from 10 to 90 percent of the Data signal.
17. Source and receiver jitter specified by the eye pattern templates in the *USB r2.0*, Section 7.1.2.2.

11.4.4 1.8V VDDIO Local Bus AC Specifications

11.4.4.1 1.8V VDDIO Local Bus Non-Multiplexed Write

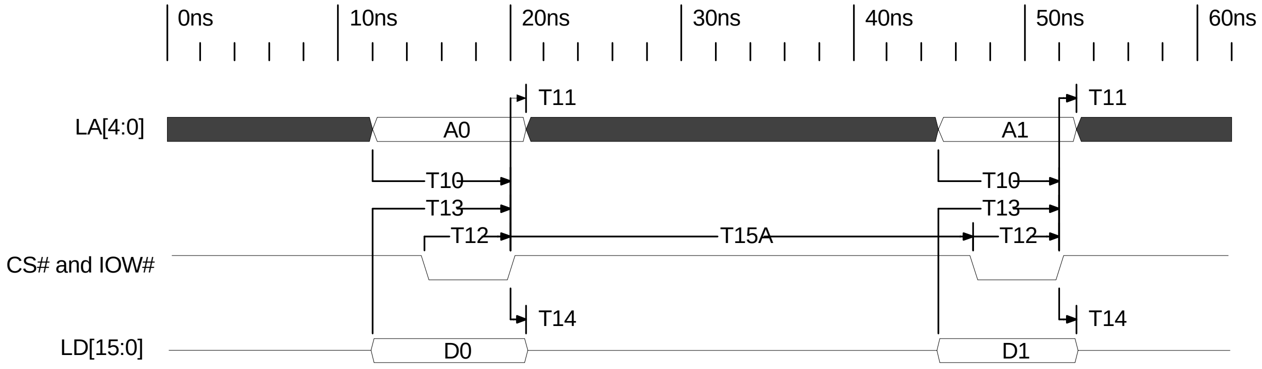
Operating Conditions – VDDC = 2.2 to 2.6V, VDDIO = 1.7 to 1.9V, T_A = 0 to 85°C, Output Load = 25 pF

Table 11-29. 1.8V VDDIO Local Bus Non-Multiplexed Write Specifications

Symbol	Parameter	Min	Max	Unit	Notes
T10	Address Setup to End of Write Enable	5		ns	1
T11	Address Hold from End of Write Enable	0		ns	1
T12	Write Enable Width	5		ns	1
T13	Data Setup to End of Write Enable	5		ns	1
T14	Data Hold Time from End of Write Enable	0		ns	1
T15A	Recovery Time to Next Write	28		ns	
T15B	Recovery Time to Next Read	52		ns	2

Notes:

1. *Write Enable is the occurrence of both CS# and IOW#.*
2. *Because reading and writing to **EP_DATA** cause **EP_AVAILx** and **EP_TRANSFERx** to change values, it is necessary to increase the recovery time to 70 ns between a read or write to **EP_DATA** and a read from **EP_AVAILx** or **EP_TRANSFERx**.*



11.4.4.2 1.8V VDDIO Local Bus Multiplexed Write

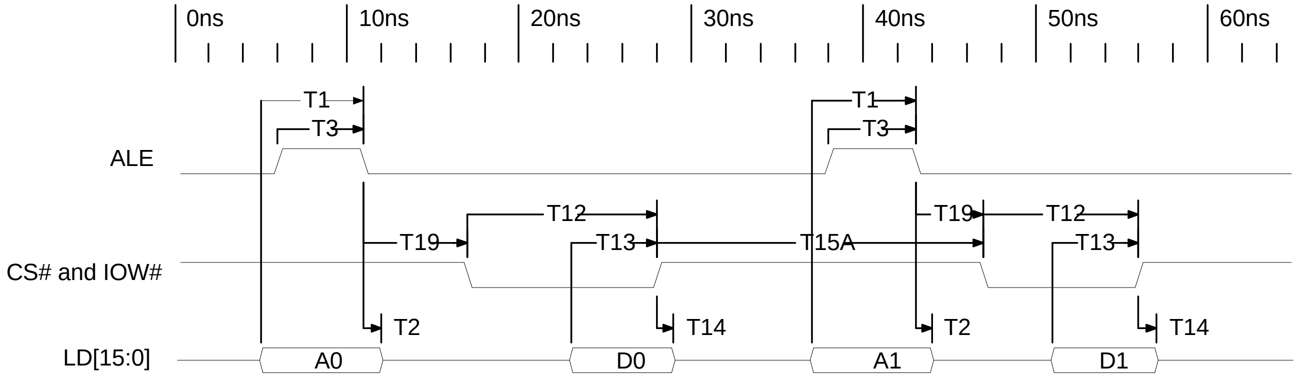
Operating Conditions – VDDC = 2.2 to 2.6V, VDDIO = 1.7 to 1.9V, T_A = 0 to 85°C, Output Load = 25 pF

Table 11-30. 1.8V VDDIO Local Bus Multiplexed Write Specifications

Symbol	Parameter	Min	Max	Unit	Notes
T1	Address Setup to Falling Edge of ALE	5		ns	
T2	Address Hold from Falling Edge of ALE	1		ns	
T3	ALE Width	5		ns	
T12	Write Enable Width	5		ns	1
T13	Data Setup to End of Write Enable	5		ns	1
T14	Data Hold Time from End of Write Enable	0		ns	1
T15A	Recovery Time to Next Write	28		ns	
T15B	Recovery Time to Next Read	52		ns	2
T19	ALE Falling Edge to Write Enable	1		ns	1

Notes:

1. *Write Enable is the occurrence of both CS# and IOW#.*
2. *Because reading and writing to **EP_DATA** cause **EP_AVAILx** and **EP_TRANSFERx** to change values, it is necessary to increase the recovery time to 70 ns between a read or write to **EP_DATA** and a read from **EP_AVAILx** or **EP_TRANSFERx**.*



11.4.4.3 1.8V VDDIO Local Bus Non-Multiplexed Read

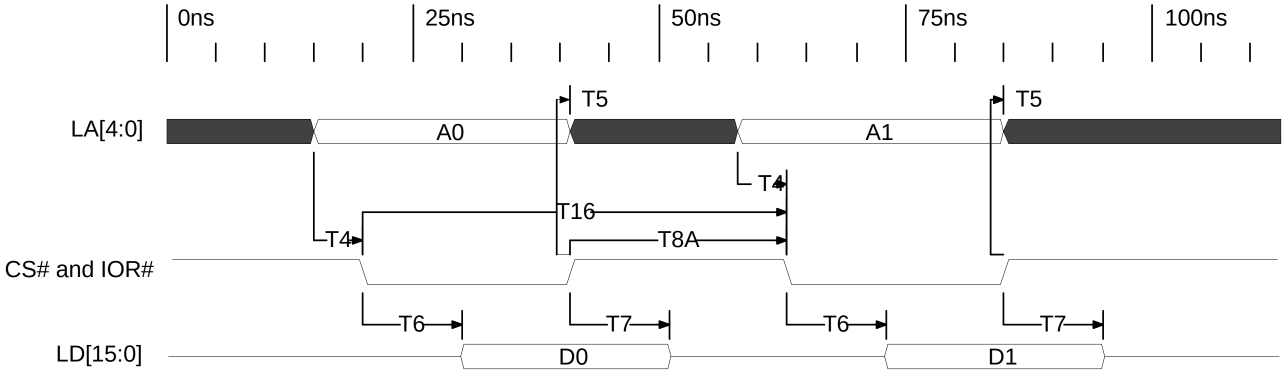
Operating Conditions – VDDC = 2.2 to 2.6V, VDDIO = 1.7 to 1.9V, T_A = 0 to 85°C, Output Load = 25 pF

Table 11-31. 1.8V VDDIO Local Bus Non-Multiplexed Read Specifications

Symbol	Parameter	Min	Max	Unit	Notes
T4	Address Setup to Read Enable	-1		ns	1
T5	Address Hold from End of Read Enable	-2		ns	1
T6	Data Access Time from LA Valid or Read Enable Asserted, Whichever Is Later		30	ns	1
T7	Data Three-State Time from End of Read Enable	3	16	ns	1
T8A	Recovery Time to Next Read	24		ns	2
T8B	Recovery Time to Next Write	24		ns	
T16	Read Cycle Time	54		ns	

Notes:

1. *Read Enable is the occurrence of both CS# and IOR#.*
2. *Because reading and writing to **EP_DATA** cause **EP_AVAILx** and **EP_TRANSFERx** to change values, it is necessary to increase the recovery time to 37 ns between a read or write to **EP_DATA** and a read from **EP_AVAILx** or **EP_TRANSFERx**.*



11.4.4.4 1.8V VDDIO Local Bus Multiplexed Read

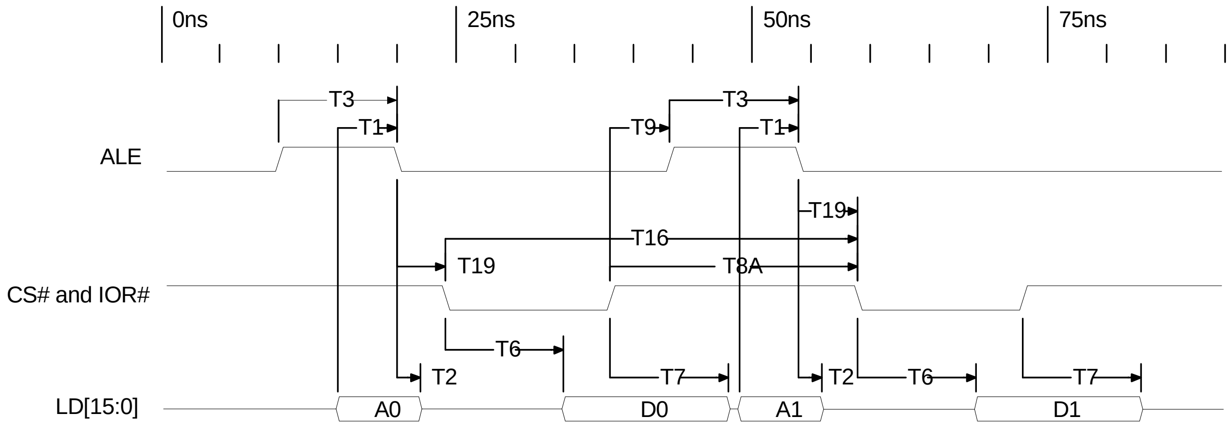
Operating Conditions – VDDC = 2.2 to 2.6V, VDDIO = 1.7 to 1.9V, T_A = 0 to 85°C, Output Load = 25 pF

Table 11-32. 1.8V VDDIO Local Bus Multiplexed Read Specifications

Symbol	Parameter	Min	Max	Unit	Notes
T1	Address Setup to Falling Edge of ALE	5		ns	
T2	Address Hold from Falling Edge of ALE	1		ns	
T3	ALE Width	5		ns	
T6	Data Access Time from Read Enable		30	ns	1
T7	Data Three-State Time from End of Read Enable	3	15	ns	1
T8A	Recovery Time to Next Read	24		ns	2
T8B	Recovery Time to Next Write	24		ns	
T9	Recovery Time to next ALE	5		ns	
T16	Read Cycle Time	54		ns	
T19	ALE Falling Edge to Read Enable	1		ns	1

Notes:

1. *Read Enable is the occurrence of both CS# and IOR#.*
2. *Because reading and writing to* ***EP_DATA*** *cause* ***EP_AVAILx*** *and* ***EP_TRANSFERx*** *to change values, it is necessary to increase the recovery time to 37 ns between a read or write to* ***EP_DATA*** *and a read from* ***EP_AVAILx*** *or* ***EP_TRANSFERx****.*



11.4.4.5 1.8V VDDIO Local Bus DMA Write – Slow Mode

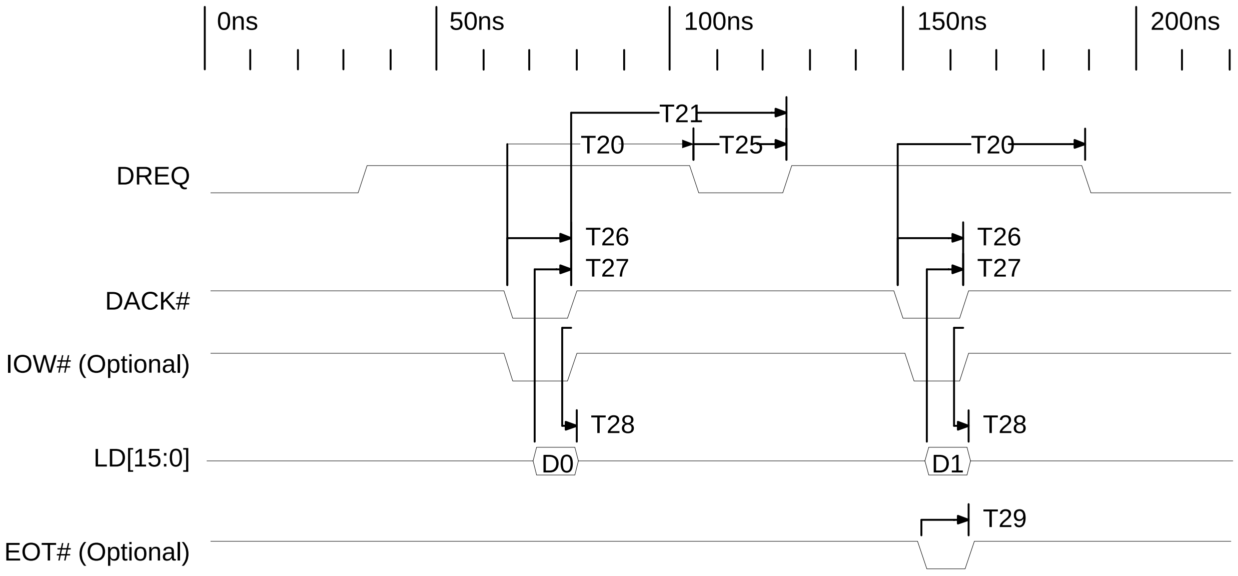
Operating Conditions – VDDC = 2.2 to 2.6V, VDDIO = 1.7 to 1.9V, T_A = 0 to 85°C, Output Load = 25 pF

Table 11-33. 1.8V VDDIO Local Bus DMA Write – Slow Mode Specifications

Symbol	Parameter	Min	Typ	Max	Unit	Notes
T20	Write Enable True to DREQ False	44	59	70	ns	2
T21	Write Enable False to DREQ True	26	43	54	ns	
T25	DREQ False to DREQ True	14	34		ns	
T26	Write Enable Width	40			ns	1
T27	Data Setup to End of Write Enable	10			ns	1
T28	Data Hold Time from End of Write Enable	0			ns	1
T29	Width of EOT# Pulse	40			ns	3

Notes:

1. *For Non-DMA Split mode, Write Enable is the occurrence of DACK# and optionally, IOW#. For DMA Split mode, Write Enable is the occurrence of DACK# and optionally, DMAWR#.*
2. *The minimum value is guaranteed only if the **DMAREQ** register DMA Request Enable bit is set.*
3. *EOT#, DACK#, and optionally, IOW# or DMAWR#, must all be true for at least T29 for proper recognition of the EOT# pulse.*



11.4.4.6 1.8V VDDIO Local Bus DMA Write – Fast Mode

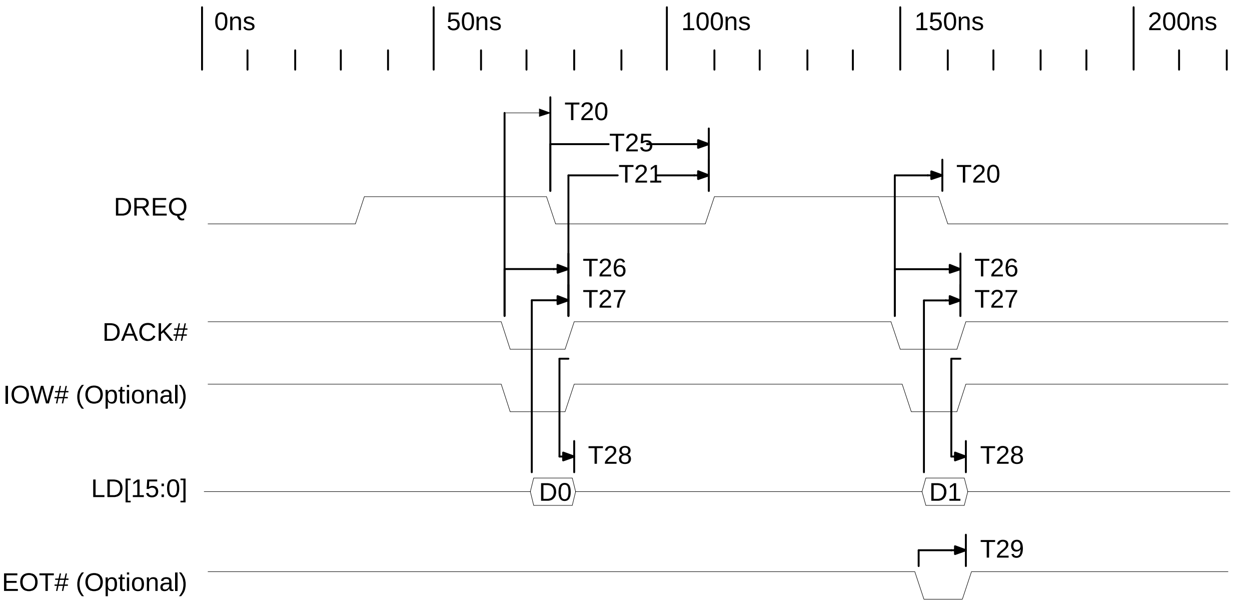
Operating Conditions – VDDC = 2.2 to 2.6V, VDDIO = 1.7 to 1.9V, T_A = 0 to 85°C, Output Load = 25 pF

Table 11-34. 1.8V VDDIO Local Bus DMA Write – Fast Mode Specifications

Symbol	Parameter	Min	Typ	Max	Unit	Notes
T20	Write Enable True to DREQ False	8		33	ns	2
T21	Write Enable False to DREQ True	25	39	55	ns	
T25	DREQ False to DREQ True	52	60		ns	
T26	Write Enable Width	40			ns	1
T27	Data Setup to End of Write Enable	10			ns	1
T28	Data Hold Time from End of Write Enable	0			ns	1
T29	Width of EOT# Pulse	40			ns	3

Notes:

1. *For Non-DMA Split mode, Write Enable is the occurrence of DACK# and optionally, IOW#. For DMA Split mode, Write Enable is the occurrence of DACK# and optionally, DMAWR#.*
2. *The minimum value is guaranteed only if the **DMAREQ** register DMA Request Enable bit is set.*
3. *EOT#, DACK#, and optionally, IOW# or DMAWR#, must all be true for at least T29 for proper recognition of the EOT# pulse.*



11.4.4.7 1.8V VDDIO Local Bus DMA Write – Burst Mode

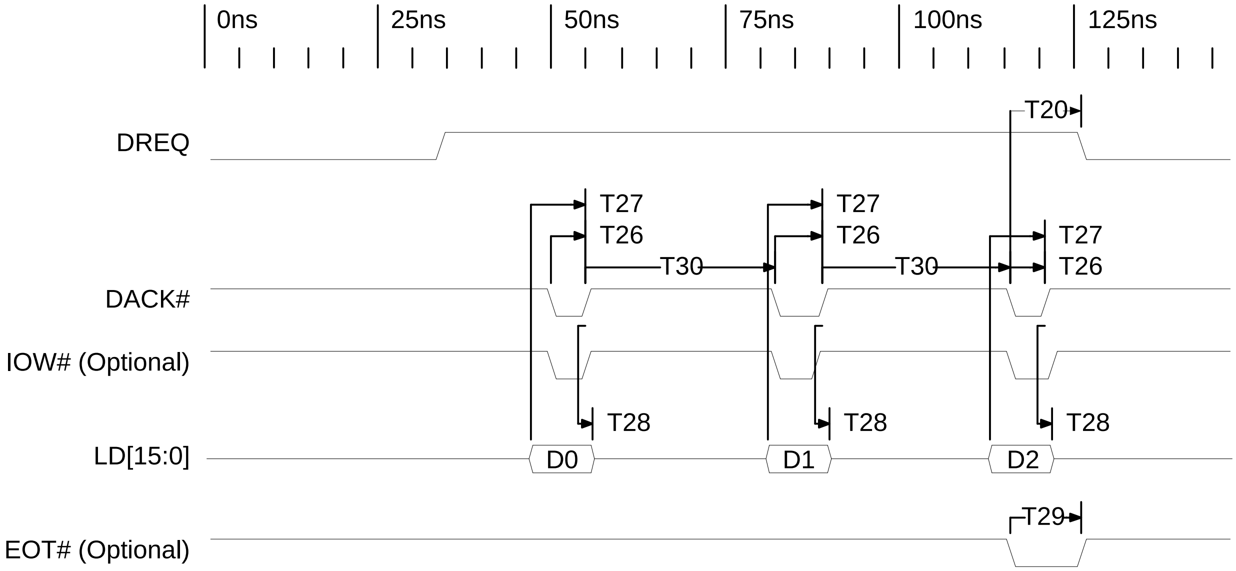
Operating Conditions – VDDC = 2.2 to 2.6V, VDDIO = 1.7 to 1.9V, T_A = 0 to 85°C, Output Load = 25 pF

Table 11-35. 1.8V VDDIO Local Bus DMA Write – Burst Mode Specifications

Symbol	Parameter	Min	Typ	Max	Unit	Notes
T20	Write Enable True to DREQ False	8		33	ns	
T26	Write Enable Width	38			ns	1
T27	Data Setup to End of Write Enable	10			ns	1
T28	Data Hold Time from End of Write Enable	0			ns	1
T29	Width of EOT# Pulse	40			ns	3
T30	DMA Write Recovery	28			ns	

Notes:

1. *For Non-DMA Split mode, Write Enable is the occurrence of DACK# and optionally, IOW#. For DMA Split mode, Write Enable is the occurrence of DACK# and optionally, DMAWR#.*
3. *EOT#, DACK#, and optionally, IOW# or DMAWR#, must all be true for at least T29 for proper recognition of the EOT# pulse.*



11.4.4.8 1.8V VDDIO Local Bus DMA Read – Slow Mode

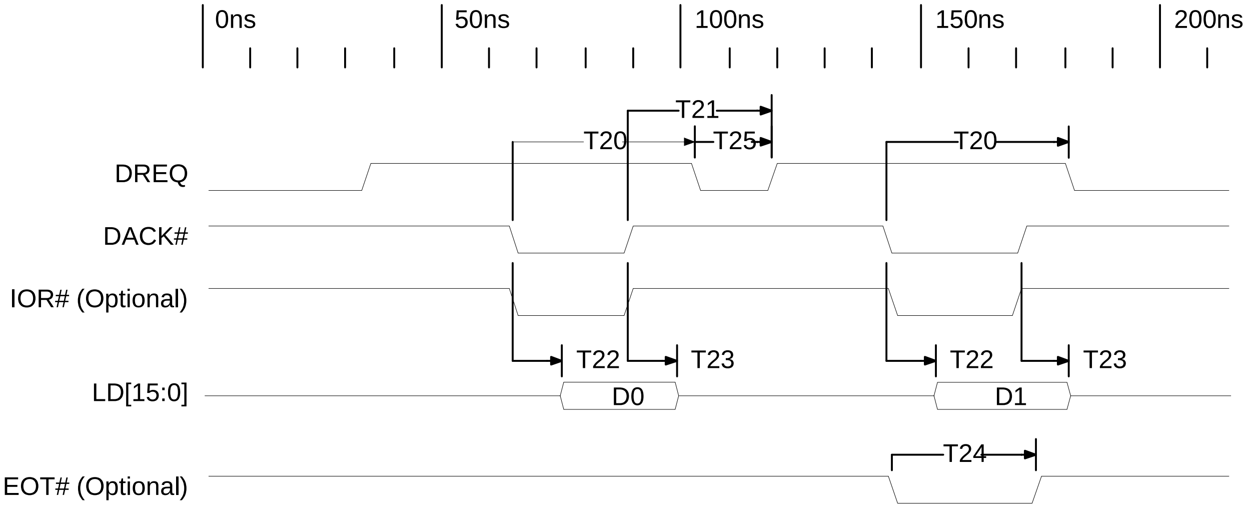
Operating Conditions – VDDC = 2.2 to 2.6V, VDDIO = 1.7 to 1.9V, T_A = 0 to 85°C, Output Load = 25 pF

Table 11-36. 1.8V VDDIO Local Bus DMA Read – Slow Mode Specifications

Symbol	Parameter	Min	Typ	Max	Unit	Notes
T20	Read Enable True to DREQ False	28	47	67	ns	2
T21	Read Enable False to DREQ True	40	55	66	ns	
T22	Data Access Time from Read Enable			25	ns	1
T23	Data Three-State Time from End of Read Enable	2		16	ns	1
T24	Width of EOT# Pulse	40			ns	3
T25	DREQ False to DREQ True	23	33		ns	

Notes:

1. *For Non-DMA Split mode, Read Enable is the occurrence of DACK# and optionally, IOR#. For DMA Split mode, Read Enable is the occurrence of DACK# and optionally, DMARD#.*
2. *The minimum value is guaranteed only if the **DMAREQ** register DMA Request Enable bit is set.*
3. *EOT#, DACK#, and optionally, IOR# or DMARD#, must all be true for at least T24 for proper recognition of the EOT# pulse.*
4. *A recovery time of 2 ns is required between DMA Read Enable de-assertion and I/O Read Enable assertion. (This note is a general requirement not shown on the timing diagrams.)*



11.4.4.9 1.8V VDDIO Local Bus DMA Read – Fast Mode

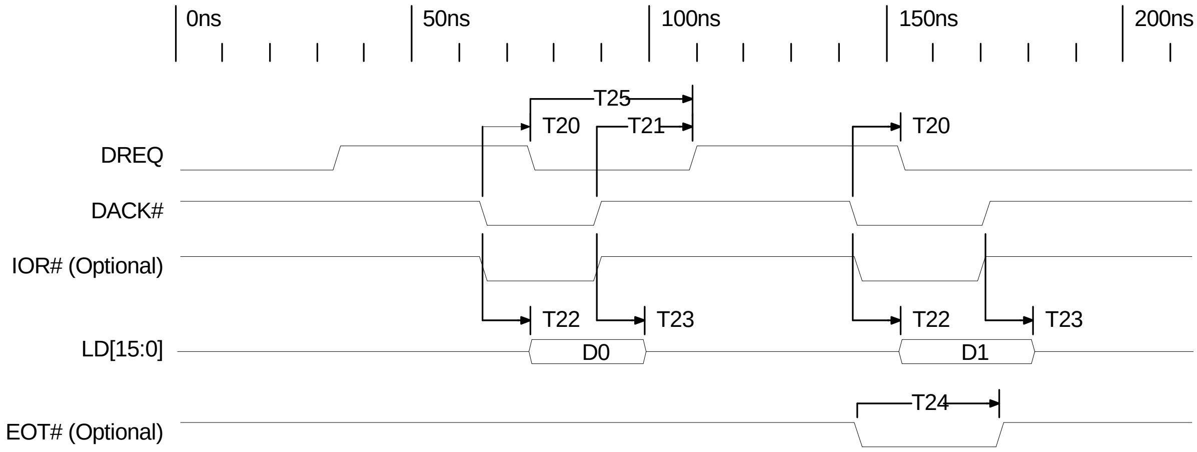
Operating Conditions – VDDC = 2.2 to 2.6V, VDDIO = 1.7 to 1.9V, T_A = 0 to 85°C, Output Load = 25 pF

Table 11-37. 1.8V VDDIO Local Bus DMA Read – Fast Mode Specifications

Symbol	Parameter	Min	Typ	Max	Unit	Notes
T20	Read Enable True to DREQ False	8		30	ns	2
T21	Read Enable False to DREQ True	7	15	25	ns	
T22	Data Access Time from Read Enable			25	ns	1
T23	Data Three-State Time from End of Read Enable	3		16	ns	1
T24	Width of EOT# Pulse	40			ns	3
T25	DREQ False to DREQ True	33	36		ns	

Notes:

1. *For Non-DMA Split mode, Read Enable is the occurrence of DACK# and optionally, IOR#. For DMA Split mode, Read Enable is the occurrence of DACK# and optionally, DMARD#.*
2. *The minimum value is only guaranteed if the* ***DMAREQ*** *register DMA Request Enable bit is set.*
3. *EOT#, DACK#, and optionally, IOR# or DMARD#, must all be true for at least T24 for proper recognition of the EOT# pulse.*
4. *A recovery time of 2 ns is required between DMA Read Enable de-assertion and I/O Read Enable assertion. (This note is a general requirement not shown on the timing diagrams.)*



11.4.4.10 1.8V VDDIO Local Bus DMA Read – Burst Mode

Operating Conditions – VDDC = 2.2 to 2.6V, VDDIO = 1.7 to 1.9V, T_A = 0 to 85°C, Output Load = 25 pF

Table 11-38. 1.8V VDDIO Local Bus DMA Read – Burst Mode Specifications

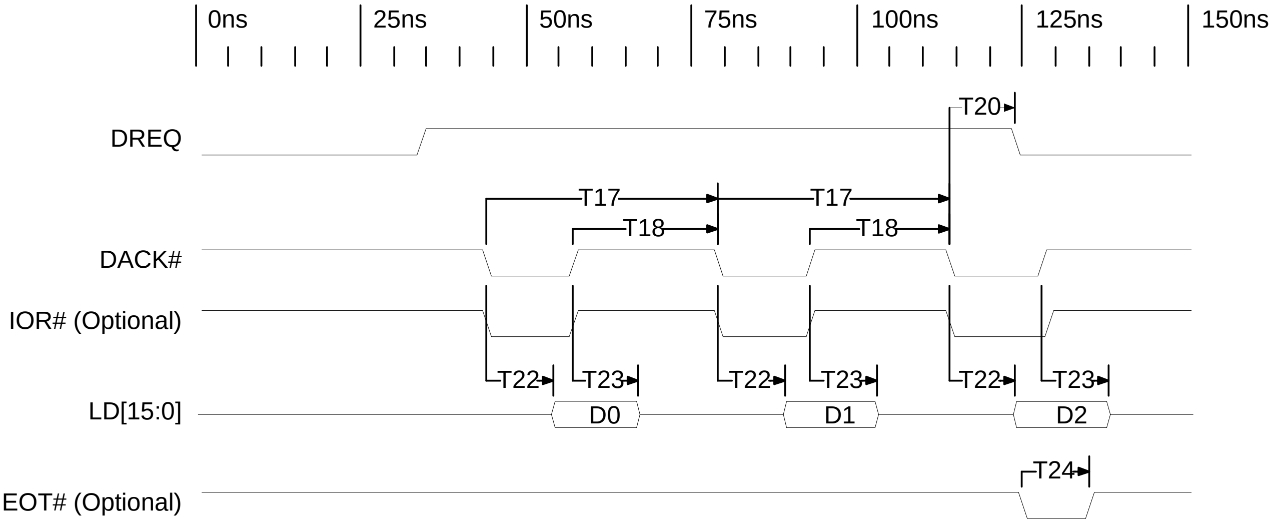
Symbol	Parameter	Min	Typ	Max	Unit	Notes
T17	DMA Read Cycle Time	44			ns	
T18	DMA Read Recovery Time	19			ns	
T20	Read Enable True to DREQ False	8		31	ns	
T22	Data Access Time from Read Enable			25	ns	1
T23	Data Three-State Time from End of Read Enable	5		16	ns	1
T24	Width of EOT# Pulse	40			ns	3

Notes:

1. *For Non-DMA Split mode, Read Enable is the occurrence of DACK# and optionally, IOR#. For DMA Split mode, Read Enable is the occurrence of DACK# and optionally, DMARD#.*

3. *EOT#, DACK#, and optionally, IOR# or DMARD#, must all be true for at least T24 for proper recognition of the EOT# pulse.*

4. *A recovery time of 2 ns is required between DMA Read Enable de-assertion and I/O Read Enable assertion. (This note is a general requirement not shown on the timing diagrams.)*



11.4.5 2.5V VDDIO Local Bus AC Specifications

11.4.5.1 2.5V VDDIO Local Bus Non-Multiplexed Write

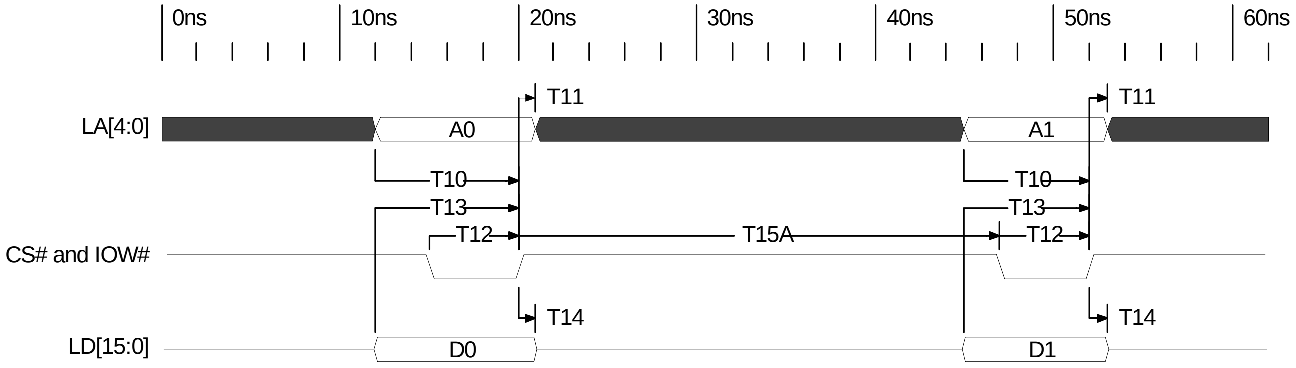
Operating Conditions – VDDC and VDDIO = 2.2 to 2.6V, T_A = 0 to 85°C, Output Load = 25 pF

Table 11-39. 2.5V VDDIO Local Bus Non-Multiplexed Write Specifications

Symbol	Parameter	Min	Max	Unit	Notes
T10	Address Setup to End of Write Enable	5	–	ns	1
T11	Address Hold from End of Write Enable	0	–	ns	1
T12	Write Enable Width	5	–	ns	1
T13	Data Setup to End of Write Enable	5	–	ns	1
T14	Data Hold Time from End of Write Enable	0	–	ns	1
T15A	Recovery Time to Next Write	28	–	ns	
T15B	Recovery Time to Next Read	52	–	ns	2

Notes:

1. *Write Enable is the occurrence of both CS# and IOW#.*
2. *Because reading and writing to **EP_DATA** cause **EP_AVAILx** and **EP_TRANSFERx** to change values, it is necessary to increase the recovery time to 70 ns between a read or write to **EP_DATA** and a read from **EP_AVAILx** or **EP_TRANSFERx**.*



11.4.5.2 2.5V VDDIO Local Bus Multiplexed Write

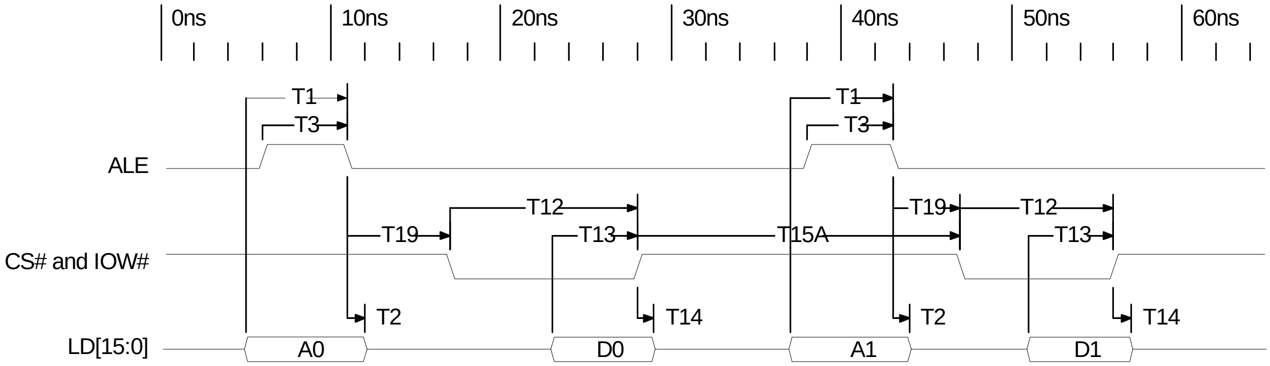
Operating Conditions – VDDC and VDDIO = 2.2 to 2.6V, T_A = 0 to 85°C, Output Load = 25 pF

Table 11-40. 2.5V VDDIO Local Bus Multiplexed Write Specifications

Symbol	Parameter	Min	Max	Unit	Notes
T1	Address Setup to Falling Edge of ALE	5	–	ns	
T2	Address Hold from Falling Edge of ALE	1	–	ns	
T3	ALE Width	5	–	ns	
T12	Write Enable Width	5	–	ns	1
T13	Data Setup to End of Write Enable	5	–	ns	1
T14	Data Hold Time from End of Write Enable	0	–	ns	1
T15A	Recovery Time to Next Write	28	–	ns	
T15B	Recovery Time to Next Read	52	–	ns	2
T19	ALE Falling Edge to Write Enable	1	–	ns	1

Notes:

1. *Write Enable is the occurrence of both CS# and IOW#.*
2. *Because reading and writing to **EP_DATA** cause **EP_AVAILx** and **EP_TRANSFERx** to change values, it is necessary to increase the recovery time to 70 ns between a read or write to **EP_DATA** and a read from **EP_AVAILx** or **EP_TRANSFERx**.*



11.4.5.3 2.5V VDDIO Local Bus Non-Multiplexed Read

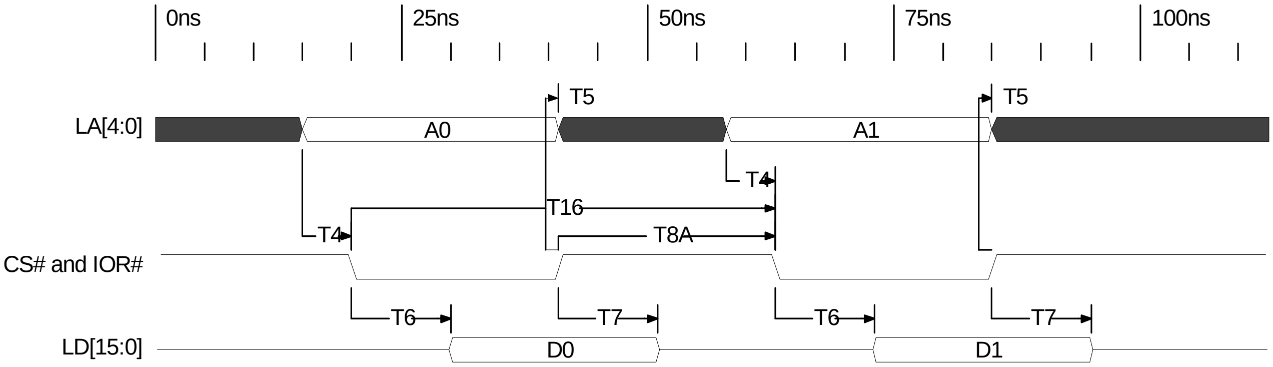
Operating Conditions – VDDC and VDDIO = 2.2 to 2.6V, T_A = 0 to 85°C, Output Load = 25 pF

Table 11-41. 2.5V VDDIO Local Bus Non-Multiplexed Read Specifications

Symbol	Parameter	Min	Max	Unit	Notes
T4	Address Setup to Read Enable	-1	–	ns	1
T5	Address Hold from End of Read Enable	-2	–	ns	1
T6	Data Access Time from LA valid or Read Enable Asserted (Whichever Occurs Later)	–	23	ns	1
T7	Data Three-State Time from End of Read Enable	2	11	ns	1
T8A	Recovery Time to Next Read	19	–	ns	2
T8B	Recovery Time to Next Write	19	–	ns	
T16	Read Cycle Time	42	–	ns	

Notes:

1. *Read Enable is the occurrence of both CS# and IOR#.*
2. *Because reading and writing to **EP_DATA** cause **EP_AVAILx** and **EP_TRANSFERx** to change values, it is necessary to increase the recovery time to 37 ns between a read or write to **EP_DATA** and a read from **EP_AVAILx** or **EP_TRANSFERx**.*



11.4.5.4 2.5V VDDIO Local Bus Multiplexed Read

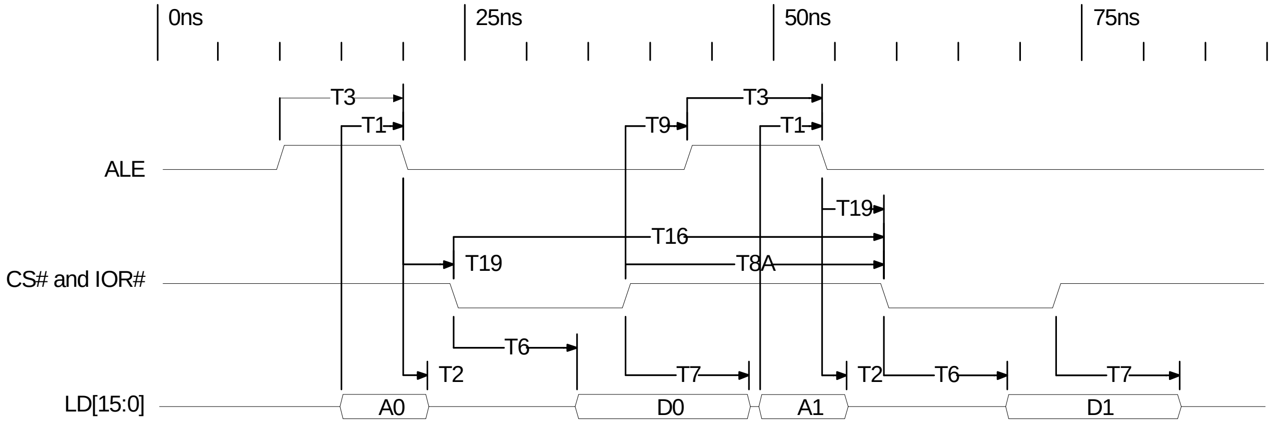
Operating Conditions – VDDC and VDDIO = 2.2 to 2.6V, T_A = 0 to 85°C, Output Load = 25 pF

Table 11-42. 2.5V VDDIO Local Bus Multiplexed Read Specifications

Symbol	Parameter	Min	Max	Unit	Notes
T1	Address Setup to Falling Edge of ALE	5	–	ns	
T2	Address Hold from Falling Edge of ALE	1	–	ns	
T3	ALE width	5	–	ns	
T6	Data Access Time from Read Enable	–	23	ns	1
T7	Data Three-State Time from End of Read Enable	2	11	ns	1
T8A	Recovery Time to Next Read	19	–	ns	2
T8B	Recovery Time to Next Write	19	–	ns	
T9	Recovery Time to Next ALE	5	–	ns	
T16	Read Cycle Time	42	–	ns	
T19	ALE Falling Edge to Read Enable	1	–	ns	1

Notes:

1. *Read Enable is the occurrence of both CS# and IOR#.*
2. *Because reading and writing to **EP_DATA** cause **EP_AVAILx** and **EP_TRANSFERx** to change values, it is necessary to increase the recovery time to 37 ns between a read or write to **EP_DATA** and a read from **EP_AVAILx** or **EP_TRANSFERx**.*



11.4.5.5 2.5V VDDIO Local Bus DMA Write – Slow Mode

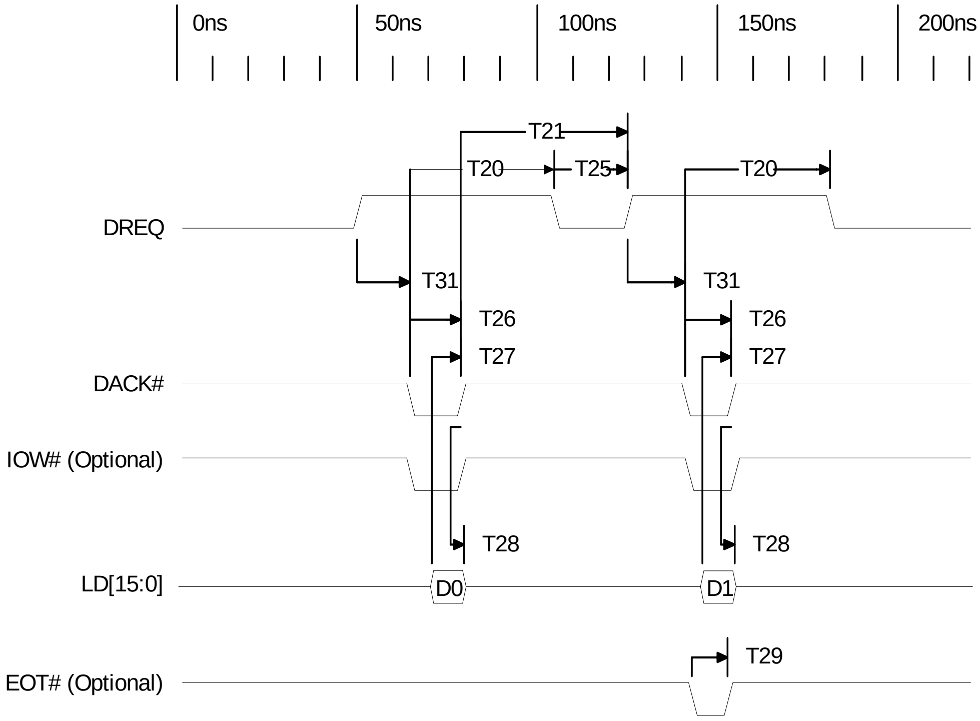
Operating Conditions – VDDC and VDDIO = 2.2 to 2.6V, T_A = 0 to 85°C, Output Load = 25 pF

Table 11-43. 2.5V VDDIO Local Bus DMA Write – Slow Mode Specifications

Symbol	Parameter	Min	Typ	Max	Unit	Notes
T20	Write Enable True to DREQ False	17	50	60	ns	2
T21	Write Enable False to DREQ True	23	58	75	ns	
T25	DREQ False to DREQ True	15	25	–	ns	
T26	Write Enable Width	5	–	–	ns	1
T27	Data Setup to End of Write Enable	5	–	–	ns	1
T28	Data Hold Time from End of Write Enable	0	–	–	ns	1
T29	EOT# Pulse Width	5	–	–	ns	3
T31	DREQ True to Write Enable True	9	–	–	ns	1

Notes:

1. *For Non-DMA Split Bus mode, Write Enable is the occurrence of DACK# and optionally, IOW#. For DMA Split Bus mode, Write Enable is the occurrence of DACK# and optionally, DMAWR#.*
2. *Minimum value is guaranteed only if the **DMAREQ** register DMA Request Enable bit is set.*
3. *EOT#, DACK#, and optionally, IOW# or DMAWR#, must all be true for at least T29 for proper recognition of the EOT# pulse.*



11.4.5.6 2.5V VDDIO Local Bus DMA Write – Fast Mode

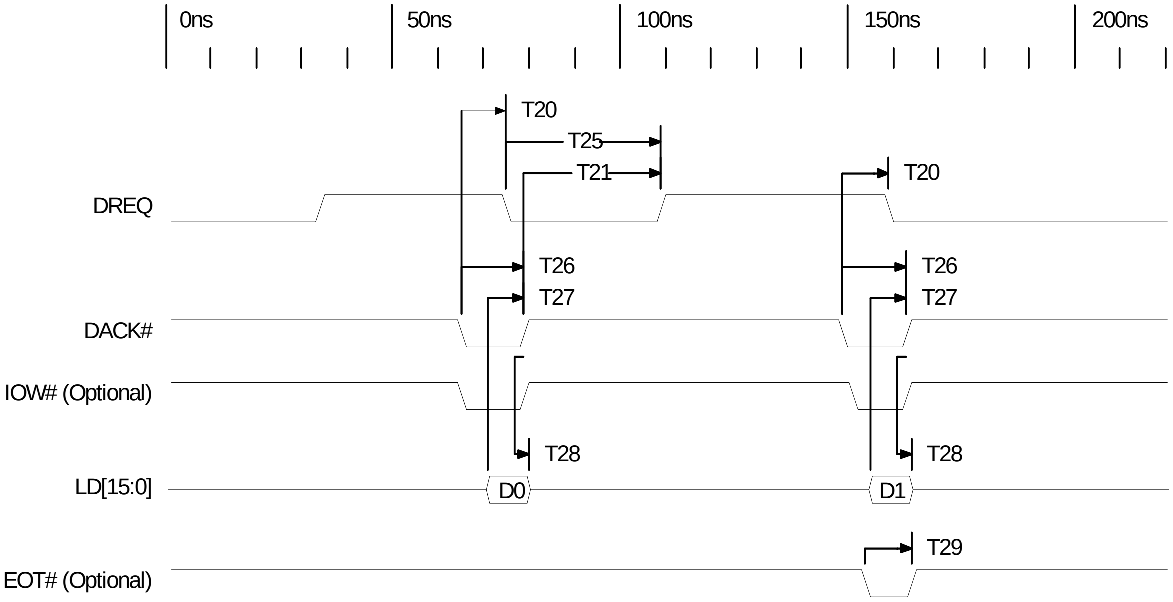
Operating Conditions – VDDC and VDDIO = 2.2 to 2.6V, T_A = 0 to 85°C, Output Load = 25 pF

Table 11-44. 2.5V VDDIO Local Bus DMA Write – Fast Mode Specifications

Symbol	Parameter	Min	Typ	Max	Unit	Notes
T20	Write Enable True to DREQ False	4	–	23	ns	2
T21	Write Enable False to DREQ True	16	45	55	ns	
T25	DREQ False to DREQ True	15	40	–	ns	
T26	Write Enable Width	5	–	–	ns	1
T27	Data Setup to End of Write Enable	5	–	–	ns	1
T28	Data Hold Time from End of Write Enable	0	–	–	ns	1
T29	EOT# Pulse Width	5	–	–	ns	3

Notes:

1. *For Non-DMA Split Bus mode, Write Enable is the occurrence of DACK# and optionally, IOW#. For DMA Split Bus mode, Write Enable is the occurrence of DACK# and optionally, DMAWR#.*
2. *Minimum value is guaranteed only if the* ***DMAREQ*** *register DMA Request Enable bit is set.*
3. *EOT#, DACK#, and optionally, IOW# or DMAWR#, must all be true for at least T29 for proper recognition of the EOT# pulse.*



11.4.5.7 2.5V VDDIO Local Bus DMA Write – Burst Mode

Operating Conditions – VDDC and VDDIO = 2.2 to 2.6V, T_A = 0 to 85°C, Output Load = 25 pF

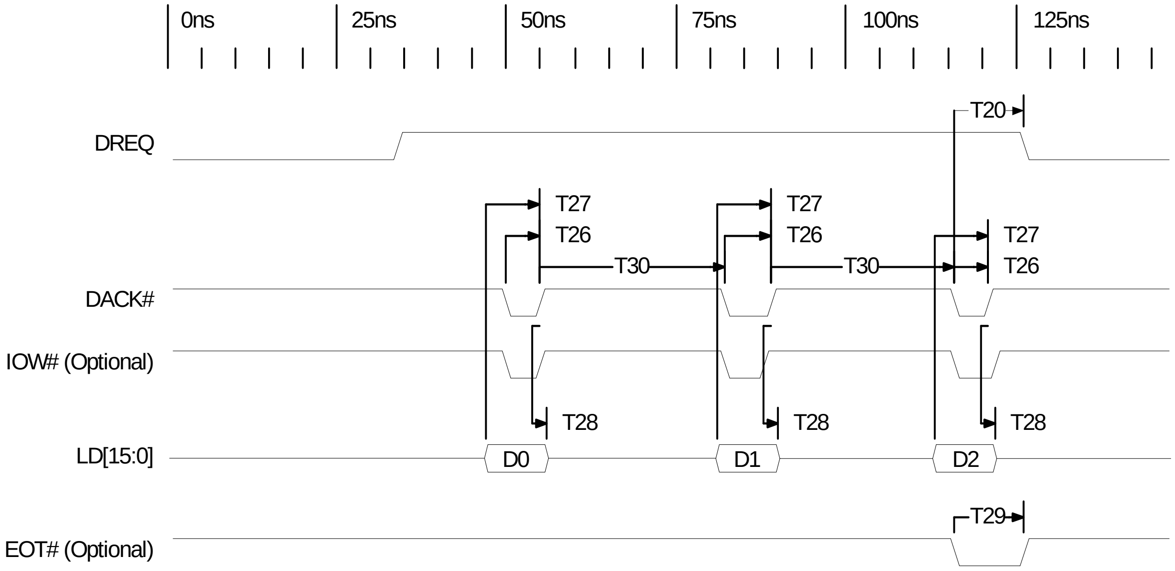
Table 11-45. 2.5V VDDIO Local Bus DMA Write – Burst Mode Specifications

Symbol	Parameter	Min	Typ	Max	Unit	Notes
T20	Write Enable True to DREQ False	4	–	23	ns	
T26	Write Enable Width	5	–	–	ns	1
T27	Data Setup to End of Write Enable	5	–	–	ns	1
T28	Data Hold Time from End of Write Enable	0	–	–	ns	1
T29	EOT# Pulse Width	5	–	–	ns	3
T30	DMA Write Recovery	36	–	–	ns	

Notes:

1. *For Non-DMA Split Bus mode, Write Enable is the occurrence of DACK# and optionally, IOW#. For DMA Split Bus mode, Write Enable is the occurrence of DACK# and optionally, DMAWR#.*

3. *EOT#, DACK#, and optionally, IOW# or DMAWR#, must all be true for at least T29 for proper recognition of the EOT# pulse.*



11.4.5.8 2.5V VDDIO Local Bus DMA Read – Slow Mode

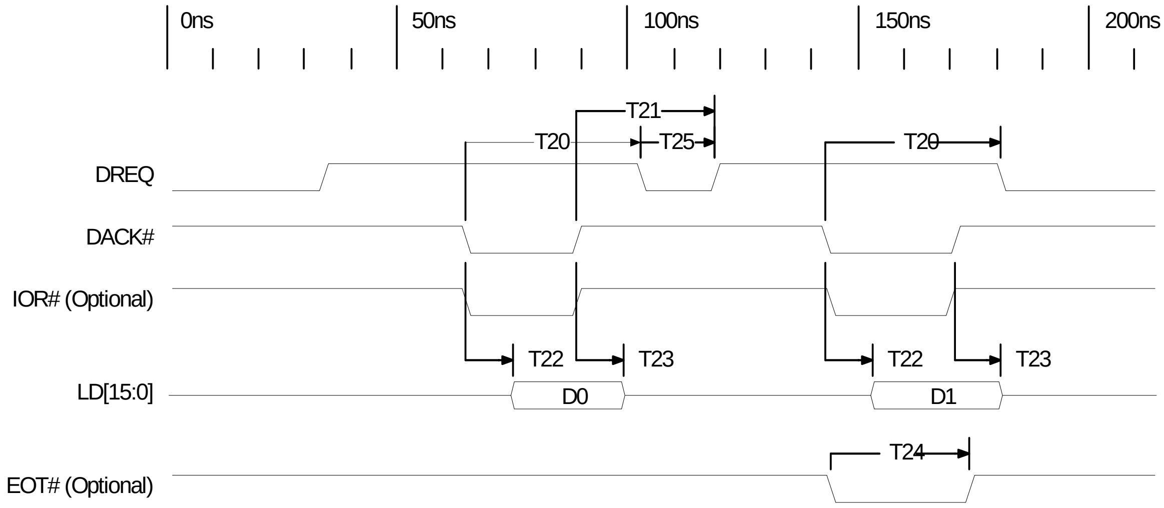
Operating Conditions – VDDC and VDDIO = 2.2 to 2.6V, T_A = 0 to 85°C, Output Load = 25 pF

Table 11-46. 2.5V VDDIO Local Bus DMA Read – Slow Mode Specifications

Symbol	Parameter	Min	Typ	Max	Unit	Notes
T20	Read Enable True to DREQ False	25	50	60	ns	2
T21	Read Enable False to DREQ True	16	50	68	ns	
T22	Data Access Time from Read Enable	–	–	17	ns	1
T23	Data Three-State Time from End of Read Enable	2	–	12	ns	1
T24	EOT# Pulse Width	10	–	–	ns	3
T25	DREQ False to DREQ True	15	25	–	ns	

Notes:

1. *For Non-DMA Split Bus mode, Read Enable is the occurrence of DACK# and optionally, IOR#. For DMA Split Bus mode, Read Enable is the occurrence of DACK# and optionally, DMARD#.*
2. *Minimum value is guaranteed only if the **DMAREQ** register DMA Request Enable bit is set.*
3. *EOT#, DACK#, and optionally, IOR# or DMARD#, must all be true for at least T24 for proper recognition of the EOT# pulse.*
4. *A recovery time of 2 ns is required between DMA Read Enable de-assertion and I/O Read Enable assertion. (This note is a general requirement not shown on the timing diagrams.)*



11.4.5.9 2.5V VDDIO Local Bus DMA Read – Fast Mode

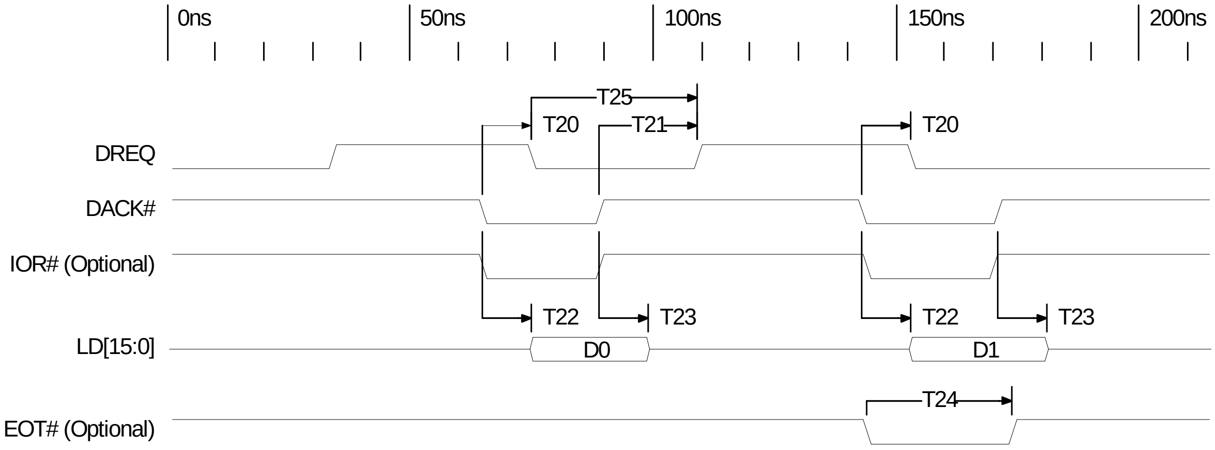
Operating Conditions – VDDC and VDDIO = 2.2 to 2.6V, T_A = 0 to 85°C, Output Load = 25 pF

Table 11-47. 2.5V VDDIO Local Bus DMA Read – Fast Mode Specifications

Symbol	Parameter	Min	Typ	Max	Unit	Notes
T20	Read Enable True to DREQ False	4	–	21	ns	2
T21	Read Enable False to DREQ True	4	35	45	ns	
T22	Data Access Time from Read Enable	–	–	17	ns	1
T23	Data Three-State Time from End of Read Enable	2	–	12	ns	1
T24	EOT# Pulse Width	10	–	–	ns	3
T25	DREQ False to DREQ True	15	36	–	ns	

Notes:

1. *For Non-DMA Split Bus mode, Read Enable is the occurrence of DACK# and optionally, IOR#. For DMA Split Bus mode, Read Enable is the occurrence of DACK# and optionally, DMARD#.*
2. *Minimum value is guaranteed only if the* **DMAREQ** *register DMA Request Enable bit is set.*
3. *EOT#, DACK#, and optionally, IOR# or DMARD#, must all be true for at least T24 for proper recognition of the EOT# pulse.*
4. *A recovery time of 2 ns is required between DMA Read Enable de-assertion and I/O Read Enable assertion. (This note is a general requirement not shown on the timing diagrams.)*



11.4.5.10 2.5V VDDIO Local Bus DMA Read – Burst Mode

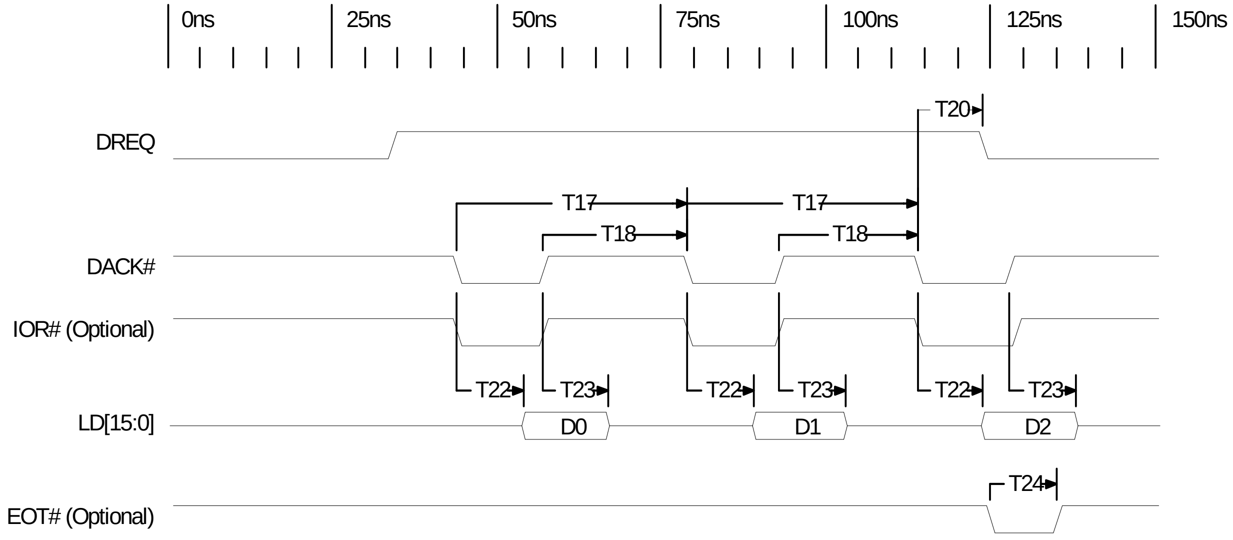
Operating Conditions – VDDC and VDDIO = 2.2 to 2.6V, T_A = 0 to 85°C, Output Load = 25 pF

Table 11-48. 2.5V VDDIO Local Bus DMA Read – Burst Mode Specifications

Symbol	Parameter	Min	Typ	Max	Unit	Notes
T17	DMA Read Cycle Time	36	–	–	ns	
T18	DMA Read Recovery Time	19	–	–	ns	
T20	Read Enable True to DREQ False	4	–	22	ns	
T22	Data Access Time from Read Enable	–	–	17	ns	1
T23	Data Three-State Time from End of Read Enable	2	–	12	ns	1
T24	EOT# Pulse Width	10	–	–	ns	3

Notes:

1. *For Non-DMA Split Bus mode, Read Enable is the occurrence of DACK# and optionally, IOR#. For DMA Split Bus mode, Read Enable is the occurrence of DACK# and optionally, DMARD#.*
3. *EOT#, DACK#, and optionally, IOR# or DMARD#, must all be true for at least T24 for proper recognition of the EOT# pulse.*
4. *A recovery time of 2 ns is required between DMA Read Enable de-assertion and I/O Read Enable assertion. (This note is a general requirement not shown on the timing diagrams.)*



11.4.6 3.3V VDDIO Local Bus AC Specifications

11.4.6.1 3.3V VDDIO Local Bus Non-Multiplexed Write

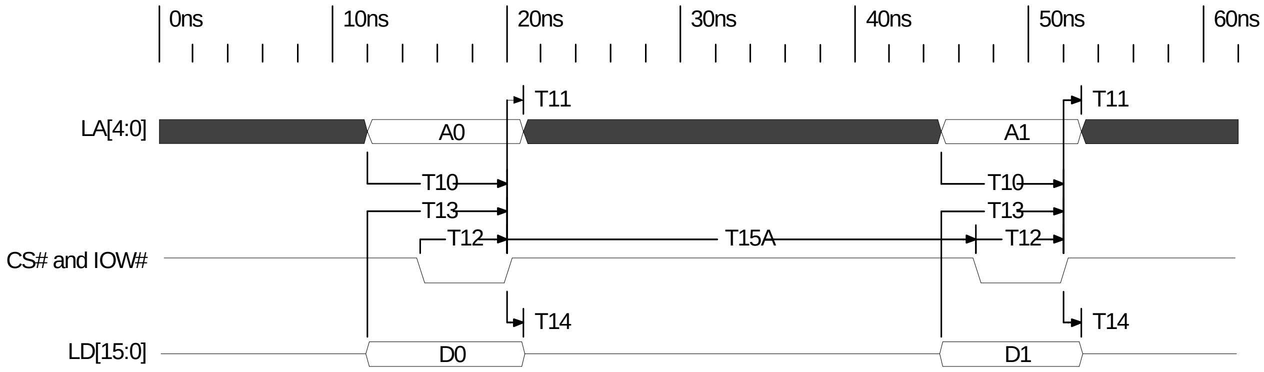
Operating Conditions – VDDC = 2.2 to 2.6V, VDDIO = 3.2 to 3.5V, T_A = 0 to 85°C, Output Load = 25 pF

Table 11-49. 3.3V VDDIO Local Bus Non-Multiplexed Write Specifications

Symbol	Parameter	Min	Max	Unit	Notes
T10	Address Setup to End of Write Enable	5	–	ns	1
T11	Address Hold from End of Write Enable	0	–	ns	1
T12	Write Enable Width	5	–	ns	1
T13	Data Setup to End of Write Enable	5	–	ns	1
T14	Data Hold Time from End of Write Enable	0	–	ns	1
T15A	Recovery Time to Next Write	28	–	ns	
T15B	Recovery Time to Next Read	52	–	ns	2

Notes:

1. *Write Enable is the occurrence of both CS# and IOW#.*
2. *Because reading and writing to EP_DATA cause EP_AVAIL**x** and EP_TRANSFER**x** to change values, it is necessary to increase the recovery time to 70 ns between a read or write to EP_DATA and a read from EP_AVAIL**x** or EP_TRANSFER**x**.*



11.4.6.2 3.3V VDDIO Local Bus Multiplexed Write

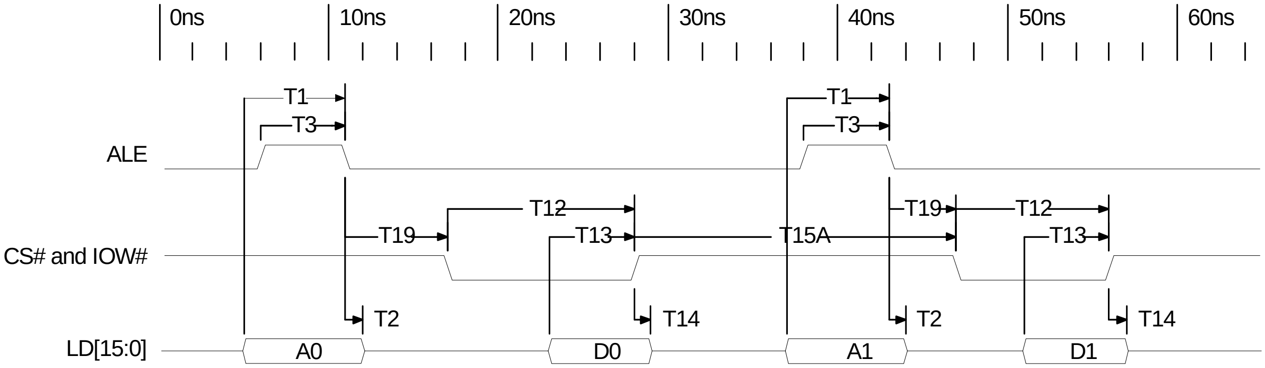
Operating Conditions – VDDC = 2.2 to 2.6V, VDDIO = 3.2 to 3.5V, T_A = 0 to 85°C, Output Load = 25 pF

Table 11-50. 3.3V VDDIO Local Bus Multiplexed Write Specifications

Symbol	Parameter	Min	Max	Unit	Notes
T1	Address Setup to Falling Edge of ALE	5	–	ns	
T2	Address Hold from Falling Edge of ALE	1	–	ns	
T3	ALE width	5	–	ns	
T12	Write Enable Width	5	–	ns	1
T13	Data Setup to End of Write Enable	5	–	ns	1
T14	Data Hold Time from End of Write Enable	0	–	ns	1
T15A	Recovery Time to Next Write	28	–	ns	
T15B	Recovery Time to Next Read	52	–	ns	2
T19	ALE Falling Edge to Write Enable	1	–	ns	1

Notes:

1. *Write Enable is the occurrence of both CS# and IOW#.*
2. *Because reading and writing to **EP_DATA** cause **EP_AVAILx** and **EP_TRANSFERx** to change values, it is necessary to increase the recovery time to 70 ns between a read or write to **EP_DATA** and a read from **EP_AVAILx** or **EP_TRANSFERx**.*



11.4.6.3 3.3V VDDIO Local Bus Non-Multiplexed Read

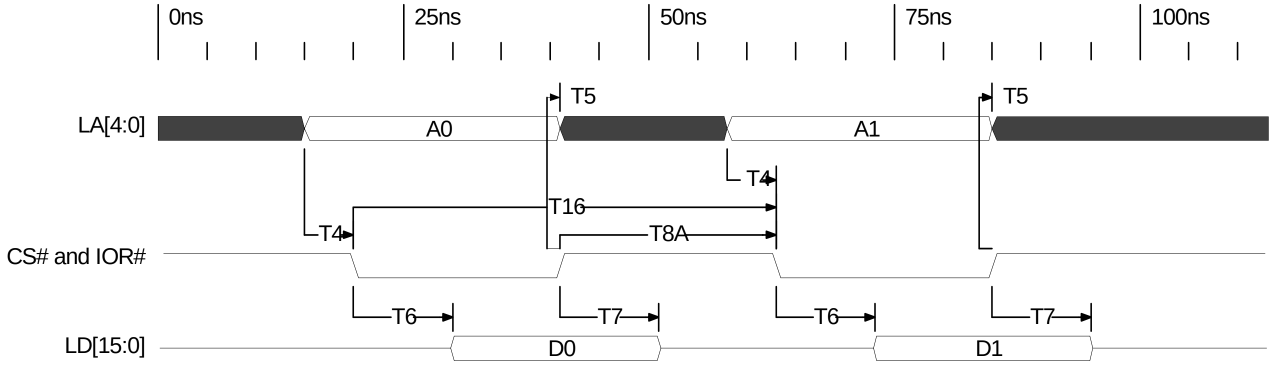
Operating Conditions – VDDC = 2.2 to 2.6V, VDDIO = 3.2 to 3.5V, T_A = 0 to 85°C, Output Load = 25 pF

Table 11-51. 3.3V VDDIO Local Bus Non-Multiplexed Read Specifications

Symbol	Parameter	Min	Max	Unit	Notes
T4	Address Setup to Read Enable	-1	–	ns	1
T5	Address Hold from End of Read Enable	-2	–	ns	1
T6	Data Access Time from LA Valid or Read Enable Asserted (Whichever occurs Later)	–	18	ns	1
T7	Data Three-State Time from End of Read Enable	2	11	ns	1
T8A	Recovery Time to Next Read	19	–	ns	2
T8B	Recovery Time to Next Write	19	–	ns	
T16	Read Cycle Time	37	–	ns	

Notes:

1. *Read Enable is the occurrence of both CS# and IOR#.*
2. *Because reading and writing to **EP_DATA** cause **EP_AVAILx** and **EP_TRANSFERx** to change values, it is necessary to increase the recovery time to 37 ns between a read or write to **EP_DATA** and a read from **EP_AVAILx** or **EP_TRANSFERx**.*



11.4.6.4 3.3V VDDIO Local Bus Multiplexed Read

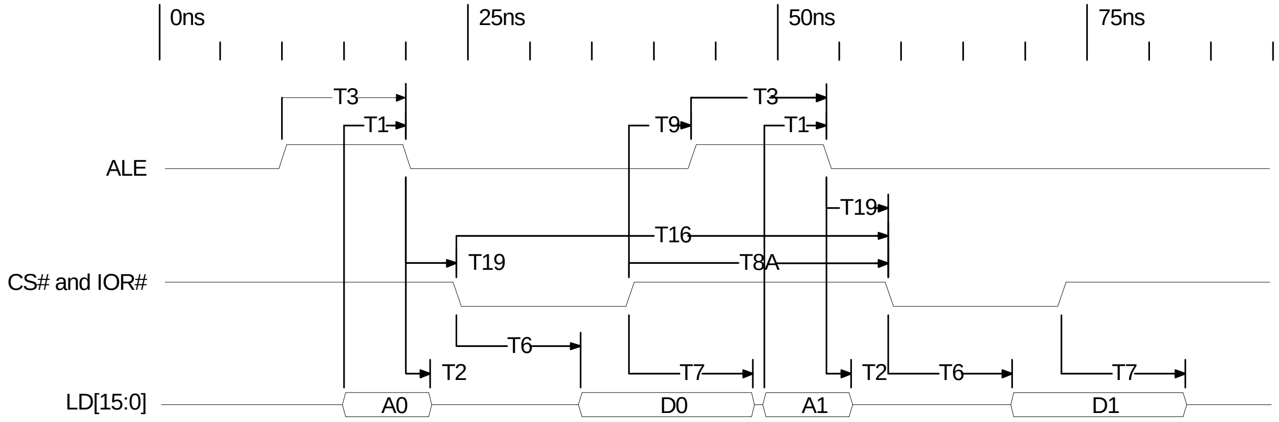
Operating Conditions – VDDC = 2.2 to 2.6V, VDDIO = 3.2 to 3.5V, T_A = 0 to 85°C, Output Load = 25 pF

Table 11-52. 3.3V VDDIO Local Bus Multiplexed Read Specifications

Symbol	Parameter	Min	Max	Unit	Notes
T1	Address Setup to Falling Edge of ALE	5	–	ns	
T2	Address Hold from Falling Edge of ALE	1	–	ns	
T3	ALE width	5	–	ns	
T6	Data Access Time from Read Enable	–	18	ns	1
T7	Data Three-State Time from End of Read Enable	2	11	ns	1
T8A	Recovery Time to Next Read	19	–	ns	2
T8B	Recovery Time to Next Write	19	–	ns	
T9	Recovery Time to Next ALE	5	–	ns	
T16	Read Cycle Time	37	–	ns	
T19	ALE Falling Edge to Read Enable	1	–	ns	1

Notes:

1. *Read Enable is the occurrence of both CS# and IOR#.*
2. *Because reading and writing to **EP_DATA** cause **EP_AVAILx** and **EP_TRANSFERx** to change values, it is necessary to increase the recovery time to 37 ns between a read or write to **EP_DATA** and a read from **EP_AVAILx** or **EP_TRANSFERx**.*



11.4.6.5 3.3V VDDIO Local Bus DMA Write – Slow Mode

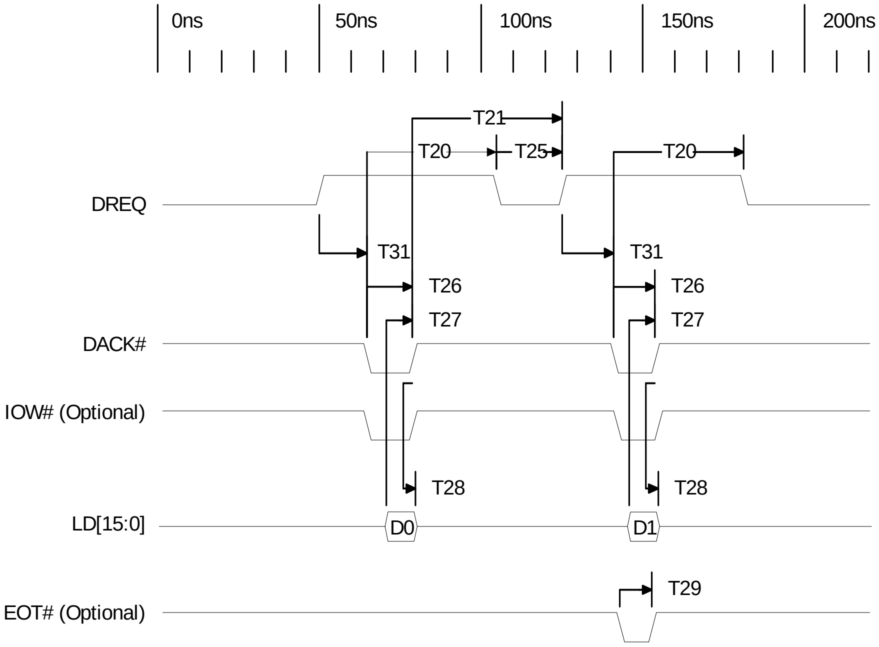
Operating Conditions – VDDC = 2.2 to 2.6V, VDDIO = 3.2 to 3.5V, T_A = 0 to 85°C, Output Load = 25 pF

Table 11-53. 3.3V VDDIO Local Bus DMA Write – Slow Mode Specifications

Symbol	Parameter	Min	Typ	Max	Unit	Notes
T20	Write Enable True to DREQ False	17	50	60	ns	2
T21	Write Enable False to DREQ True	23	50	72	ns	
T25	DREQ False to DREQ True	15	25	–	ns	
T26	Write Enable Width	5	–	–	ns	1
T27	Data Setup to End of Write Enable	5	–	–	ns	1
T28	Data Hold Time from End of Write Enable	0	–	–	ns	1
T29	EOT# Pulse Width	5	–	–	ns	3
T31	DREQ True to Write Enable True	9	–	–	ns	1

Notes:

1. *For Non-DMA Split Bus mode, Write Enable is the occurrence of DACK# and optionally, IOW#. For DMA Split Bus mode, Write Enable is the occurrence of DACK# and optionally, DMAWR#.*
2. *Minimum value is guaranteed only if the* ***DMAREQ*** *register DMA Request Enable bit is set.*
3. *EOT#, DACK#, and optionally, IOW# or DMAWR#, must all be true for at least T29 for proper recognition of the EOT# pulse.*



11.4.6.6 3.3V VDDIO Local Bus DMA Write – Fast Mode

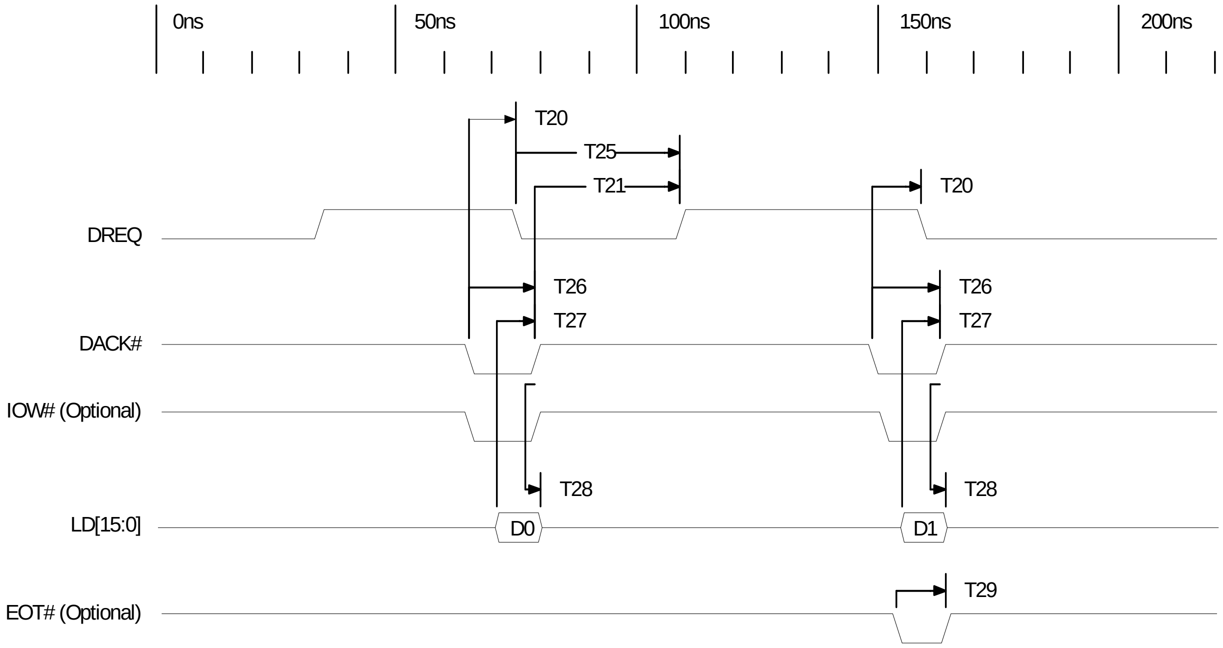
Operating Conditions – VDDC = 2.2 to 2.6V, VDDIO = 3.2 to 3.5V, T_A = 0 to 85°C, Output Load = 25 pF

Table 11-54. 3.3V VDDIO Local Bus DMA Write – Fast Mode Specifications

Symbol	Parameter	Min	Typ	Max	Unit	Notes
T20	Write Enable True to DREQ False	4	–	19	ns	2
T21	Write Enable False to DREQ True	16	45	55	ns	
T25	DREQ False to DREQ True	15	40	–	ns	
T26	Write Enable Width	5	–	–	ns	1
T27	Data Setup to End of Write Enable	5	–	–	ns	1
T28	Data Hold Time from End of Write Enable	0	–	–	ns	1
T29	EOT# Pulse Width	5	–	–	ns	3

Notes:

1. *For Non-DMA Split Bus mode, Write Enable is the occurrence of DACK# and optionally, IOW#. For DMA Split Bus mode, Write Enable is the occurrence of DACK# and optionally, DMAWR#.*
2. *Minimum value is guaranteed only if the* ***DMAREQ*** *register DMA Request Enable bit is set.*
3. *EOT#, DACK#, and optionally, IOW# or DMAWR#, must all be true for at least T29 for proper recognition of the EOT# pulse.*



11.4.6.7 3.3V VDDIO Local Bus DMA Write – Burst Mode

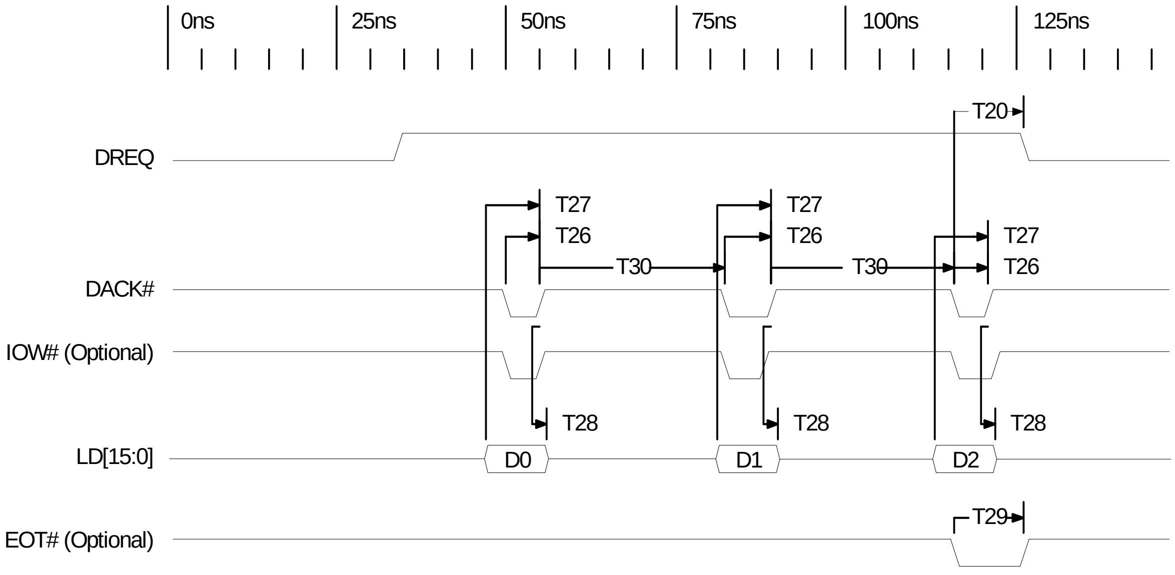
Operating Conditions – VDDC = 2.2 to 2.6V, VDDIO = 3.2 to 3.5V, T_A = 0 to 85°C, Output Load = 25 pF

Table 11-55. 3.3V VDDIO Local Bus DMA Write – Burst Mode Specifications

Symbol	Parameter	Min	Typ	Max	Unit	Notes
T20	Write Enable True to DREQ False	4	–	20	ns	
T26	Write Enable Width	5	–	–	ns	1
T27	Data Setup to End of Write Enable	5	–	–	ns	1
T28	Data Hold Time from End of Write Enable	0	–	–	ns	1
T29	EOT# Pulse Width	5	–	–	ns	3
T30	DMA Write Recovery	36	–	–	ns	

Notes:

1. *For Non-DMA Split Bus mode, Write Enable is the occurrence of DACK# and optionally, IOW#. For DMA Split Bus mode, Write Enable is the occurrence of DACK# and optionally, DMAWR#.*
3. *EOT#, DACK#, and optionally, IOW# or DMAWR#, must all be true for at least T29 for proper recognition of the EOT# pulse.*



11.4.6.8 3.3V VDDIO Local Bus DMA Read – Slow Mode

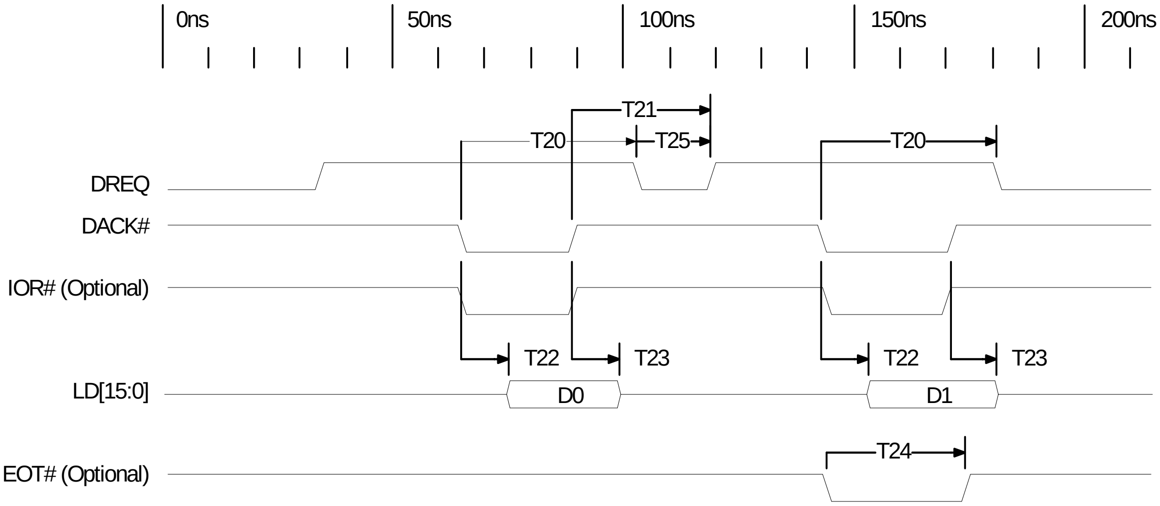
Operating Conditions – VDDC = 2.2 to 2.6V, VDDIO = 3.2 to 3.5V, T_A = 0 to 85°C, Output Load = 25 pF

Table 11-56. 3.3V VDDIO Local Bus DMA Read – Slow Mode Specifications

Symbol	Parameter	Min	Typ	Max	Unit	Notes
T20	Read Enable True to DREQ False	25	50	60	ns	2
T21	Read Enable False to DREQ True	16	50	68	ns	
T22	Data Access Time from Read Enable	–	–	16	ns	1
T23	Data Three-State Time from End of Read Enable	2	–	12	ns	1
T24	EOT# Pulse Width	10	–	–	ns	3
T25	DREQ False to DREQ True	15	25	–	ns	

Notes:

1. *For Non-DMA Split Bus mode, Read Enable is the occurrence of DACK# and optionally, IOR#. For DMA Split Bus mode, Read Enable is the occurrence of DACK# and optionally, DMARD#.*
2. *Minimum value is guaranteed only if the* ***DMAREQ*** *register DMA Request Enable bit is set.*
3. *EOT#, DACK#, and optionally, IOR# or DMARD#, must all be true for at least T24 for proper recognition of the EOT# pulse.*
4. *A recovery time of 2 ns is required between DMA Read Enable de-assertion and I/O Read Enable assertion. (This note is a general requirement not shown on the timing diagrams.)*



11.4.6.9 3.3V VDDIO Local Bus DMA Read – Fast Mode

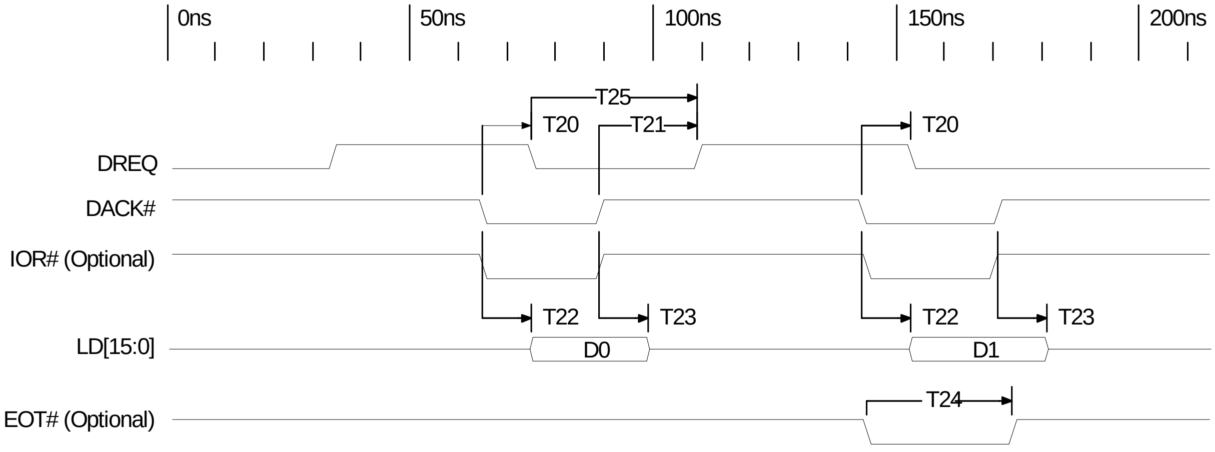
Operating Conditions – VDDC = 2.2 to 2.6V, VDDIO = 3.2 to 3.5V, T_A = 0 to 85°C, Output Load = 25 pF

Table 11-57. 3.3V VDDIO Local Bus DMA Read – Fast Mode Specifications

Symbol	Parameter	Min	Typ	Max	Unit	Notes
T20	Read Enable True to DREQ False	4	–	19	ns	2
T21	Read Enable False to DREQ True	4	35	45	ns	
T22	Data Access Time from Read Enable	–	–	16	ns	1
T23	Data Three-State Time from End of Read Enable	2	–	12	ns	1
T24	EOT# Pulse Width	10	–	–	ns	3
T25	DREQ False to DREQ True	15	35	–	ns	

Notes:

1. *For Non-DMA Split Bus mode, Read Enable is the occurrence of DACK# and optionally, IOR#. For DMA Split Bus mode, Read Enable is the occurrence of DACK# and optionally, DMARD#.*
2. *Minimum value is guaranteed only if the* ***DMAREQ*** *register DMA Request Enable bit is set.*
3. *EOT#, DACK#, and optionally, IOR# or DMARD#, must all be true for at least T24 for proper recognition of the EOT# pulse.*
4. *A recovery time of 2 ns is required between DMA Read Enable de-assertion and I/O Read Enable assertion. (This note is a general requirement not shown on the timing diagrams.)*



11.4.6.10 3.3V VDDIO Local Bus DMA Read – Burst Mode

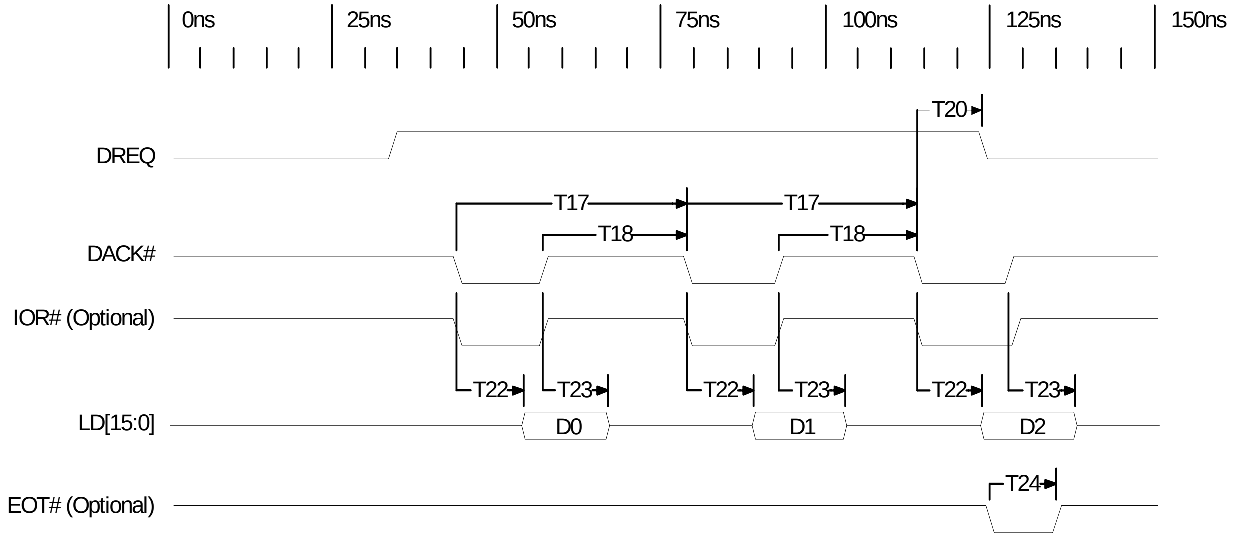
Operating Conditions – VDDC = 2.2 to 2.6V, VDDIO = 3.2 to 3.5V, T_A = 0 to 85°C, Output Load = 25 pF

Table 11-58. 3.3V VDDIO Local Bus DMA Read – Burst Mode Specifications

Symbol	Parameter	Min	Typ	Max	Unit	Notes
T17	DMA Read Cycle Time	35	–	–	ns	
T18	DMA Read Recovery Time	19	–	–	ns	
T20	Read Enable True to DREQ False	4	–	20	ns	
T22	Data Access Time from Read Enable	–	–	16	ns	1
T23	Data Three-State Time from End of Read Enable	2	–	12	ns	1
T24	EOT# Pulse Width	10	–	–	ns	3

Notes:

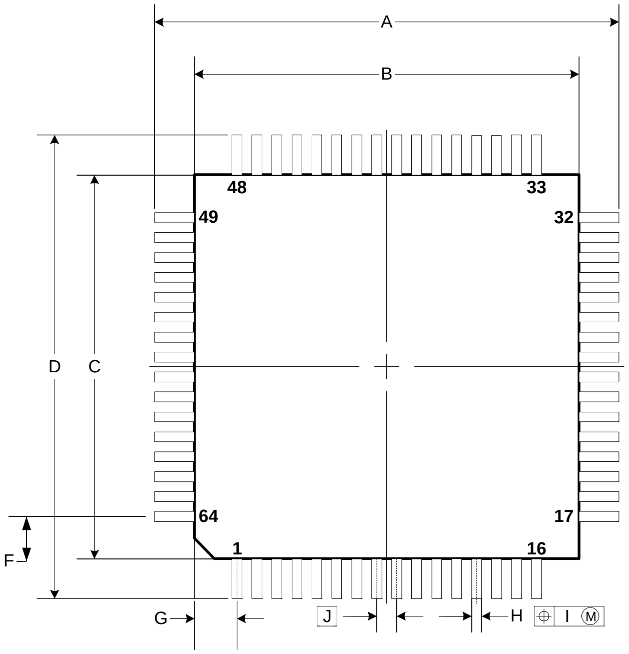
1. *For Non-DMA Split Bus mode, Read Enable is the occurrence of DACK# and optionally, IOR#. For DMA Split Bus mode, Read Enable is the occurrence of DACK# and optionally, DMARD#.*
3. *EOT#, DACK#, and optionally, IOR# or DMARD#, must all be true for at least T24 for proper recognition of the EOT# pulse.*
4. *A recovery time of 2 ns is required between DMA Read Enable de-assertion and I/O Read Enable assertion. (This note is a general requirement not shown on the timing diagrams.)*

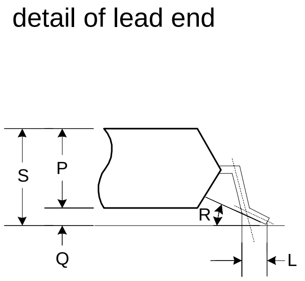


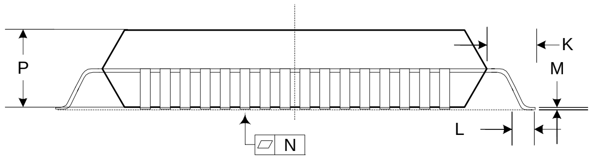
Chapter 12 Mechanical Specifications

12.1 64-Pin Plastic TQFP (10 x 10 mm)

Figure 12-1. NET2272 Mechanical Drawing (64-Pin Plastic TQFP Package)





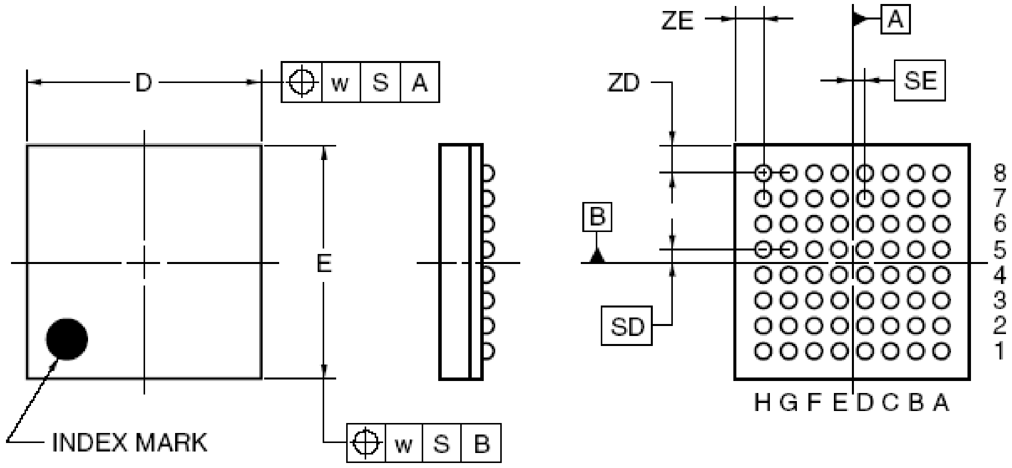


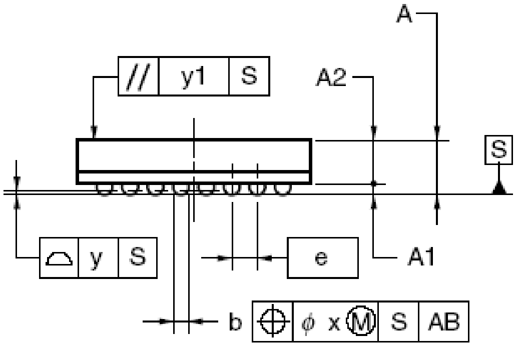
ITEM	MILLIMETERS	INCHES
A	12.0±0.2	$0.472\pm^{0.009}_{0.008}$
B	10.0±0.2	$0.394\pm^{0.008}_{0.009}$
C	10.0±0.2	$0.394\pm^{0.008}_{0.009}$
D	12.0±0.2	$0.472\pm^{0.009}_{0.008}$
F	1.25	0.049
G	1.25	0.049
H	$0.22\pm^{0.055}_{0.045}$	0.009±0.002
I	0.10	0.004
J	0.5 (T.P.)	0.020 (T.P.)
K	1.0±0.2	$0.039\pm^{0.009}_{0.008}$
L	0.5±0.2	$0.020\pm^{0.008}_{0.009}$
M	$0.145\pm^{0.055}_{0.045}$	0.006±0.002
N	0.10	0.004
P	1.0±0.1	$0.039\pm^{0.005}_{0.004}$
Q	0.1±0.05	0.004±0.002
R	$3°\pm^{7°}_{3°}$	$3°\pm^{7°}_{3°}$
S	1.27 MAX	0.050 MAX

NOTE
Each lead centerline is located within 0.13 mm of its true position (T.P.) at maximum material condition.

12.2 64-Ball Plastic FBGA (6 x 6 mm)

Figure 12-2. NET2272 Mechanical Drawing (64-Ball Plastic FBGA Package)





(UNIT:mm)

ITEM	DIMENSIONS
D	6.00±0.10
E	6.00±0.10
w	0.20
A	1.43±0.10
A1	0.30±0.05
A2	1.13
e	0.65
SD	0.325
SE	0.325
b	0.40±0.05
x	0.08
y	0.10
y1	0.20
ZD	0.725
ZE	0.725

Appendix A General Information

A.1 Product Ordering Information

Contact your local PLX sales representative for ordering information.

Table A-1. Product Ordering Information

Part Numbers	Description
NET2272REV1A-LF	Lead-Free ROHS Green 64-Pin Plastic TQFP Local Bus to USB 2.0 Hi-Speed Peripheral Controller, 10 x 10 mm
NET2272REV1A-BC F	Lead-Free ROHS Green 64-Ball Plastic FBGA Local Bus to USB 2.0 Hi-Speed Peripheral Controller, 6 x 6 mm
NET2272PCI-RDK	NET2272 Reference Design Kit

Where: *NET = NetChip USB 2.0 Controller product family*
2272 = Local Bus to USB 2.0 Peripheral Controller
REV1A = Silicon Revision
B = Package Type (FBGA) (if available)
C = 0 to +70°C Commercial Operating Temperature range (if available)
LF, F = Lead-Free ROHS Packaging (if available)

A.2 United States and International Representatives, and Distributors

PLX Technology, Inc., representatives and distributors are listed at www.plxtech.com.

A.3 Technical Support

PLX Technology, Inc., technical support information is listed at www.plxtech.com/support/, or call 408 774-9060 or 800 759-3735.