NetChip
Technology, Inc.

335 Pioneer Way
Mt View, California 94041
(650) 526-1490 Fax (650) 526-1494
e-mail: sales@netchip.com
Internet: www.netchip.com

NET2272 USB 2.0 Peripheral Controller

Patent Pending

For Revision 1A

Doc #: 605-0213-0110
Revision: 1.2
Date: October 15, 2003

Revision History

Revision	Issue Date	Comments
1.0	May 5, 2003	Revision 1 silicon initial release
1.1	October 7, 2003	Revision 1.1 silicon release
1.2	October 15, 2003	Power consumption update

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

NET2272 USB Peripheral Controller

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com
Rev 1.2, October 15, 2003

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

1 Introduction

1.1 Features

- USB Specification Version 2.0 Compliant (high and full speed)
- Interfaces between a local CPU bus and a USB bus
- Supports USB Full Speed (12 Mbps) and High Speed (480 Mbps)
- Supports optional Split Bus DMA, with dedicated DMA and CPU access
- Provides 3 Configurable Physical Endpoints, in addition to Endpoint 0
- Provides 30 Configurable Virtual Endpoints
- Each endpoint can be Isochronous, Bulk, or Interrupt, as well as IN or OUT
- Supports high-bandwidth isochronous mode
- Supports Max Packet Size up to 1K bytes, double buffered
- Internal 3 Kbyte memory provides transmit and receive buffers
- Local CPU bus easily interfaces to generic CPUs
- 8-bit or 16-bit CPU or DMA bus transfers
- Multiple register address modes supports both 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, simplifying software
- Low power CMOS in 64 Pin Plastic TQFP Package
- 30 MHz oscillator with internal phase-lock loop multiplier
- Provides an output clock to the local bus - 8 programmable frequencies from OFF to 60 MHz
- 2.5V, 3.3V operating voltages with 5V tolerant I/O

1.2 Overview

The NET2272 USB Peripheral Controller allows control, isochronous, bulk and interrupt transfers between a local bus and a Universal Serial Bus (USB). The NET2272 supports the Device side of a connection between a USB host computer and intelligent peripherals such as image scanners, printers, and digital cameras.

The six main modules of the NET2272 are the USB Transceiver, Serial Interface Engine, USB Protocol Controller, Endpoint Packet Buffers, Local Bus Interface, and the Configuration Registers.

USB Transceiver:

- Supports Full Speed (12 Mbps) or High Speed (480 Mbps) operation
- Serial data transmitter and receiver
- Parallel data interface to SIE
- Single parallel data clock output with on-chip PLL to generate higher speed serial data clocks
- Data and clock recovery from USB serial data stream
- SYNC/EOP generation and checking
- Bit-stuffing/unstuffing; bit stuff error detection
- Logic to facilitate Resume signaling
- Logic to facilitate Wake Up and Suspend detection
- Ability to switch between Full-Speed and High-Speed terminations/signaling

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

Serial Interface Engine (SIE):
- Interface between Packet Buffers and USB transceiver
- CRC generator and checker
- Packet Identifier (PID) decoder
- Forced Error Conditions
- USB 2.0 Test Modes

USB Protocol Controller
- Host to Device Communication
- Automatic retry of failed packets
- Up to 3 Isochronous, Bulk, or Interrupt physical endpoints, each with a configurable packet buffer
- Up to 30 virtual endpoints can be mapped to the physical endpoints
- Configurable Control Endpoint 0
- Interface to packet buffers
- Software controlled disconnect signaling allows device enumeration
- Software control of USB suspend and root port reset detection
- Software controlled device remote wakeup
- Software control of root port wakeup

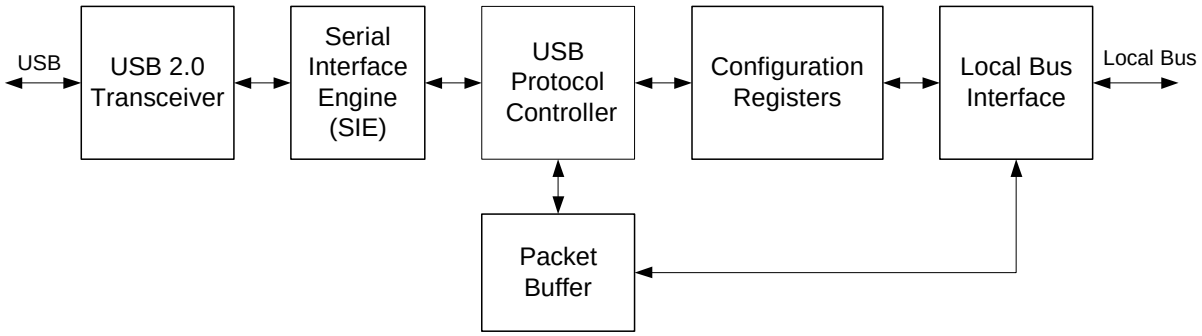
Endpoint Packet Buffers
- Choice from 4 preset configurations simplify programming
- Separate 128 byte, packet buffer for physical endpoint 0
- 3 Kbytes of configurable packet buffer memory for physical endpoints A, B, and C
- Supports Max Packet Size up to 1K bytes, double buffered

Local Bus Interface
- Provides slave interfaces to 8-bit or 16-bit CPU
- Provides access to internal Transmit and Receive packet buffers.
- Supports Split DMA transactions (DMA and CPU on separate data bus)
- Supports DMA burst mode.
- Local interface supports both DMA and Interrupt transfers
- Supports optional multiplexed Address/Data bus using ALE for low pin count applications
- Supports indirect addressing, allowing access to all registers with only a single address bit
- Supports 5V tolerant I/O

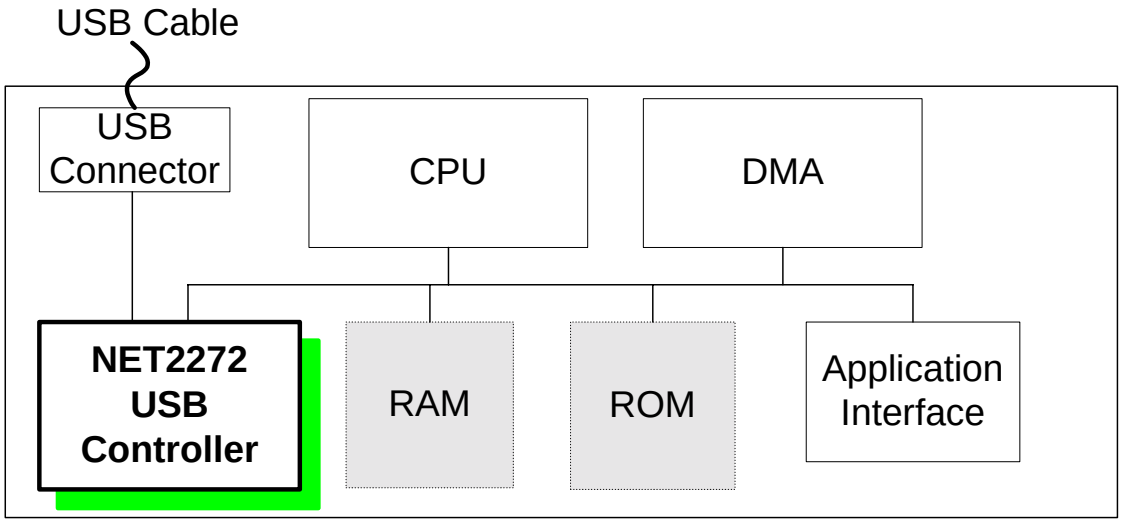
Configuration Registers
- Internal registers are accessible from the local bus
- Main registers for common functions
- USB Registers for the USB Interface Module
- Control registers for each endpoint

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

1.3 NET2272 Block Diagram

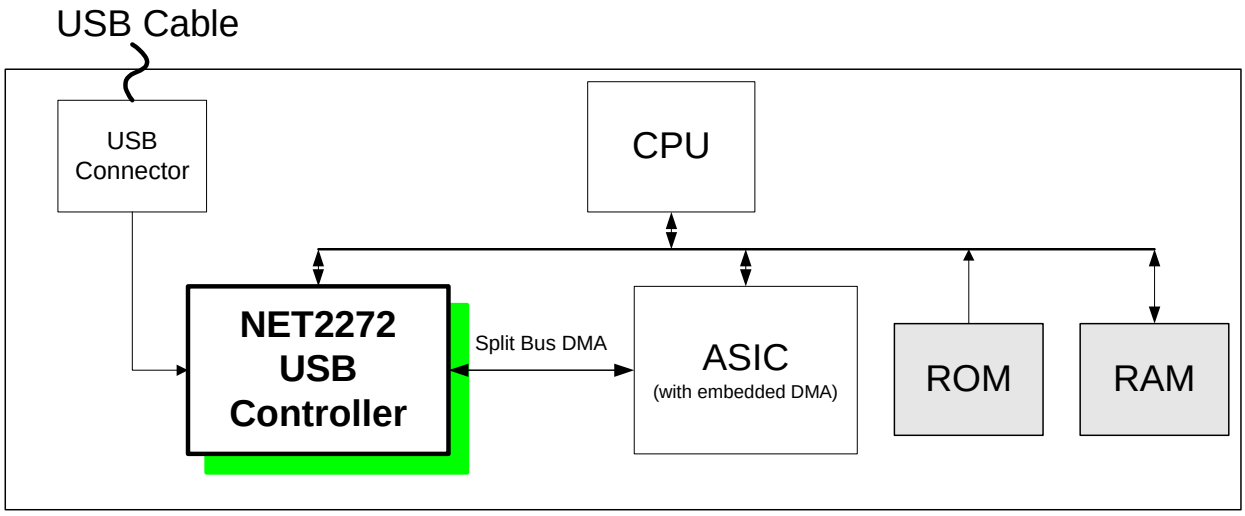


1.4 NET2272 Typical System Block Diagrams



CPU-based Device Controller

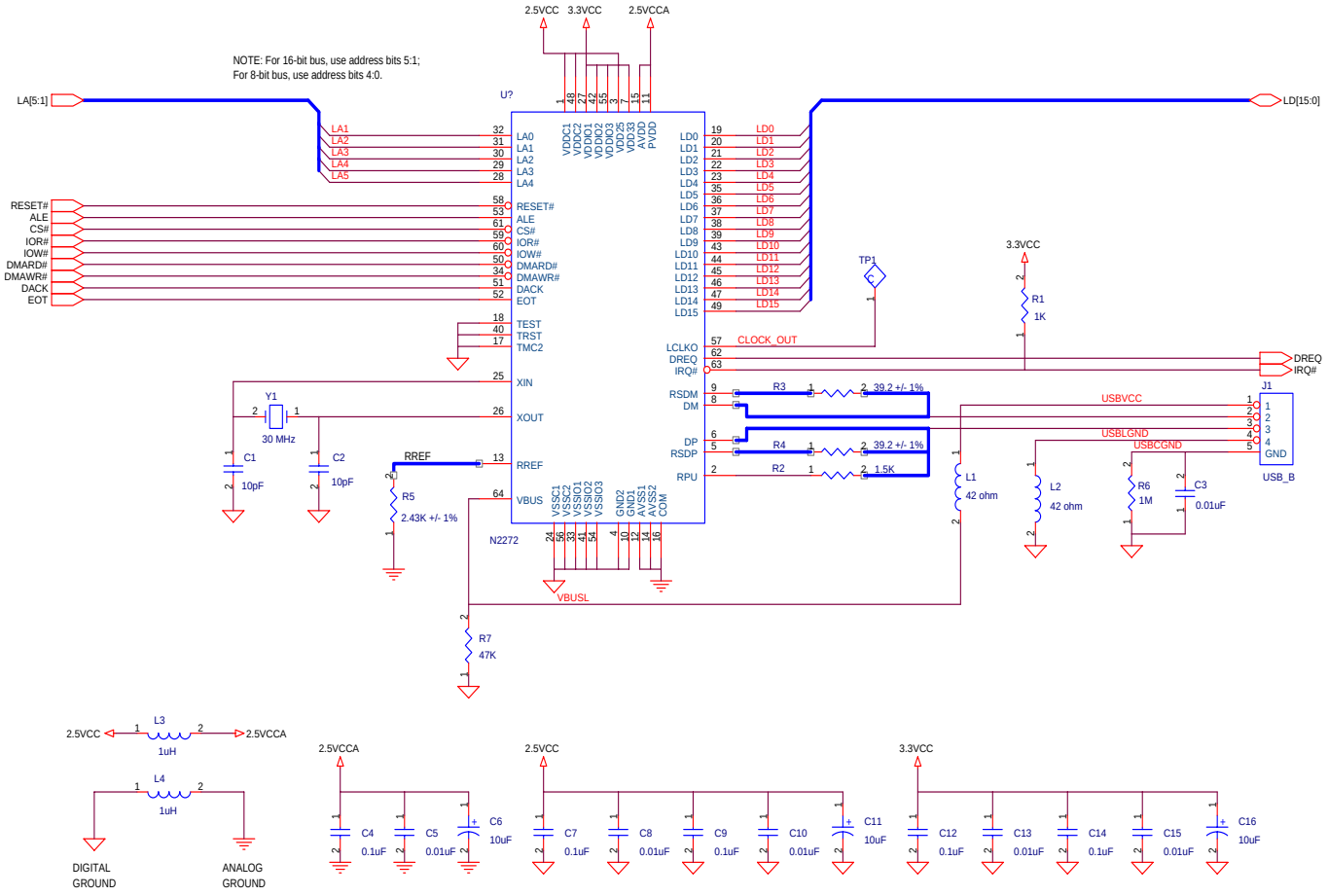
335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com



ASIC with Split Bus DMA

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

1.5 Example connections to NET2272



335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

1.5.1 Example Part Numbers

Part	Manufacturer	Part Number	Website
30 MHz Fundamental Crystal (Y1)	KDS	AT-49 30.000-16	http://www.kdsj.co.jp/english.html
Ferrite Beads (L1-L2)	Taiyo Yuden	FBMJ2125HS420-T	http://www.t-yuden.com/ferritebeads/index.cfm
1 µH Inductor, 10%, 0805 Pkg (L3-L4)	Taiyo Yuden	LK21251R0K	http://www.t-yuden.com/inductors/index.cfm
2.43K, 1% resistor, 0.1 Watt, 0603 Pkg (R5)	Panasonic	ERJ6ENF2431V	http://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	http://www.newnex.com

Note that the crystal should have a tolerance of +/- 0.005% (50 ppm) to guarantee a data rate of 480 Mbps +/- 500 ppm.

1.5.2 General PCB Layout Guidelines

USB2.0 high-speed 480 Mbits/sec data transfers utilize 400 mV differential signaling. This requires special Printed Circuit Board layout requirements. Intel provides some USB layout guidelines in the following document: http://www.usb.org/developers/docs/hs_usb_pdg_r1_0.pdf. In addition, NetChip provides the following guidelines:

The following guidelines must be followed to insure proper operation of the NET2272. It is strongly suggested that schematics and PCB layout be submitted to support@netchip.com for review prior to PCB fabrication.

1.5.2.1 USB Differential Signals

- Consult with board manufacturer for determining layer separation, trace width, and trace separation for maintaining differential impedance of 90 ohms.
- Maintain equal trace lengths for D+ and D-.
- Minimize number of vias and curves on D+ and D- traces.
- Use two 45 degree turns instead of one 90 degree turn.
- Minimize trace lengths shown in **bold** in the schematic in section 1.5.
- Prevent D+ and D- traces from crossing a power plane void. The same ground layer shall be kept next to the D+ and D- traces.
- Digital Ground (VSS) layer should be placed next to the layer where D+ and D- are routed.
- Avoid using studs or test points for observing USB signals.
- Maximize the distance of D+ and D- from other signals to prevent crosstalk.

1.5.2.2 Analog VDD (power)

- Analog power must be filtered from the digital power using the recommended circuit provided.
- Analog VDD and digital VDD must be connected via 1µH inductor.
- Analog VDD must be separated from digital VDD. If analog VDD and digital VDD are in the same layer, split the layer to accommodate the two power signals.
- AVDD and PVDD pins should be connected to analog VDD.

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

1.5.2.3 Analog VSS (ground)

- Analog ground must be filtered from the digital ground using the recommended circuit provided.
- Analog VSS and digital VSS must be connected via a 1µH inductor.
- AVSS, PVSS, COM pins, and RREF's resistor should be connected to analog VSS.

1.5.2.4 Decoupling Capacitors

- At least one 0.1µF decoupling capacitor for every two pairs of digital/analog VDD and VSS should be located near the NET2272 device.
- At least one 0.01µF decoupling capacitor for every two pairs of digital/analog VDD and VSS should be located near the NET2272 device.
- Decoupling capacitors may be placed on the solder side of the PCB.
- At least one 10µF filter capacitor for every five 0.1µF or 0.01µF decoupling capacitors.
- Use capacitors that have good quality at high frequency for low ESR, such as tantalum or ceramic capacitors. Do not use electrolytic capacitors.

1.5.2.5 EMI Noise Suppression

- A common-mode choke coil may suppress EMI noise effectively, although such a coil could affect USB 2.0 signal quality.
- Choose a good quality noise filter, if necessary.
- For a typical implementation, a choke is not required.
- Use good quality, shielded cables.

1.6 Terminology

Byte. 8-bit quantity of data.

Word. 16-bit quantity of data.

Scalar. Multi-byte data element.

Local Transaction. A read or write operation on the local bus. It 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.

Clock cycle. One period of the internal 60 MHz clock.

Big Endian. The most significant byte in a scalar is located at address 0.

Little Endian. The least significant byte in a scalar is located at address 0.

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

2 Pin Description

Abbreviation	Description
I	Input
O	Output
I/O	Bi-directional
S	Schmitt Trigger
TS	Tri-State
TP	Totem-Pole
OD	Open-Drain
PD	50K Pull-Down
PU	50K pull-up
#	Active low

NOTE: Input pins that do not have an internal pull-up or pull-down resistor (designated by PU or PD in the 'Type' column below) must be driven externally when the NET2272 is in the suspended state.

2.1 Digital Power & Ground (10 pins)

Signal Name	Pin	Type	Description
VDDC	1, 48	Power	**Digital Core Supply Voltage**. Connect to 2.5V.
VSSC	24, 56	GND	**Digital Core Ground**. Connect to GND.
VDDIO	27, 42, 55	Power	**I/O Interface Supply Voltage**. Connect to 3.3V.
VSSIO	33, 41, 54	GND	**I/O Interface Ground**. Connect to GND.

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

2.2 USB Transceiver (15 pins)

Signal Name	Pin	Type	Description
DP	6	I/O	**High Speed USB Positive Data Port.** DP is the high speed positive differential data signal of the USB data port. It also acts as the full speed positive differential input data port. This pin connects directly to the USB connector.
DM	8	I/O	**High Speed USB Negative Data Port.** DM is the high speed negative differential data signal of the USB data port. It also acts as the full speed negative differential input data port. This pin connects directly to the USB connector.
RSDP	5	O	**Full Speed USB Positive Output Data Port.** RSDP is the full speed positive differential output data signal of the USB data port. This pin connects through a 39.2 ohm +/- 1% resistor to the USB connector.
RSDM	9	O	**Full Speed USB Negative Output Data Port.** RSDM is the full speed negative differential output data signal of the USB data port. This pin connects through a 39.2 ohm +/- 1% resistor to the USB connector.
RPU	2	O	**DP Pull Up Resistor.** Connect to DP pin through a 1.5K resistor.
RREF	13	I	**Reference Resistor.** Connect 2.43K +/- 1% resistor to analog ground. The typical voltage on this pin is 1.27 volts.
VDD25	3	Power	**Supply Voltage**. Connect to digital 2.5 V.
VDD33	7	Power	**Supply Voltage**. Connect to digital 3.3 V.
PVDD	11	Power	**PLL Supply Voltage.** Connect to analog 2.5 V.
AVDD	15	Power	**Analog Supply Voltage**. Connect to analog 2.5 V.
GND	4,10	Ground	**Digital Ground.** Connect to ground.
AVSS	12, 14	Ground	**Analog Ground.** Connect to analog ground.
COM (AVSS)	16	Ground	**PLL Ground.** Connect to analog ground.

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

2.3 Clocks, Reset, Misc. (8 pins)

Signal Name	Pin	Type	Description
XIN	25	I	**Oscillator input.** Connect to 30 MHz crystal or external oscillator module.
XOUT	26	O	**Oscillator output.** Connect to crystal, or leave open if using an external oscillator module. The oscillator stops when the device is suspended.
LCLKO	57	O, 12mA, TS	**Local Clock Output.** This pin is a buffered clock output from the internal PLL, with the frequency depending on the state of the *Local Clock Output* field in the **LOCCTL** configuration register. This pin stops oscillating as soon as the NET2272 is put into suspend mode, and is not driven while the device is suspended. When the internal oscillator is started, LCLKO is prevented from being driven for 2 msec. LCLKO doesn't oscillate while the chip is in the power-down mode.
RESET#	58	I, S	**Reset.** External reset. Connect to local or power-on reset. To reset when oscillator is stopped (initial power-up or in suspend state), assert for at least two milliseconds. When oscillator is running, assert for at least five 60 MHz clock periods.
VBUS	64	I, S	**USB VBUS.** This input indicates when the NET2272 is connected to a powered-up USB host connector. An external 47K pull-down resistor should be connected to this pin to keep it low when not connected to the USB.
TEST	18	I	**Test input**. Connect to ground for normal operation.
TRST	40	I	**TRST Test input.** TAP controller reset. Connect to ground for normal operation.
TMC2	17	I	**TMC2 Test input**. I/O buffer control. Connect to ground for normal operation.

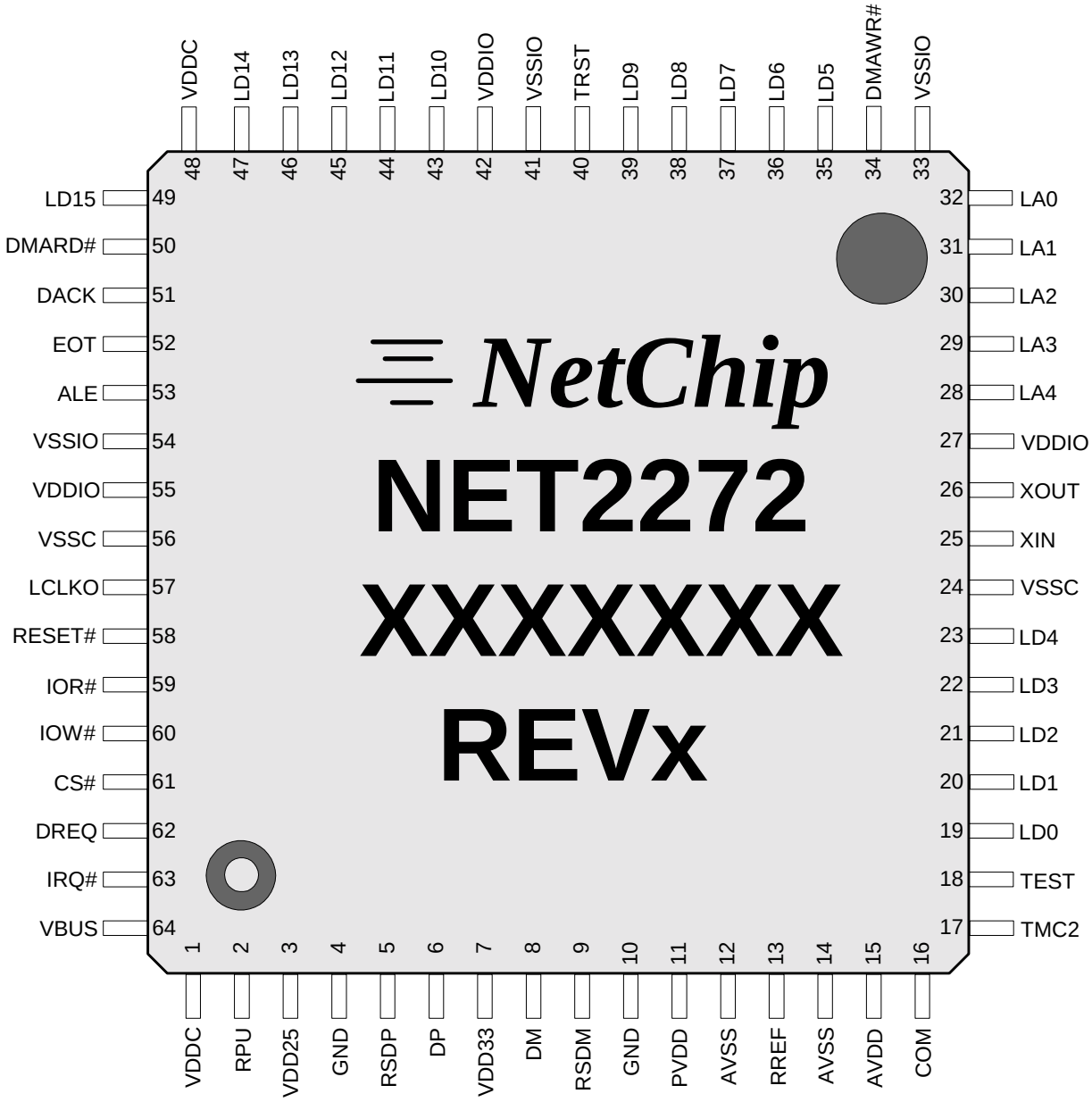
335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

2.4 Local Bus Pin Descriptions (31 pins)

Signal Name	Pin	Type	Description
LA[4:0]	28-32	I	**Address Bus.** Five bits of address can directly address most of the internal registers of the NET2272. A minimum of 1 address bit is required to operate the NET2272, using an optional Register-Indirect mode. A third addressing mode (multiplexed address and data) uses ALE with LD[4:0] to provide 5 bits of register addressing.
ALE	53	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 will automatically detect use of the ALE pin to indicate use of multiplexed address and data on the LD[4:0] pins.
LD[15:0]	49, 47-43, 39-35, 23-19	I/O, 6mA	**Data Bus.** These pins serve as the input and output data bus for the local CPU. In multiplexed address/data mode, ALE can be used with LD[4:0] to provide 5-bits of address on the falling edge of ALE. In 16-bit mode, the data bus is 16-bits wide.
IOR#	59	I	**Read Strobe.** The local bus master asserts this signal during a read transaction.
IOW#	60	I	**Write Strobe.** The local bus master asserts this signal during a write transaction.
IRQ#	63	O, 12mA, OD	**Interrupt Request Output.** This signal interrupts the local processor based on events selected in internal program registers. Since this pin is open-drain, an external 1K pull-up resistor is required.
CS#	61	I	**Chip Select.** This signal enables access to registers within the NET2272. Asserting this pin during suspend will cause the device to wake up. Asserting this pin during RESET# holds the NET2272 in a low-power mode by disabling the internal oscillator.
DMAWR#	34	I	**DMA Write Strobe.** The DMA bus master asserts this signal during a DMA write transaction when split-bus DMA is selected.
DMARD#	50	I	**DMA Read Strobe.** The DMA bus master asserts this signal during a DMA read transaction when split-bus DMA is selected
DREQ	62	O, 3mA, TS	**DMA Request.** This signal requests DMA transfers from an external DMA controller. This output floats when the USB Host suspends the device. The polarity of this signal is programmable.
DACK	51	I	**DMA Acknowledge.** Used to transfer data to/from the packet buffer in response to DREQ. This pin is ignored unless the *DMA DACK Enable* bit in the **LOCCTL1** register is set. The polarity of this signal is programmable.
EOT	52	I	**End of Transfer.** This signal from an external DMA controller is used to terminate a DMA transfer. The current word will be transferred, but no additional transfers will be requested. EOT can be programmed to cause an interrupt. The polarity of this signal is programmable.

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

2.5 Physical Pin Assignment



Note: This drawing is for informational purposes only. Please contact NetChip for additional chip marking, PCB layout and manufacturing information.

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

3 Reset and Initialization

3.1 Overview

The NET2272 normal initialization sequence consists of the following:

- Assert/De-Assert RESET# pin
- Local CPU initializes USB and local bus configuration registers

3.2 RESET# Pin

The RESET# pin causes all logic in the NET2272 to be set to its default state. It 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 microseconds, it is interpreted as a root port reset. The root port reset is only recognized if the VBUS input pin is high, and the *USB Detect Enable* bit in the **USBCTL0** register is set. The following resources are reset:

- Serial Interface Engine (SIE)
- USB state machines
- Local state machines
- **OURADDR** Register
- Buffer pointers

The root port reset does not affect the remainder of the configuration registers. The *Root Port Reset Interrupt* bit is set when a change in the root port reset has been detected. The local CPU should take appropriate action when this interrupt occurs.

According to the USB Specification, the width of the USB reset is minimally 10ms and may be longer depending on the upstream host or hub. There is no specified maximum width for the USB reset.

3.4 Reset Summary

The following table shows which device resources are reset when each of the 2 reset sources are asserted.

Device Resources / Reset Sources	USB, SIE modules, **OURADDR,** registers	All Configuration Registers	Endpoint Buffer Pointers
RESET# pin	X	X	X
USB Root Port Reset	X		X

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

4 Local Bus Interface

4.1 Introduction

The Local Bus Interface allows the NET2272 to be easily interfaced with many generic processors and custom ASIC interfaces. Both multiplexed and non-multiplexed buses are supported.

4.2 Register Addressing Modes

Several addressing methods are provided in the NET2272 in order to support various user architectures. These modes are always active and nothing special needs to be done to use them. These modes act on a transaction-by-transaction basis, allowing DMA and CPU to operate with inherently different bus architectures. For instance, the CPU could operate with a multiplexed bus (using ALE to de-multiplex the Address/Data bus) while the DMA could operate using a non-multiplexed Data bus.

4.2.1 Direct Address Mode

Direct Address Mode uses LA[4:0] to directly access the first 32 configuration registers.

4.2.2 Indirect Address Mode

Indirect Address mode uses registers **REGADDRPTR** (address 0x00) and **REGDATA** (address 0x01) to provide a Command/Data interface to the NET2272 internal registers and buffers. All CPU transactions performed with **REGDATA** will have their address sourced by **REGADDRPTR**. The local CPU first programs **REGADDRPTR** with the desired register address, then reads or writes to **REGDATA** with the data intended for the register pointed to by **REGADDRPTR**. This method of register addressing requires only 1 physical address bit (to access address 0x00 or address 0x01). All unused address bits of the NET2272 should be connected to ground. When all five address bits are being used, this addressing method allows access to registers above address 1Fh.

4.2.3 Multiplexed Address Mode

Multiplexed Address mode uses the ALE pin to de-multiplex the desired address from the data bus. The NET2272 automatically detects the use of ALE and will use the address represented by LD[4:0] on the falling edge of ALE as the address of the current transaction. This addressing mode is supported by several common microcontrollers. ALE should be grounded if this mode is not used.

4.3 Control Signal Definitions

The 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 / Byte Alignment

The local bus supports 8 or 16-bit buses. In 8-bit mode, all configuration registers and the buffers are accessed one byte at a time. A typical 8-bit application would connect the CPU address bits A[4:0] to the NET2272 address bus LA[4:0].

In 16-bit mode, the configuration registers are still accessed a byte at a time, but the buffers are accessed a word at a time. The *Byte Swap* bit in the **LOCCTL** configuration register determines whether the bytes are swapped as they are being written into the buffer in 16-bit mode. This allows for connections to little or big endian processors. A typical 16-bit application would connect the CPU address bits A[5:1] to the NET2272 address bus LA[4:0].

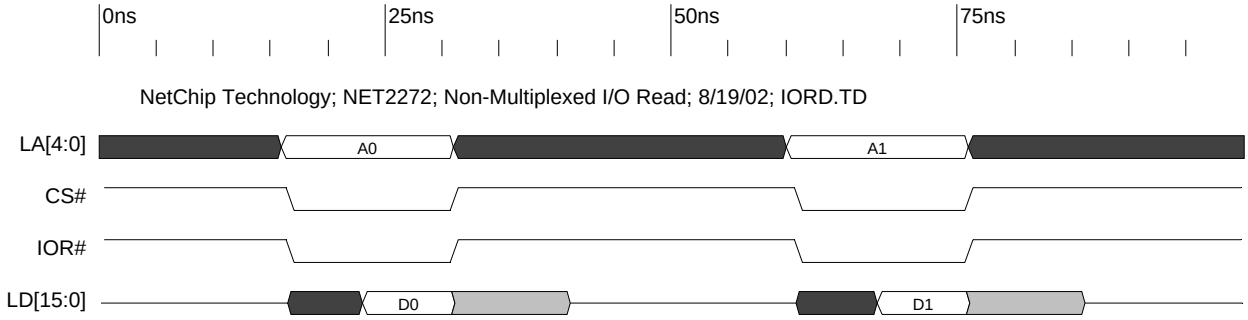
335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

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.

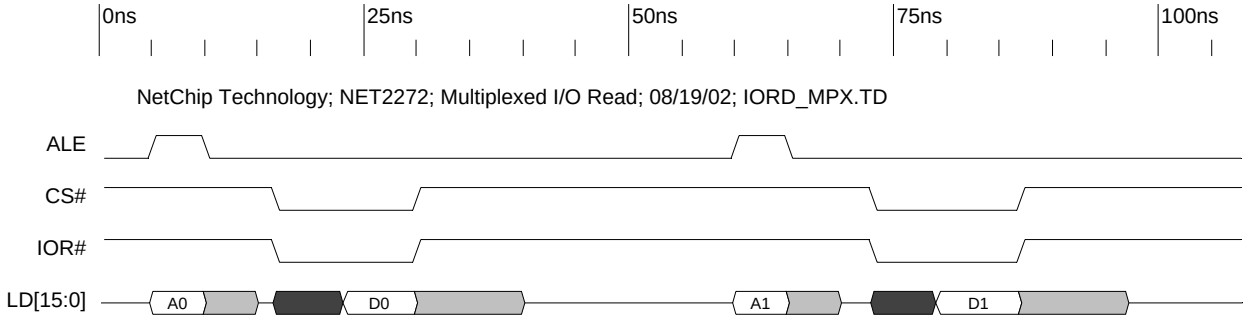
4.5.1 Non-Multiplexed I/O Read

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 either CS# or IOR# are de-asserted. A new I/O read transaction cannot be started until the T8A recovery time has expired, and a new I/O write transaction cannot be started until the T8B recovery time has expired. ALE must be held low for non-multiplexed mode.



4.5.2 Multiplexed I/O Read

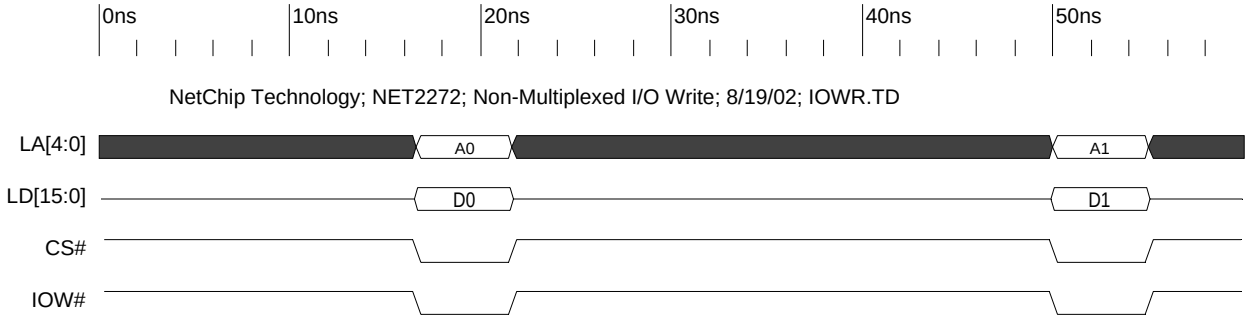
Multiplexed I/O reads are started when the address is driven onto the lower bits of the data bus, and ALE is pulsed. Once the address has been latched into the NET2272, the data phase is initiated with the assertion of both CS# and IOR#. To prevent bus contention on the data bus, CS# and IOR# should not be asserted until the local bus master has tri-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 either CS# or IOR# are de-asserted. A new I/O read transaction cannot be started until the T8A recovery time, has expired, and a new I/O write transaction cannot be started until the T8B recovery time has expired.



335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

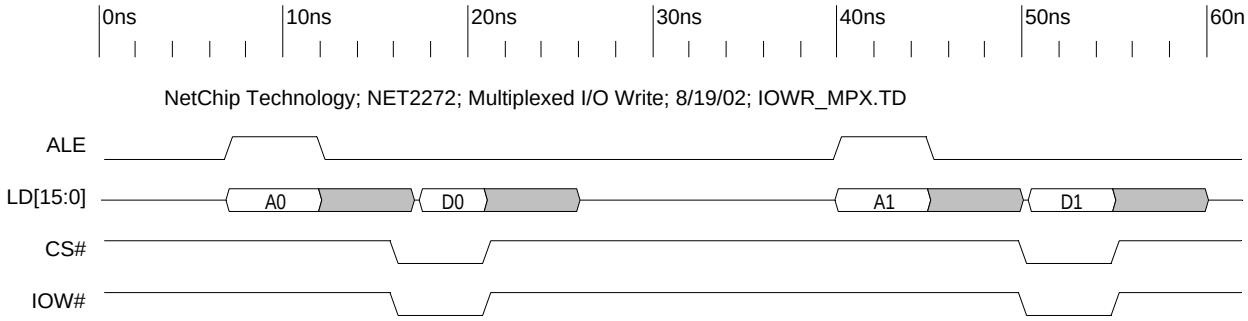
4.5.3 Non-Multiplexed I/O Write

Non-multiplexed I/O writes are started when both CS# and IOW# are asserted. The address and write data must meet the setup time with respect to the end of the write transaction. Data is written into the register or packet buffer when either CS# or IOW# are negated. A new I/O write transaction cannot be started until the T15A recovery time has expired, and a new I/O read transaction cannot be started until the T15B recovery time has expired. ALE must be held low for non-multiplexed mode.



4.5.4 Multiplexed I/O Write

Multiplexed I/O writes are started when the address is driven onto the lower bits of the data bus, and ALE is pulsed. Once the address has been latched into the NET2272, the data phase is initiated with the assertion of both CS# and IOW#. The write data must meet the setup time with respect to the end of the write cycle. Data is written into the register or packet buffer when either CS# or IOW# are negated. A new I/O write transaction cannot be started until the T15A recovery time has expired, and a new I/O read transaction cannot be started until the T15B recovery time has expired.



335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

4.5.5 I/O Performance

4.5.5.1 Non-Multiplexed Read Transaction

- Read Access Time (T6): 18 nsec
- Read Recovery Time (T8): 19 nsec
- For an 8-bit bus, the maximum performance is:
  1/37 nsec = 27 Mbytes/sec
- For a 16-bit bus, the maximum performance is:
  2/37 nsec = 54 Mbytes/sec

4.5.5.2 Multiplexed Read Transaction

- ALE Width (T3): 5 nsec
- ALE to Read Command (T19): 1 nsec minimum
- Read Access Time (T6): 18 nsec
- Read Recovery Time (T8): 19 nsec
- For an 8-bit bus, the maximum performance is:
  1/43 nsec = 23 Mbytes/sec
- For a 16-bit bus, the maximum performance is:
  2/43 nsec = 46 Mbytes/sec

4.5.5.3 Non-Multiplexed Write Transaction

- Write Width (T12): 5 nsec
- Write to Write Recovery Time (T15A): 28 nsec
- For an 8-bit bus, the maximum performance is:
  1/33 nsec = 30 Mbytes/sec
- For a 16-bit bus, the maximum performance is:
  2/33 nsec = 60 Mbytes/sec

4.5.5.4 Multiplexed Write Transaction

- ALE Width (T3): 5 nsec
- ALE to Write Command (T19): 1 nsec minimum
- Write Width (T12): 5 nsec
- Write to Write Recovery Time (T15A): 28 nsec
- For an 8-bit bus, the maximum performance is:
  1/39 nsec = 25 Mbytes/sec
- For a 16-bit bus, the maximum performance is:
  2/39 nsec = 50 Mbytes/sec

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

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 for endpoint A or endpoint B only. The local CPU handles transfers to/from endpoints 0 and C. The external DMA controller is programmed to perform fly-by demand mode transfers. In this mode, transfers occur only when the NET2272 requests them and 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 *DMA Endpoint Select* field of the **DMAREQ** register. During CPU transactions, the endpoint buffer is determined by the *Page Select* field of the **PAGESEL** register. In split DMA mode, CPU accesses to one endpoint buffer can occur simultaneously with DMA accesses to another endpoint buffer.

4.6.1 DMA Read

For OUT transfers (host to device), 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 desired number of bytes. The signals DREQ, DACK, IOR#, and EOT are used to control the transactions between the external DMA controller and the NET2272. DREQ and DACK are minimally needed to exchange data with the NET2272 since the direction (read) is established by the *Endpoint Direction* bit in the **EP_CFG** register. The mode of operation is set by the *DMA Control DACK* bit in the **DMAREQ** register. If the *DMA Control DACK* bit is high, then both DACK and IOR# are needed by the NET2272 for a DMA read. If the *DMA Control DACK* bit is low, then only DACK is needed.

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

When the NET2272 has data available in an endpoint buffer, and that endpoint has been assigned to the DMA channel, the DMA request (DREQ) signal is asserted. The external DMA controller then requests the local bus from the local bus master. After the external DMA controller has been granted the bus, it drives a valid memory address and asserts DACK, IOR# (optional), and MEMW# (to memory), thus transferring a byte from an endpoint's buffer to local memory. In DMA slow mode, the NET2272 de-asserts DREQ within T20 after the start of the transaction, while in fast mode, it de-asserts DREQ at the start of the transaction. If there is still data in the buffer, the NET2272 then re-asserts DREQ. The DMA transfers continue until the DMA byte count reaches zero or the EOT pin is asserted during the last DMA transfer. The *DMA Done Interrupt* bit in the **IRQSTAT0** register will be set for the following conditions:

- The EOT pin is asserted during the last DMA transfer.
- The local CPU writes a zero to the **EP_TRANSFER** register after the DMA has finished.

If DMA Burst Mode is selected, DREQ is asserted when there is data in the buffer and the DMA is enabled. It remains asserted until the FIFO becomes empty, the DMA is disabled, or the EOT pin is asserted.

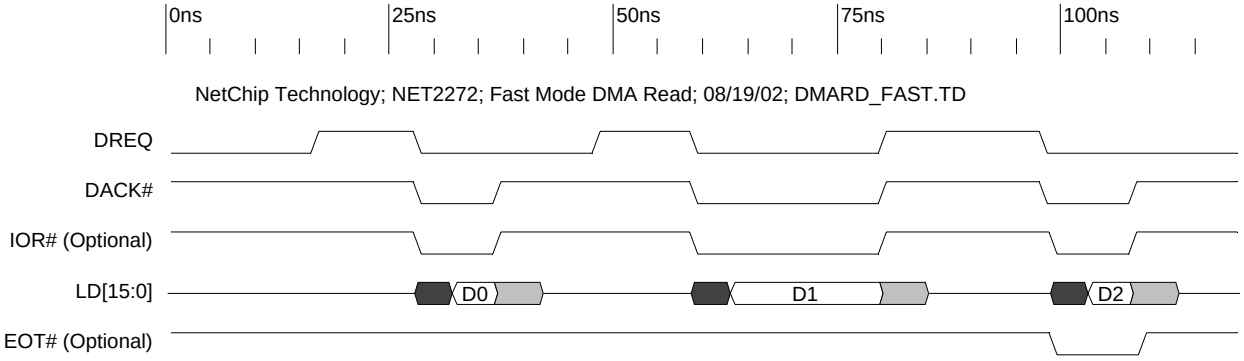
335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

4.6.1.1 Slow DMA Read Timing

In this mode, DREQ is de-asserted within T20 after the beginning of the read transaction. It is then re-asserted within T21 after the end of the read transaction.

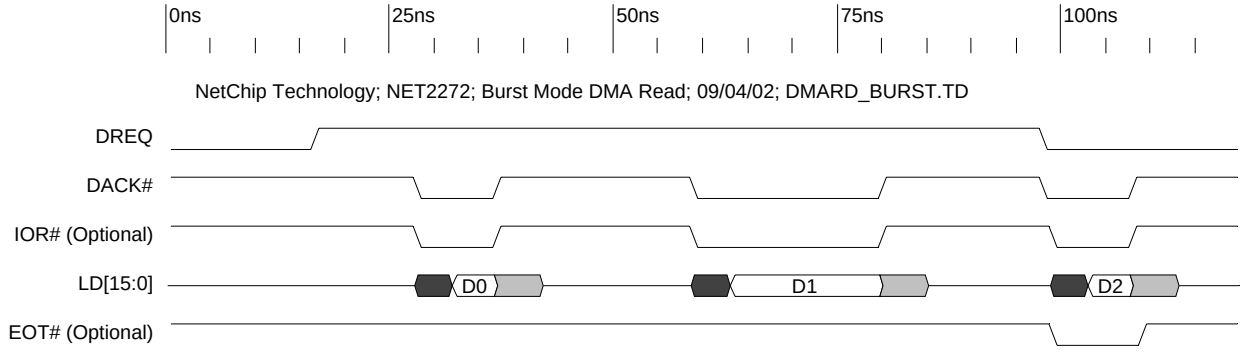
4.6.1.2 Fast DMA Read Timing

In this mode, DREQ is de-asserted at the beginning of the read transaction. It is then re-asserted either at the end of the read transaction if the read enable width is greater than T21, or at T21 after the beginning of the read transaction if the read enable width is less than T21.



4.6.1.3 Burst DMA Read Timing

In this mode, DREQ remains asserted until the DMA transfer completes.



335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

4.6.2 DMA Write

For IN transfers (device 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 desired number of bytes. The NET2272 **EP_TRANSFER** register is also programmed with the desired transfer size (in bytes). The transfer size programmed into the **EP_TRANSFER** register can span many packets, allowing a single DMA setup to transfer multiple packets. The DMA Request signal (DREQ) is asserted anytime there is space available in the buffer. The endpoint's **Max Packet Size** register controls the maximum number of bytes transmitted to the host in a single packet. A short packet will be sent if there are remaining bytes to be sent when all **EP_TRANSFER** bytes have been written or when EOT has been asserted during the last DMA cycle.

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

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

As long as there is space available in the selected endpoint's buffer, and there are still bytes to be transferred, the NET2272 will request DMA transfers by asserting DREQ. The external DMA controller then requests the local bus from the local CPU. After the DMA controller has been granted the bus, it drives DACK, IOW# (optional), and MEMR# (to memory) to transfer a byte from memory to the endpoint's buffer. For DMA slow mode, the NET2272 de-asserts DREQ within T20 after the start of the transaction while for DMA fast mode, the NET2272 de-asserts DREQ at the beginning of the transaction. If there is still space in the buffer and there are more bytes to be transferred, the NET2272 re-asserts DREQ.

The USB host sends an IN token to the NET2272 and starts an IN data transaction from the selected endpoint's buffer. The DMA transfers continue until **EP_TRANSFER** bytes have been transferred or EOT has been asserted. The *DMA Done Interrupt* bit in the **IRQSTAT0** register will be set for the following conditions:

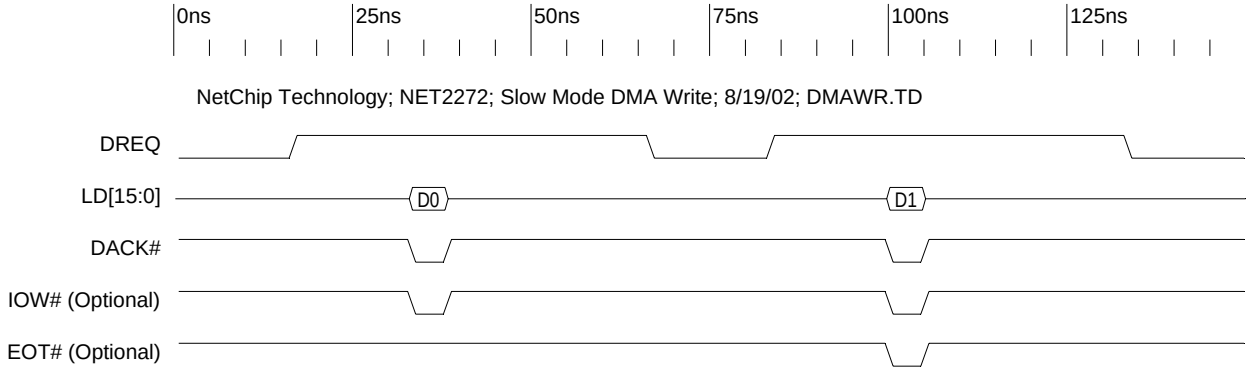
- The EOT pin is asserted during the last DMA transfer.
- The **EP_TRANSFER** counter counts down to 0.
- The local CPU writes a zero to the **EP_TRANSFER** register after the DMA has finished.

If DMA Burst Mode is selected, DREQ is asserted when there is space in the buffer and the DMA is enabled. It remains asserted until the FIFO becomes full, the DMA is disabled, the EP_TRANSFER counter counts down to 0, or the EOT pin is asserted.

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

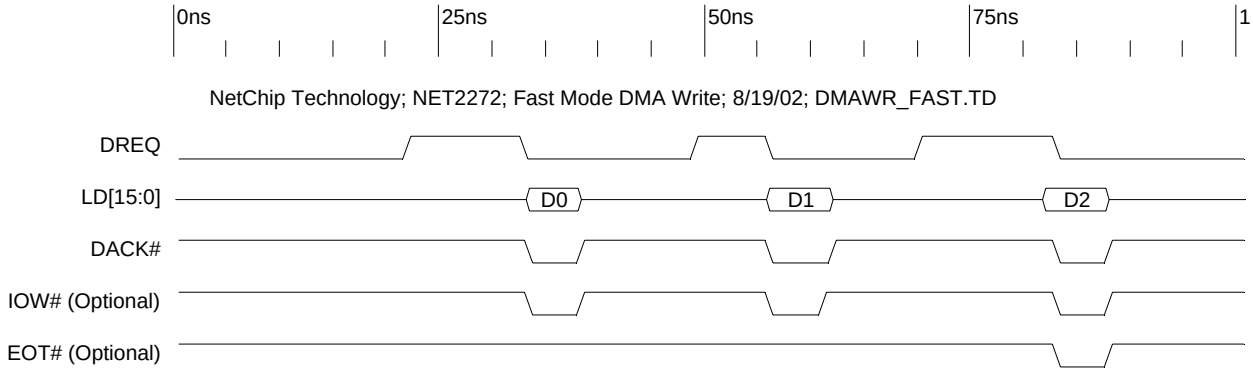
4.6.2.1 Slow DMA Write Timing

In this mode, DREQ is de-asserted within T20 after the beginning of the write transaction. It is then re-asserted within T21 after the end of the write transaction.



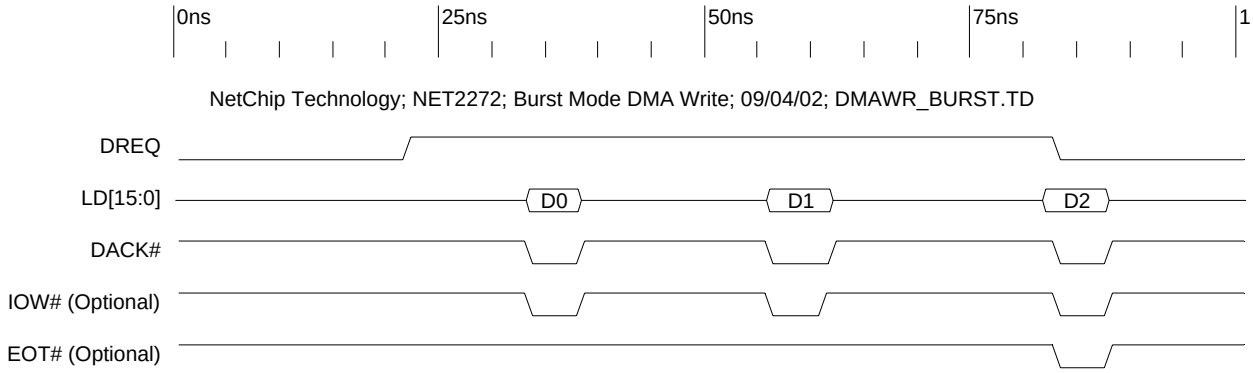
4.6.2.2 Fast DMA Write Timing

In this mode, DREQ is de-asserted at the beginning of the write transaction. It is then re-asserted within T21 after the end of the write transaction.



4.6.2.3 Burst DMA Write Timing

In this mode, DREQ remains asserted until the DMA transfer completes.



335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

4.6.3 DMA Split Bus Mode

In this 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 *DMA Split Bus Mode* bit of the **LOCCTL** register enables DMA split Bus Mode.

The split mode DMA transactions are the same as normal DMA transactions, except that DMARD# is used instead of IOR#, and DMAWR# is used instead of IOW#. While DMA transactions are taking place 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.

4.6.4 Terminating DMA Transfers

The EOT signal is used to halt a DMA transfer, and is typically provided by an external DMA controller. It should be asserted 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 DMA has terminated, the USB transfer (in the case of an IN transaction) is not complete until the last byte has been transferred from the endpoint's buffer to the USB. The EOT input resets the NET2272 *DMA Request En*able bit of the **DMAREQ** register. When EOT is detected, the current endpoint buffer is automatically validated, causing any remaining data in the current packet to be sent 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 will be returned in response to the next IN token. The *DMA Request Enable* bit is also automatically cleared when the **EP_TRANSFER** counter reaches zero for IN endpoints or when the local CPU writes a 0 to the **EP_TRANSFER** counter.

If the external DMA controller doesn't 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 (only possible if using DMA split mode), the current cycle will complete before DMA requests are terminated. The endpoint buffer is not automatically validated when the *DMA Request Enable* is cleared. In this case, the CPU needs to explicitly validate the packet by writing a zero to the **EP_TRANSFER** register.

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

4.6.5 DMA Performance

4.6.5.1 DMA Read; Slow Mode

- DREQ to DACK: 5 nsec (depends on DMA controller)
- DACK asserted to DREQ de-asserted (T20): 50 nsec
- DREQ de-asserted to DREQ asserted (T25): 25 nsec
- For an 8-bit bus, the maximum performance is:
  1/80 nsec = 12.5 Mbytes/sec
- For a 16-bit bus, the maximum performance is:
  2/80 nsec = 25 Mbytes/sec
- The actual throughput will be reduced if the DMA controller DREQ to DACK delay is longer than 5 nsec.

4.6.5.2 DMA Read; Fast Mode

- DREQ to DACK: 5 nsec (depends on DMA controller)
- Read Width (T22): 16 nsec
- DACK de-asserted to DREQ asserted (T21): 35 nsec
- For an 8-bit bus, the maximum performance is:
  1/56 nsec = 17.8 Mbytes/sec
- For a 16-bit bus, the maximum performance is:
  2/56 nsec = 35.7 Mbytes/sec
- The actual throughput will be reduced if the DMA controller DREQ to DACK delay is longer than 5 nsec.

4.6.5.3 DMA Read; Burst Mode

- DMA Read Cycle time (T17): 35 nsec
- For an 8-bit bus, the maximum performance is:
  1/35 nsec = 28 Mbytes/sec
- For a 16-bit bus, the maximum performance is:
  2/35 nsec = 57 Mbytes/sec

4.6.5.4 DMA Write; Slow Mode

- DREQ to DACK: 5 nsec (depends on DMA controller)
- DACK asserted to DREQ de-asserted (T20): 50 nsec
- DREQ de-asserted to DREQ asserted (T25): 25 nsec
- For an 8-bit bus, the maximum performance is:
  1/80 nsec = 12.5 Mbytes/sec
- For a 16-bit bus, the maximum performance is:
  2/80 nsec = 25 Mbytes/sec
- The actual throughput will be reduced if the DMA controller DREQ to DACK delay is longer than 5 nsec.

4.6.5.5 DMA Write; Fast Mode

- DREQ to DACK: 5 nsec (depends on DMA controller)
- Write Width (T26): 5 nsec
- DACK de-asserted to DREQ asserted (T21): 45 nsec
- For an 8-bit bus, the maximum performance is:
  1/55 nsec = 18 Mbytes/sec
- For a 16-bit bus, the maximum performance is:
  2/55 nsec = 36 Mbytes/sec
- The actual throughput will be reduced if the DMA controller DREQ to DACK delay is longer than 5 nsec.

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

4.6.5.6 DMA Write; Burst Mode

- Write Width (T26): 5 nsec
- Write recovery (T30): 28 nsec
- For an 8-bit bus, the maximum performance is:
  1/33 nsec = 30 Mbytes/sec
- For a 16-bit bus, the maximum performance is:
  2/33 nsec = 60 Mbytes/sec

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

5 USB Functional Description

5.1 USB Interface

The NET2272 is a USB function device, and as a result is always a slave to the USB host. The bit and packet level protocols, as well as the electrical interface of the NET2272, conform to USB Specification Version 2.0. The USB host initiates all USB data transfers to and from the NET2272 USB port. The NET2272 can be configured for up to 3 physical and 30 virtual endpoints, in addition to Endpoint 0. Each endpoint can be an isochronous, bulk or interrupt type. The configuration registers are used to program the characteristics of each endpoint. The NET2272 operates in either Full speed (12 Mbps) or High speed (480 Mbps) modes.

5.2 USB Protocol

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

5.2.1 Tokens

Tokens are a type of Packet Identifier (PID), and follow the sync byte at the beginning of a token packet. The four classic types of tokens 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. Each packet begins with a sync field and a Packet Identifier (PID). The other fields vary depending on the type of packet.

An SOF packet consists of the following fields:

- Sync byte (8-bits)
- Packet Identifier (8-bits)
- Frame Number (11-bits)
- CRC (5-bits)

A token packet consists of the following fields:

- Sync byte (8-bits)
- Packet Identifier (8-bits)
- Address (7-bits)
- Endpoint (4-bits)
- CRC (5-bits)

A data packet consists of the following fields: a token packet always precedes Data packets.

- Sync byte (8-bits)
- Packet Identifier (8-bits)
- Data (n bytes)
- CRC (16-bits)

A handshake packet consists of the following fields:

- Sync byte (8-bits)
- Packet Identifier (8-bits)

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

5.2.3 Transaction

A transaction consists of a token packet, optional data packet(s), and a 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 handshake (status) transaction.

5.3 Automatic Retries

5.3.1 Out Transactions

If an error occurs during an OUT transaction, the NET2272 reloads its local side buffer read pointer back to the beginning of the failed packet. The host then sends another OUT token and re-transmits the packet. Once the packet has been successfully received by the NET2272, the Packet Received interrupt is set. The NET2272 can handle any number of back-to-back retries, but the host determines how many times a packet is retried.

5.3.2 In Transactions

If an error occurs during an IN transaction, the NET2272 reloads its USB side buffer read pointer back to the beginning of the failed packet. The host then sends another IN token and the NET2272 re-transmits the packet. Once the host has successfully received the packet, the Packet Transmitted interrupt is set.

5.4 Ping Flow Control

When operating in high speed mode, the NET2272 supports the PING protocol for bulk OUT and control endpoints. This protocol allows the NET2272 to indicate to the host that it can't accept an OUT data packet. The host then sends PING tokens to query the NET2272. Once the NET2272 can accept a maximum size packet, it returns an ACK in response to the PING. Now the host sends an OUT token and data packet. The NET2272 returns an ACK handshake if the packet is accepted, and there is space to receive an additional packet. If it can accept the current packet, but no others, it returns a NYET handshake to the host. The host then starts sending PING tokens again.

5.5 Packet Sizes

The maximum packet size of an endpoint is determined by the corresponding **EP_MAXPKT** register. For IN transactions, the NET2272 will return a maximum size packet to the host if there are at least ‘max packet’ bytes in the buffer. A packet of size less than the maximum is returned to the host in response to an IN token if the data in the buffer has been explicitly validated.

The following table shows the allowable maximum packet sizes:

Type of Endpoint	Low Speed Mode (Not Supported)	Full Speed Mode	High Speed Mode
Control	8	8, 16, 32, 64	64
Bulk	N/A	8, 16, 32, 64	512
Interrupt	8	64 max	1024 max
Isochronous	N/A	1023 max	1024 max

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

5.6 USB Endpoints

The NET2272 supports Control, Isochronous, Bulk, and Interrupt endpoints. All endpoints are unidirectional except for Control endpoints. Bi-directional bulk, isochronous, or 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 gain information about the device, its configurations, interfaces and other endpoints. Control endpoints are bi-directional, and data delivery is guaranteed.

The host sends 8-byte setup packets to Endpoint 0 to which the device interprets and responds. The NET2272 has 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 bus to the local bus. For Control reads, data flows through the packet buffer from the local bus to the USB bus.

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 any data returned to the host, including status and descriptors. Refer to Chapter 9, Standard Device Requests, for a description of the data that must be returned for each USB request. The host will reject descriptors that have unexpected values in any of the fields.

5.6.1.1 Control Write Transfer

A successful control write transfer to Control Endpoint 0 consists of the following:

Transaction	Stage	Packet Contents	# 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 8 bytes have been received. A *Setup Packet Interrupt* bit is set to notify the local CPU that a setup packet has been received. The 8-byte data packet is then read and interpreted by the local CPU. A Setup transaction cannot be stalled or NAKed, but if the data is corrupted, then the NET2272 will not return an ACK to the host.

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

- Interrupt bits are set and can interrupt the local CPU
- The local CPU reads the buffer
- The NET2272 returns an ACK if no error has occurred.

For a successful Status transaction, the NET2272 returns a zero length data packet. A NAK or STALL handshake can be returned if an error occurred.

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

5.6.2 Control Write Transfer Details

For control write transfers, the host first sends 8 bytes of setup information. The setup bytes are stored into an 8-byte register bank that can be accessed by the local CPU. After the eight bytes have been 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 transactions. The number of bytes to be transferred in the Data transactions is specified in the setup packet. When the setup packet is received, the *Control Status Phase Handshake* bit is automatically set in anticipation of the control status phase. While this bit is set, the control status phase will be acknowledged with a NAK, allowing the local CPU to prepare its handshake response (ACK or STALL). Once the *Control Status Phase Handshake* bit is cleared and the OUT buffer is empty, the ACK or STALL handshake will be returned to the host. Waiting for the OUT FIFO to become empty prevents another Control Write from corrupting the current packet data in the FIFO.

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 packet buffer. If the buffer fills up and another byte is transferred from the host, the NET2272 will return a NAK handshake to the host, signaling that the data could not be accepted.

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

If the local CPU has stalled this endpoint by setting the *Endpoint Halt* bit, the NET2272 will not store any data into the buffer, and will respond with a STALL acknowledge to the host. There will not be a Status transaction in this case.

The local CPU can either poll the *Data Packet Received Interrupt* bit, or enable it as an interrupt, and then read the packet from the buffer when the interrupt occurs. If the host tries to write more data than was indicated in the setup packet, then the local CPU should set the *Endpoint Halt* bit for Endpoint 0. In this case there will not be a status stage from the host.

After all of the optional Data transaction packets have been received, the host will send an IN token, signifying the Status transaction. The *Control Status Interrupt* bit is set after the IN token of the Status transaction has been received. Until the *Control Status Phase Handshake* bit is cleared by the local CPU and the OUT buffer is empty, the NET2272 will respond to the Status transaction with NAKs, indicating that the device is still processing the setup command. When the *Control Status Phase Handshake* bit has been cleared by the local CPU and the firmware has emptied the data form the OUT buffer, the NET2272 will respond with a zero length data packet (transfer OK) or STALL (device had an error).

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

5.6.2.1 Control Read Transfer

A successful control read transfer from Control Endpoint 0 consists of the following:

Transaction	Stage	Packet Contents	# 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 in the same way as for control write transfers.

During the Data transaction, zero, one or more data packets are read from the Endpoint 0 buffer. For each packet:

- Interrupt bits are set and can interrupt the local CPU
- The local CPU writes data to the buffer
- If there is no data in the buffer, a NAK or zero length packet is returned to the host
- The Host returns an ACK to the NET2272 if no error has occurred.

For a successful Status transaction, the Host sends a zero length data packet, and the NET2272 responds with an ACK. A NAK or STALL can be returned if an error occurred.

5.6.2.2 Control Read Transfer Details

For control read transfers, the host first sends 8 bytes of setup information. The setup bytes are stored into an 8-byte register bank that can be accessed from the local CPU. After the eight bytes have been 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 transactions. The number of bytes to be transferred in the Data transactions is specified in the setup packet. When the setup packet is received, the *Control Status Phase Handshake* bit is automatically set. While this bit is set, the control status phase will be acknowledged with a NAK, allowing the local CPU to prepare its handshake response (ACK or STALL). Once the *Control Status Phase Handshake* bit is cleared, the ACK or STALL handshake will be 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, it is returned to the host. If Endpoint 0 has no data to return, it returns either a zero length packet (signaling that there is no more data available) or a NAK handshake (the data is not available yet).

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

Packet Validated	Amount of Data in buffer	Action
0	< Max Packet Size	NAK to host
X	>= Max Packet Size	Return data to host
1	empty	Zero length packet to host
1	>0	Return data to host

After each packet has been sent 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 will not be set, and the same packet is sent 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 was requested in the setup packet, the local CPU should set the STALL bit for the endpoint.

After all of the optional Data transaction packets have been transmitted, the host will send an OUT token, followed by a zero length data packet, signifying the Status transaction. The *Control Status Interrupt* bit is set after the OUT token of the Status transaction has been received. Until the *Control Status Phase Handshake* bit is cleared by the local CPU, the NET2272 will respond to the Status transaction with NAKs, indicating that the device is still processing the command specified by the Setup transaction. When the *Control Status Phase Handshake* bit has been cleared by the local CPU, the NET2272 will respond with an ACK (transfer OK) or STALL (Endpoint 0 is stalled).

5.6.3 Isochronous Endpoints

Isochronous endpoints are used for the transfer of time critical data. Isochronous transfers do not support any handshaking or error checking protocol, and are guaranteed a certain amount of bandwidth during each frame. The Serial Interface Engine in the NET2272 ignores CRC and bit stuffing errors during isochronous transfers, but sets the handshaking status bits in the **EP_STAT** registers the same as for non-isochronous packets so that the local CPU can detect errors. Isochronous endpoints are unidirectional, 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 1024 bytes.

For an Isochronous OUT endpoint, the local CPU or DMA can read data from the endpoint buffer after an entire packet has been received. If the endpoint buffer is the same size as the maximum packet size, then the ISO bandwidth must be set such that the buffer can be emptied before the next ISO packet arrives.

For an Isochronous IN endpoint, the local CPU or DMA 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).

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

5.6.3.1 Isochronous Out Transactions

Isochronous Out endpoints are used to transfer data from a USB host to the NET2272 local bus. An Isochronous OUT transaction consists of the following:

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 sending 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's buffer. Isochronous transactions are not retried, so if the buffer is full when a packet is transferred from the host (or the *NAK OUT Packets* bit is set), the packet is discarded and the *FIFO Overflow* status bit is set. No handshake packets are returned to the host, but the *USB OUT ACK Sent*, and *Timeout* status bits are still set to indicate the status of the transaction. If a CRC error is detected, the packet is accepted and the *Timeout* status bit is set. After every data packet is received, the local CPU should sample these status bits to determine if the NET2272 successfully received the packet.

By definition, isochronous endpoints do not utilize handshaking with the host. Since there is no way to return a stall handshake from an isochronous endpoint to the host, data that is sent to a stalled isochronous endpoint will be 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 a single-buffering, then the host should be programmed to allow the local CPU enough time to unload the buffer before the next packet is sent. If the endpoint is programmed for double-buffering, then the local side can unload one packet while the next one is being received.

5.6.3.2 High Bandwidth Isochronous OUT Transactions

The host sends high-bandwidth OUT PID sequences for each microframe depending on the *Additional Transaction Opportunities* field in the Endpoint Descriptor as follows:

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

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

High-Bandwidth OUT Transaction PID **field**	PID Received
00	DATA0
01	DATA1
10	DATA2
11	MDATA

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

5.6.3.3 Isochronous In Transactions

Isochronous IN endpoints are used to transfer data from the NET2272 local bus to a USB host. An isochronous IN transaction consists of the following:

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 sending 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's buffer, it 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 following table.

Packet Validated	Amount of Data in Buffer	Action
0	< Max Packet Size	Zero length packet to host; USB IN NAK Sent status bit set
X	>= Max Packet Size	Return data to host.
1	empty	Zero length packet to host
1	>0	Return data to host

After the packet has been sent 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, and the *FIFO Underflow* status bit is set. No handshake packets are returned to the host, but the *USB IN ACK Sent*, and *Timeout* status bits are still set to indicate the status of the transaction. After every data packet is transmitted, the local CPU should sample these status bits to determine if the packet was successfully transmitted to the host.

By definition, isochronous endpoints do not utilize handshaking with the host. Since there is no way to return a stall handshake from an isochronous endpoint to the host, data that is requested from a stalled isochronous endpoint will be transmitted normally.

5.6.3.4 High Bandwidth Isochronous IN Transactions

A USB device is required to send ISO PID sequences for each microframe according to the *Additional Transaction Opportunities* field in the Endpoint Descriptor and the **EP_n_HS_MAXPKT** register as follows:

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

When the first IN token of a microframe arrives, the NET2272 copies the *Additional Opportunities* field from the **EP_n_HS_MAXPKT** register to determine the initial PID. On each succeeding IN token of the microframe, the PID advances to the next token.

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

5.6.4 Bulk Endpoints

Bulk endpoints are used for guaranteed error-free delivery of large amounts of data between a host and device. Bulk endpoints are unidirectional, with the direction defined by the endpoint configuration registers.

5.6.4.1 Bulk Out Transactions

Bulk Out endpoints are used to transfer data from a USB host to the NET2272 local bus. A bulk OUT transaction to a Bulk Out endpoint consists of the following:

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 sending 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's buffer. If the buffer is full when another packet is transferred from the host, the packet will be discarded and the *USB OUT NAK Sent* status bit will be set. At the completion of the packet, a NAK handshake will be returned to the host, indicating that the packet could not be accepted.

All USB data passes through the endpoint's buffer to the local bus. The CPU waits until the *Data Packet Received Interrupt* occurs before reading the data from the buffer.

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

If the local CPU has stalled this endpoint by setting the *Endpoint Hal*t bit, the NET2272 will not store any data into the buffer, and will respond with a STALL handshake to the host.

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

5.6.4.2 Bulk In Endpoints

Bulk IN Endpoints are used to transfer data from the NET2272 local bus to a USB host. A bulk IN transaction from a Bulk IN Endpoint consists of the following:

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 sending 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's buffer, it is returned to the host. If the endpoint has no data to return, it returns either a zero length packet (signaling that there is no more data available) or a NAK handshake (the data is not available yet).

Packet Validated	Amount of Data in Buffer	Action
0	< Max Packet Size	NAK to host
X	>= Max Packet Size	Return data to host
1	empty	Zero length packet to host
1	>0	Return data to host

After the packet has been sent 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 will not be set, and the same packet is sent to the host when another IN token is received. The retry operation is transparent to the local CPU.

If the local CPU has stalled this endpoint by setting the *Endpoint Halt* bit, the NET2272 will respond to the IN token with a STALL handshake to the host.

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

5.6.5 Interrupt Endpoints

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

5.6.5.1 Interrupt Out Transactions

Interrupt Out endpoints are used to transfer data from a USB host to the NET2272 local bus. An interrupt OUT transaction to an Interrupt OUT endpoint consists of the following:

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 almost the same as a Bulk OUT endpoint, except for the toggle bit. If the *Interrupt Mode* bit is cleared, the toggle bit of the Interrupt OUT endpoint is initialized to 0 (DATA0 PID), and behaves the same as a Bulk OUT endpoint. If the *Interrupt Mode* bit is set, the toggle bit of the Interrupt OUT endpoint changes after each data packet is received from the host, without regard to the Status stage. Note that the PING protocol is not allowed for Interrupt OUT endpoints, as per the USB Specification.

5.6.5.2 Interrupt In Endpoints

An Interrupt IN endpoint is polled at a rate which is specified in the endpoint descriptor. An interrupt transaction from an Interrupt IN endpoint consists of the following:

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 *Interrupt Mode* bit is cleared, the toggle bit of the Interrupt IN endpoint is initialized to 0 (DATA0 PID), and behaves the same as a Bulk IN endpoint. An interrupt endpoint may be used to communicate rate feedback information for certain types of isochronous functions. To support this mode, the *Interrupt Mode* bit is set, and the toggle bit of the Interrupt IN endpoint changes after each data packet is sent to the host, without regard to the Status stage.

5.6.5.3 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 can be any value from 1 to 1024. Normal INTERRUPT endpoints in full-speed mode can set MAXPKT from 1 to 64.

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

5.7 NetChip Virtual Endpoints

5.7.1 Overview:

The Net2272 features NetChip Virtual Endpoint hardware support which enables firmware to implement any number of USB device endpoints up to the maximum of 15 per direction, excluding endpoint 0 (see the USB 2.0 Specification sections 8.3.2.2 and 9.6.6). Hardware support for endpoint virtualization consists of the Virtual Endpoint registers (**VIRTOUT0, VIRTOUT1, VIRTIN0,** and **VIRTIN1)** and the Virtual Endpoint Interrupt.

In this section, "logical endpoint" refers to the endpoint number from the point of view of the host (embedded in IN, OUT, and SETUP tokens), "physical endpoint" refers to the Net2272 hardware endpoint (0, A, B, or C), and "unassigned endpoint" refers to a logical endpoint number that is not currently assigned (via the *Endpoint Address* field in the **EP_CFG** register) to a physical endpoint.

If the *Virtual Endpoint Enable* bit of the **USBCLT1** register is low (default), the Net2272 responds to a host request to an unassigned endpoint with a timeout. The host considers a timeout response to be a fatal error. The host retries a transaction with a fatal error, but after a fixed number of retries, the host shuts down the pipe (endpoint).

If *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 on endpoint 3 when none of the Net2272 physical endpoints is assigned to address 3 with direction IN, bit 3 of the **VIRTIN0** register is set (and the Net2272 NAKs the IN request).

While any of the Virtual Endpoint register bits are set, the *Virtual Endpoint Interrupt* status bit will be set. If this interrupt is enabled, firmware is notified that the host has tried to access an unassigned endpoint. Firmware can then re-assign one of the physical endpoints to the new logical endpoint so the data transaction can proceed.

Virtual Endpoint participation is completely flexible: all physical endpoints (excluding endpoint 0) may participate in Virtual Endpoint re-assignment, or some physical endpoints can be dedicated to specific high-usage logical endpoints.

5.7.2 Endpoint Virtualization

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

When re-assigning endpoints, firmware must take care that USB traffic is not disturbed. Specifically, an endpoint should not be reprogrammed or flushed while the endpoint is enabled. NetChip 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, so firmware should first disable the endpoint, and then check that the enable has succeeded (there may be a delay while a pending USB transaction completes).

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

Before re-assigning an IN endpoint, firmware should:

1. Stop loading data into the endpoint
   2a. Wait until there are no buffers are waiting to be sent to the host (buffer states are available in the **EP_BUFF_STATE** register). Note that because of the USB-inherent problem discussed in section 8.5.3.3 of the USB 2.0 Specification, waiting for the endpoint to empty can potentially lead to deadlock.
   3a. Write 0 to the *Endpoint Enable* bit in the **EP_CFG** register.
   4a. Check that the *Endpoint Enable* bit is clear.

   OR

   2b. Write 0 to the *Endpoint Enable* bit in the **EP_CFG** register.
   3b. Check that the *Endpoint Enable* bit is clear.
   4b. Read out and store any packets still loaded in the endpoint buffers. Reading is accomplished by clearing the *Endpoint Direction* bit in the **EP_CFG** register so that buffer data is available to **EP_DATA,** and the count is available in **EP_AVAIL.** Note that zero-length packets should also be detected, read out, and stored. Zero-length packets can be detected from the **EP_BUFF_STATE** register.
5. Save the endpoint registers: **EP_CFG, EP_IRQENB, EP_TRANSFER, EP_RSPSET,** and **EP_MAXPKT.**
6. Re-assign the endpoint.

Before re-assigning an OUT endpoint:

1. Write 0 to the *Endpoint Enable* bit in the **EP_CFG** register.
2. Check that the *Endpoint Enable* bit is clear.
3. If there is any data available in the endpoint buffers, read it all out. Note that the host may send a packet after firmware clears *Endpoint Enable* but before the endpoint is disabled.
4. Save the endpoint registers: **EP_CFG, EP_IRQENB, EP_TRANSFER, EP_RSPSET,** and **EP_MAXPKT.**
5. Re-assign the endpoint.

Re-assigning a physical endpoint consists of programming the direction, type, and logical endpoint number and setting *Endpoint Enable* in **EP_CFG** (these can all be done in a single register write operation), and loading **EP_IRQENB, EP_TRANSFER, EP_RSPSET,** and **EP_MAXPKT.** The endpoint should also be flushed before loading any data or enabling it.

Restoring **EP_RSPSET/CLR** 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. Note that 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 (method 2b-4b above), care should be take to restore **EP_TRANSFER** correctly. Either **EP_TRANSFER** can be loaded with the stored **EP_TRANSFER** value plus the count of stored data before the stored data is loaded, or the stored data can be loaded first (and packets validated by writing 0 to **EP_TRANSFER**) followed by restoring **EP_TRANSFER** to the stored value.

5.7.3 Efficiency Considerations:

Depending on the number of virtual endpoints and the host-controller requirements, firmware may need to prioritize virtual endpoint re-assignment. For example, a simple scheme of scheduling the lowest virtual endpoint request each time might end up starving higher logical endpoint addresses. A "round-robin" priority is one method to ensure at least some data travel on all endpoints.

Note that INTERRUPT endpoints may require special care because the polling interval between accesses can be very long. It is not efficient to detect the endpoint access on an INTERRUPT endpoint (which was NAKed), switch to that endpoint, and then detect and switch to a different endpoint before the next INTERRUPT polling interval arrives. Once possible solution is to “lock down” the physical endpoint (prevent further endpoint switching) until a minimum number of packets have passed through the endpoint.

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

5.7.4 Deadlock Considerations:

Usually, the USB host-controller retries NAKed BULK transactions in a "round-robin" priority, so deadlocks will not normally occur. However, it may be possible that some host drivers may be susceptible to deadlocks. Specific device and host driver implementations should be evaluated specifically for deadlock exposure.

For example, step 2a of re-assigning an IN endpoint (above) requires firmware to wait until the physical endpoint is empty. If the host is not requesting data on the (old) logical endpoint, but is instead waiting for a transaction on the new logical endpoint (the logical endpoint firmware is trying to switch to, but is prevented because the physical endpoint is not empty), deadlock occurs.

This particular deadlock can be broken by flushing the IN endpoint if the device is prepared to reload the packet(s), or if the data can be discarded. Alternatively, the deadlock can be avoided entirely by not loading any data into the IN logical endpoint until the host has sent an IN token request, or by using method 2b-4b instead.

Another potential source of deadlock is the USB-inherent problem discussed in section 8.5.3.3 of the USB 2.0 Specification. In this situation, the endpoint is not able to flush the final packet of a transfer because it does not know that the host has received it correctly, and the host may not send another IN on the same endpoint for an indeterminate time.

5.7.5 Buffer Control

The Net2272 buffers can be read and written from the local bus. This feature can be used for both chip diagnostics (power-on tests) and during Virtual Endpoint context switching.

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

After a packet has been validated (or both if the endpoint is configured to be double-buffered), the *Endpoint Direction* bit can be switched to OUT (by clearing bit 4 of **EP_CFG**) and the buffer data can be read out. **EP_AVAIL** indicates the number of bytes available in the buffer.

Zero-length packets behave the same as OUT zero-length packets sent by the host, and they are similarly flushed by a dummy read from **EP_DATA**. The presence of a zero-length packet is indicated by *Local OUT ZLP* (bit 6 of **EP_STAT1**) or by the buffer states in **EP_BUFF_STATES**.

Note that data is only available for reading on the OUT **EP_DATA** register after the packet has been validated while the endpoint is configured for IN. Data written but not validated is lost when the direction is switched to OUT.

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

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

5.7.6 Summary

NetChip'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 any host driver (including USB Class Drivers) can be supported.

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

5.8 Packet Buffers

The NET2272 contains one 3-Kbyte bank of memory that is allocated to the endpoint packet buffers. The configuration of the endpoint A and B buffers is selected by the *Buffer Configuration* field of the **LOCCTL** configuration register. Available configurations for endpoints A and B are:

- 512 bytes, double-buffered (total of 1K bytes)
- 1024 bytes, single-buffered
- 1024 bytes double-buffered

The total size of all of the endpoint A and B buffers cannot exceed 2 Kbytes. Endpoint C has a 1K byte buffer, configured as two 512-byte buffers, and endpoint 0 has a 128 byte buffer, configured as two 64-byte buffers. Data is stored in the buffers in 32-bit words, so each entry contains 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 IN Endpoint Buffers

IN packet data is written by the local CPU or DMA into one of the IN endpoint buffers. Once the buffer data has been validated, it is returned to the USB host in response to an IN token. The NET2272 will not send more than **EP_MAXPKT** bytes per packet. Once a packet has been written into a double-buffered buffer and validated, the local CPU can continue loading data for the next packet. The NET2272 will automatically divide the data flow into individual packets with a maximum size determined by the associated **EP_MAXPKT** register. This allows USB transactions to overlap with loading of data from the local bus.

If the buffer data hasn't been validated, the NET2272 responds to an IN token with a NAK handshake. There are several methods for validating the data in the IN buffer:

- For large amounts of data, the local bus controller can write data to the buffer as long as there is space available. When there are at least **EP_MAXPKT** bytes in the buffer, the NET2272 will respond to an IN token with a packet of data. If the entire data transfer is a multiple of **EP_MAXPKT** bytes, then nothing else needs to be done to validate the buffer data. If a zero length packet needs to be sent to the host, the local CPU can write a zero to the **EP_TRANSFER** register without writing any additional data to the buffer.
- For moderate amounts of data (between **EP_MAXPKT** and 16 Mbytes), a counter (**EP_TRANSFER**) is used. This counter is initialized to the total transfer byte count before any data is written to the buffer. The counter is decremented as data is written to the buffer. When the counter reaches zero, the remaining data 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 has **EP_MAXPKT** bytes, then the NET2272 will respond 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. Then a zero is written to the **EP_TRANSFER** register, thus causing the data to be validated.
- For a DMA write terminated with an EOT, the buffer data is validated if the *DMA Buffer Valid* bit in the **DMAREQ** register is set. This causes a short or zero-length packet to be transmitted in response to the next IN token.

Up to 2 short (less than **EP_MAXPKT** bytes) packets can be stored in a double-buffered packet buffer.

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

5.8.1.1 16-bit Post-Validation

Post-validation is the technique in which data is written to the buffer before it is validated. The following steps must be followed to post-validate an odd length packet, size 2n+1, when operating in 16-bit mode:

- Write '2n' bytes using 'n' 16-bit transactions to the endpoint buffer via **EP_DATA.**
- Change to an 8-bit bus by clearing the *Data Width* bit in the **LOCCTL** register.
- Write the last byte to the endpoint buffer via **EP_DATA**.
- Post-validate the buffer by writing 0 to **EP_TRANSFER0**.
- Change back to a 16-bit bus by setting the *Data Width* bit in the **LOCCTL** register.

5.8.2 OUT Endpoint Buffers

When receiving data, the NET2272 will NAK the host (indicating that it cannot accept the data) if either the buffer runs out of room, or if both the *NAK OUT Packets Mode* bit and the *NAK OUT Packets* bits are set. If the packets received are of maximum size, then additional packets can be received independently of the *NAK OUT Packets Mode* bit. This bit will only cause additional OUT packets to be NAKed if the last packet received was a short packet.

If *NAK OUT Packets Mode* is true (blocking mode), USB OUT transfers can overlap with the local CPU unloading the data using the following sequence:

- Local CPU responds to the *Data Packet Received Interrupt* and reads the **EP_AVAIL** register so it knows how many bytes are in the current packet.
- Local CPU clears the *Data Packet Received Interrupt* and the *NAK OUT Packets* bit, allowing the next packet to be received.
- Now the local CPU can unload data from the buffer while the next USB OUT transaction is occurring.

If *NAK OUT Packets Mode* is false (non-blocking mode), the NET2272 will accept packets as long as there is room for the complete packet in the next available buffer. Note that there are no indications of packet boundaries when there are multiple packets in the buffers in a double-buffered configuration.

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

5.9 USB Test Modes

The *Force Full Speed* and *Force High Speed* bits of the **XCVRDIAG** register can be used to force the NET2272 into full and high speed modes, respectively. These forcing bits **must not be used in normal operation**; they are for testing purposed only. In normal operation, the NET2272 automatically performs USB 2.0 Chirp Protocol negotiation with the host to determine the correct operating speed.

USB 2.0 Test Mode support is provided via the *Test Mode Select* field of the **USBTEST** register. These bits select the appropriate USB Test Mode settings (see section 9.4.9 in the USB Specification Revision 2.0 for more details). Normally, the host sends a SET_FEATURE request with the Test Selector in the upper byte of wIndex. The Test Selector can be copied directly into the NET2272 **USBTEST** register to select the correct test mode.

Note that USB Test Mode settings only have an effect if the NET2272 is in high-speed mode. Also, if the NET2272 is in high-speed mode, and the *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.

Note also that the NET2272 can be forced into high-speed mode (using the *Force High Speed* bit) even if the NET2272 is not connected to a host controller. After selecting the 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 (0x04) Test Mode Selector requires a specific packet to be returned by the device. The NET2272 will respond correctly by:

1. Set *Test Mode Select* to 0x04
2. Flush endpoint 0
3. Load the following 53 (0x35) byte packet 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

You may validate the packet with your normal validation method, either pre-validating by writing 0x35 into **EP_TRANSFER**[EP0] before loading the bytes, or by writing 0 to **EP_TRANSFER**[EP0] after loading the bytes.

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

6 Interrupt and Status Register Operation

6.1 Interrupt Status Registers (IRQSTAT0, IRQSTAT1)

Bits 3:0 of the **IRQSTAT0** register indicate whether one of the endpoints 0, A-C has an interrupt pending. These bits cannot be written, and can cause a local interrupt if the corresponding interrupt enable bits are set in the **IRQENB0** register. Bit 7 is automatically set when a start-of-frame (SOF) token is received, and is cleared by writing a 1. This bit can cause a local interrupt if the corresponding interrupt enable bit is set in the **IRQENB0** register. Note that the interrupt bits can be set without the corresponding interrupt enable bit being set. This allows the local CPU to operate in a polled, as well as an interrupt driven environment.

Bits 6:5 of **IRQSTAT0** and bits 7:4 and 2:1 of **IRQSTAT1** are set when a particular event occurs in the NET2272, and are cleared by writing a 1 to the corresponding bit. These bits can cause a local interrupt if the corresponding interrupt enable bits are set in the **IRQENB0** and **IRQENB1** registers.

Bit 3 of **IRQSTAT1** is set when there is a suspend request from the host, but it typically not enabled to generate an interrupt. Writing a 1 clears this bit and causes the 2272 to enter the suspend state.

6.2 Endpoint Response Registers (EPRSP_CLR, EPRSP_SET)

Each endpoint has a pair of Endpoint Response Registers. The bits in these registers determine how the NET2272 will respond to various situations during a USB transaction. Writing a 1 to any of the bits in the **EP_RSPCLR** register will clear the corresponding bits. Writing a 1 to any of the bits in the **EP_RSPSET** register will set the corresponding bits. Reading either of the registers returns the current state of the bits.

6.3 Endpoint Status Register (EP_STAT0, EP_STAT1)

Each endpoint has a pair of Endpoint Status Registers**.** Each of the bits of these registers is set when a particular endpoint event occurs, and is cleared by writing a 1 to the corresponding bit. A local interrupt can be generated if the corresponding interrupt enable bits are set in the **EP_IRQENB** registers. Reading the **EP_STAT** registers returns the current state of the bits.

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

7 Power Management

7.1 Suspend Mode

When there is a three-millisecond period of inactivity on the USB, the USB specification requires a device to enter into a low-power suspended state. A low power device may not draw more than 500 μA, and a high-power device may not draw more than 2.5 mA while in this state. This requirement only applies to bus-powered devices. To facilitate this, the NET2272 provides a *Suspend Request Change Interrupt* bit and a *Suspend Request Interrupt* bit. Additionally, the NET2272 allows local bus hardware to initiate a "device remote wake-up" to the USB.

7.1.1 The Suspend Sequence

The typical sequence of a suspend operation is as follows:

- During device configuration, the local CPU enables the *Suspend Request Change Interrupt* bit to generate a local interrupt.
- When the USB is idle for three milliseconds, the NET2272 sets the *Suspend Request Change Interrupt* bit, generating an interrupt to the local CPU. This interrupt can also occur if the NET2272 is not connected to a host, and the USB data lines are pulled to the idle state (DP high, DM low), or if the VBUS input is low.
- The local CPU accepts this interrupt by clearing the *Suspend Request Change Interrupt* bit, and performs the tasks required to ensure that no more than 500 μA of current is drawn from the USB power bus.
- The local CPU writes a 1 to the *Suspend Request Interrupt* bit to initiate the suspend.
- The LCLKO output continues to operate for 500 μsec before the NET2272 enters the suspend state. This allows time for a local CPU that uses LCLKO to power down.
- A device remote wakeup event will not be recognized during the 500 μsec suspend delay period.

In suspend mode, the NET2272's oscillator shuts down, and most output pins are tri-stated to conserve power (see section 3, Pin Description). Note that input pins to the NET2272 should not be allowed to float during suspend mode. The NET2272 will leave suspend mode by detecting a host initiated wake-up or by a device remote wake-up.

If a device is self-powered, it may ignore the USB suspend request and never write a 1 to the *Suspend Request Interrupt* bit.

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

7.1.2 Host-Initiated Wake-Up

The host may wake up the NET2272 by driving any non-idle state on the USB. The NET2272 will detect the host's wake-up request, and re-starts its internal oscillator. The host initiated wake-up is only recognized if the VBUS input pin is high, and the *USB Detect Enable* and *USB Root Port Wakeup Enable* bits in the **USBCTL0** register are set.

7.1.3 Device-Remote Wake-Up

The device hardware signals a device remote wake-up by driving the CS# input pin low. If the *I/O Wakeup* Enable bit in the **USBCTL0** register is set, the NET2272 re-starts its local oscillator. Two milliseconds after the CS# pin is asserted, the local CPU must write to the *Generate Resume* bit of the **USBCTL1** register. This will cause a 10-ms wake-up signal to be sent to the USB host.

7.1.4 Resume Interrupt

When the NET2272 begins either a Device-Remote Wake-Up or Host-Initiated Wake-Up, it may be programmed to generate a resume interrupt. The *Resume Interrupt* bit of the **IRQSTAT1** register is set when a resume is detected, and can be enabled to generate an interrupt with the *Resume Interrupt Enable* bit.

7.2 NET2272 Power Configuration

The USB specification defines both bus-powered and self-powered devices. A *bus-powered* device is a peripheral that derives all of its power from the upstream USB connector, while a *self-powered* device has an external power supply.

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

- A device not configured by the host can draw only 100 mA from the USB power pins.
- A device may not draw more than 500 mA from the USB connector's power pins.
- In suspend mode, the device may not draw more than 500 μA (or 2.5 mA for a high-power device) from the USB connector's power pins

If these power considerations can be met without the use of an external power supply, the device can be bus-powered; otherwise a self-powered design should be implemented.

7.2.1 Self-Powered Device

Generally, a device with higher power requirements will be self-powered. In a self-powered device, the NET2272 V_{DD} pins are powered by the local power supply. This allows the local bus to continue accessing the NET2272, even when the device is not connected to the USB bus. The USB connector's power pin is connected directly to the NET2272 VBUS pin, and is only used to detect whether it is connected to a USB host.

While the device is connected to the USB, the NET2272 will automatically request suspend mode when appropriate. The NET2272 should not be powered-down when its local bus is still connected to a powered-up device. There are ESD protection circuits in the NET2272 that will short V_{DD} pins to ground. If the V_{DD} pins are not powered, they will sink too much current from the board.

7.2.2 Low-Power Modes

7.2.2.1 USB Suspend (Unplugged from USB)

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

- Un-plug the USB cable.
- Clear the *USB Detect Enable* bit in the **USBCTL0** register.

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

In power-sensitive applications, the local CPU can force the NET2272 to enter low-power suspend mode when disconnected from the USB by writing a 1 to the *Suspend Request Interrupt* bit. The NET2272 will automatically wake-up when the peripheral is re-connected (cable plugged in and *USB Detect Enable* bit set) to the USB. *Do not force suspend mode unless the peripheral is disconnected from the USB. When the NET2272 is connected to the USB, it is a violation of the USB specification to enter the suspend state unless the upstream port has been idle for at least 3 milliseconds.*

This is the preferred method of suspending the NET2272, since a USB re-connection will automatically cause the NET2272 to wake-up and set the *Resume Interrupt* bit.

7.2.2.2 Power-On Standby

The local CPU can prevent the NET2272 from starting its oscillator on power-up by driving a LOW into the CS# pin while RESET# is asserted (LOW). In this state the NET2272 requires only a small quiescent standby current.

When the peripheral wishes to start the oscillator, it releases the CS# pin and continues to assert RESET# for a minimum of 2 milliseconds. Note that while the oscillator is stopped, the NET2272 cannot respond to USB requests, so the oscillator must be allowed to start when the peripheral detects a USB connection event. The local CPU is responsible for detecting the connection, and ending the standby condition.

This standby technique is appropriate when the device's power budget does not allow the NET2272 to be active long enough to shut it down by setting the *Suspend Request Interrupt* bit.

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

8 Configuration Registers

8.1 Register Description

The NET2272 occupies a 32-byte address space that can be accessed by a CPU on the local bus. Registers can be accessed directly or indirectly through a pointer register. Accessing the registers directly provides higher performance, while accessing the registers through the pointer register requires fewer physical address pins. The most commonly used registers have been located at the lowest addresses, thus providing the highest performance when using less than 5 address bits.

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

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.

For compatibility with future revisions, **reserved** bits within a register should always be written with a zero.

8.2 Register Summary

Register Groups
Main Control Registers
USB Control Registers
Endpoint Registers

8.2.1 Main Control Registers

Address	Register Name	Register Description	Page
00h	REGADDRPTR	Register Address Pointer	56
01h	REGDATA	Register Data	56
02h	IRQSTAT0	Interrupt Status Register (low byte)	56
03h	IRQSTAT1	Interrupt Status Register (high byte)	57
04h	PAGESEL	Endpoint Page Select Register	57
1Ch	DMAREQ	DMA Request Control	58
1Dh	SCRATCH	General Purpose Scratch-pad	58
20h	IRQENB0	Interrupt Enable Register (low byte)	59
21h	IRQENB1	Interrupt Enable Register (high byte)	59
22h	LOCCTL	Local Bus Control	60
23h	CHIPREV_LEGACY	Legacy Chip Silicon Revision	60
24h	LOCCTL1	Local Bus Control 1	60
25h	CHIPREV_2272	Net2272 Chip Silicon Revision	61

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

8.2.2 USB Control Registers

Address	Register Name	Register Description	Page
18h	USBCTL0	USB Control Register (low byte)	62
19h	USBCTL1	USB Control Register (high byte)	62
1Ah	FRAME0	USB Frame Number (low byte)	62
1Bh	FRAME1	USB Frame Number (high byte)	62
30h	OURADDR	Our USB Address	63
31h	USBDIAG	Diagnostic register	63
32h	USBTEST	USB 2.0 Test Control Register	64
33h	XCVRDIAG	USB Transceiver Diagnostic Register	64
34h	VIRTOUT0	Virtual OUT Interrupt 0	64
35h	VIRTOUT1	Virtual OUT Interrupt 1	65
36h	VIRTIN0	Virtual IN Interrupt 0	65
37h	VIRTIN1	Virtual IN Interrupt 1	65
40h	SETUP0	Setup byte 0	65
41h	SETUP1	Setup byte 1	66
42h	SETUP2	Setup byte 2	66
43h	SETUP3	Setup byte 3	66
44h	SETUP4	Setup byte 4	66
45h	SETUP5	Setup byte 5	66
46h	SETUP6	Setup byte 6	67
47h	SETUP7	Setup byte 7	67

8.2.3 Endpoint Registers

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

Address	Register Name	Register Description	Page
05h	EP_DATA	Endpoint Data Register	68
06h	EP_STAT0	Endpoint Status (low byte)	68
07h	EP_STAT1	Endpoint Status (high byte)	69
08h	EP_TRANSFER0	IN endpoint byte count (byte 0)	69
09h	EP_TRANSFER1	IN endpoint byte count (byte 1)	69
0Ah	EP_TRANSFER2	IN endpoint byte count (byte 2)	69
0Bh	EP_IRQENB	Endpoint interrupt enable	70
0Ch	EP_AVAIL0	Buffer space/byte count (byte 0)	70
0Dh	EP_AVAIL1	Buffer space/byte count (byte 1)	70
0Eh	EP_RSPCLR	Endpoint Response Control Clear	71
0Fh	EP_RSPSET	Endpoint Response Control Set	72
28h	EP_MAXPKT0	Endpoint Maximum Packet (low byte)	72
29h	EP_MAXPKT1	Endpoint Maximum Packet (high byte)	72
2Ah	EP_CFG	Endpoint configuration	73
2Bh	EP_HBW	Endpoint high bandwidth	73
2Ch	EP_BUFF_STATES	Endpoint buffer states	74

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

8.3 Numeric Register Listing

This table shows the number of address bits required to access a register directly. If only four address bits are supplied to the chip, then the registers that require 5 address bits must be addressed indirectly using **REGADDRPTR** and **REGDATA**.
Addresses marked with (P) are paged registers selected by the *Page Select* field in the **PAGESEL** register.

Address	Address bits Required	D[15:8]	D[7:0]	Register Description
00h	1		REGADDRPTR	Register Address Pointer for indirect register addressing
01h	1	REGDATA1	REGDATA	Register Data port for indirect register addressing
02h	2		IRQSTAT0	Interrupt Status Register (low byte)
03h	2		IRQSTAT1	Interrupt Status Register (high byte)
04h	3		PAGESEL	Page Select Register. Select current Endpoint
05h (P)	3	EP_DATA1	EP_DATA	Endpoint Data Register
06h (P)	3		EP_STAT0	Endpoint Main Status
07h (P)	3		EP_STAT1	
08h (P)	4		EP_TRANSFER0	For IN endpoint, number of bytes to transfer to host.
09h (P)	4		EP_TRANSFER1	
0Ah (P)	4		EP_TRANSFER2	
0Bh (P)	4		EP_IRQENB	Endpoint Interrupt Enable
0Ch (P)	4		EP_AVAIL0	For IN endpoints, number of available spaces in buffer.
0Dh (P)	4		EP_AVAIL1	For OUT endpoints, number of bytes in buffer.
0Eh (P)	4		EP_RSPCLR	Endpoint Response Register Clear
0Fh (P)	4		EP_RSPSET	Endpoint Response Register Set
10-17h	5		Reserved	
18h	5		USBCTL0	USB Control
19h	5		USBCTL1	
1Ah	5		FRAME0	Frame Counter
1Bh	5		FRAME1	
1Ch	5		DMAREQ	DMA Request Control Register
1Dh	5		SCRATCH	General-Purpose Scratchpad register
1E-1Fh	5		Reserved	
20h	Indirect Only		IRQENB0	Interrupt Enable Register
21h	Indirect Only		IRQENB1	
22h	Indirect Only		LOCCTL	Local Bus Control Register
23h	Indirect Only		CHIPREV_LEGACY	Legacy Chip Revision Number
24h	Indirect Only		LOCCTL1	Local Bus Control Register 1
25h	Indirect Only		CHIPREV_2272	Net2272 Chip Revision Number
26-27h	Indirect Only		Reserved	
28h (P)	Indirect Only		EP_MAXPKT0	Endpoint Max Packet Size
29h (P)	Indirect Only		EP_MAXPKT1	
2Ah (P)	Indirect Only		EP_CFG	Endpoint Configuration
2Bh (P)	Indirect Only		EP_HBW	Endpoint High-Bandwidth
2Ch (P)	Indirect Only		EP_BUFF_STATES	Endpoint Buffer States
2D-2Fh	Indirect Only		Reserved	
30h	Indirect Only		OURADDR	Our USB address
31h	Indirect Only		USBDIAG	Diagnostic Register
32h	Indirect Only		USBTEST	USB 2.0 Test Control Register
33h	Indirect Only		XCVRDIAG	Transceiver Diagnostic Register
34h	Indirect Only		VIRTOUT0	Virtual OUT Interrupt 0
35h	Indirect Only		VIRTOUT1	Virtual OUT Interrupt 1
36h	Indirect Only		VIRTIN0	Virtual IN Interrupt 0
37h	Indirect Only		VIRTIN1	Virtual IN Interrupt 1
38-3Fh	Indirect Only		Reserved	
40h	Indirect Only		SETUP0	Setup byte 0

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

41h	Indirect Only		SETUP1	Setup byte 1
42h	Indirect Only		SETUP2	Setup byte 2
43h	Indirect Only		SETUP3	Setup byte 3
44h	Indirect Only		SETUP4	Setup byte 4
45h	Indirect Only		SETUP5	Setup byte 5
46h	Indirect Only		SETUP6	Setup byte 6
47h	Indirect Only		SETUP7	Setup byte 7

8.4 Main Control Registers

8.4.1 (Address 00h; REGADDRPTR) Indirect Register Address Pointer

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

8.4.2 (Address 01h; REGDATA) Indirect Register Data

Bits	Description	Read	Write	Default Value
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	0

8.4.3 (Address 02h; IRQSTAT0) Interrupt Status Register (low byte)

Bits	Description	Read	Write	Default Value
7	**SOF Interrupt.** This bit indicates when a start-of-frame packet has been received by the NET2272. Writing a 1 clears this status bit.	Yes	Yes/CLR	0
6	**DMA Done Interrupt.** For IN endpoints, this bit indicates that EOT# has been asserted, the **EP_TRANSFER** counter reaches zero during a DMA, or the corresponding **EP_TRANSFER** counter is loaded with a 0. For OUT endpoints, this bit indicates that EOT# has been asserted, or that a short packet has been received and the endpoint buffers have gone empty. Writing a 1 clears this status bit. This bit is set independently of the corresponding interrupt enable bit.	Yes	Yes/CLR	0
5	**Setup Packet Interrupt.** This bit is set when a setup packet has been received from the host. Writing a 1 clears this status bit.	Yes	Yes/CLR	0
4	**Virtualized Endpoint Interrupt.** This bit is set when one of the Virtual Endpoint interrupts is set.	Yes	No	0
3	**Endpoint C Interrupt.** This bit conveys the interrupt status for Endpoint C. When set, Endpoint C's interrupt status register should be read to determine the cause of the interrupt. This bit is set independently of the interrupt enable bit.	Yes	No	0
2	**Endpoint B Interrupt.** This bit conveys the interrupt status for Endpoint B. When set, Endpoint B's interrupt status register should be read to determine the cause of the interrupt. This bit is set independently of the interrupt enable bit.	Yes	No	0
1	**Endpoint A Interrupt.** This bit conveys the interrupt status for Endpoint A. When set, Endpoint A's interrupt status register should be read to determine the cause of the interrupt. This bit is set independently of the interrupt enable bit.	Yes	No	0
0	**Endpoint 0 Interrupt.** This bit conveys the interrupt status for Endpoint 0. When set, Endpoint 0's interrupt status register should be read to determine the cause of the interrupt. This bit is set independently of the interrupt enable bit.	Yes	No	0

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

8.4.4 (Address 03h; IRQSTAT1) Interrupt Status Register (high byte)

Bits	Description	Read	Write	Default Value
7	**Reset Status.** When set, this bit indicates that either the RESET# pin is asserted, or a USB root port reset is currently active.	Yes	No	0
6	**Root Port Reset Interrupt.** This bit indicates a change in state of the root port reset detector. Writing a 1 clears this status bit.	Yes	Yes/CLR	0
5	**Resume Interrupt.** When set, this bit indicates that a device resume has occurred. Writing a 1 clears this status bit.	Yes	Yes/CLR	0
4	**Suspend Request Change Interrupt.** This bit is set whenever there is a change in the Suspend Request Interrupt state (bit 3 of this register). Writing a 1 clears this status bit.	Yes	Yes/CLR	0
3	**Suspend Request Interrupt**. This bit is set when the NET2272 detects a USB Suspend request from the host. The Suspend Request state cannot be set or cleared by writing this bit. Instead, writing a 1 to this bit puts the NET2272 into the low-power suspend mode (see section 7.1.1).	Yes	Yes/ Suspend	0
2	**VBUS Interrupt.** When set, this bit indicates that a change occurred on the VBUS input pin. Read the USBCTL1 register for the current state of this pin. Writing a 1 clears this status bit.	Yes	Yes/CLR	0
1	**Control Status Interrupt.** This bit is set when an IN or OUT token indicating Control Status has been received. Writing a 1 clears this status bit.	Yes	Yes/CLR	0
0	**Reserved.**	0	No	0

8.4.5 (Address 04h; PAGESEL) Endpoint Page Select Register

Bits	Description	Read	Write	Default Value
7:2	**Reserved.**	0	No	0
1:0	**Page Select.** The NET2272 uses a paged architecture for accessing the registers associated with each endpoint. This field selects which set of endpoint registers can be accessed. **VALUE** **ENDPOINT** 00 EP 0 01 EP A 10 EP B 11 EP C	Yes	Yes	00

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

8.4.6 (Address 1Ch; DMAREQ) DMA Request Control Register

Bits	Description	Read	Write	Default Value
7	**DMA Buffer Valid.** When clear, the buffer will not be automatically validated at the end of a DMA transfer. When set, the buffer is automatically validated at the end of a DMA if EOT is asserted. This bit only applies to IN endpoints.	Yes	Yes	0
6	**DMA Request.** This status bit reflects the state of the DREQ output pin, and allows a CPU on the local bus to monitor DMA transfers.	Yes	No	0
5	**DMA Request Enable.** Writing a 1 to this bit enables the NET2272 to start requesting DMA cycles from a DMA controller on the local bus. If the EOT input is asserted, or the **EP_TRANSFER** counter reaches zero, or a short OUT packet is received and the endpoint buffer becomes empty, this bit is automatically reset. A CPU on the local bus may 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. This bit can be read to determine whether a DMA transfer is still in progress.	Yes	Yes	0
4	**DMA Control DACK.** When clear, only DACK is used to perform DMA read and write transactions. When set, IOR# and IOW# (or DMARD# and DMAWR# for split mode DMA) control signals are used with DACK to perform DMA read and write transactions.	Yes	Yes	0
3	**EOT Polarity.** When clear, the EOT input pin is active low. When set, the EOT input pin is active high.	Yes	Yes	0
2	**DACK Polarity.** When clear, the DACK input pin is active low. When set, the DACK input pin is active high.	Yes	Yes	0
1	**DREQ Polarity.** When clear, the DREQ output pin is active low. When set, the DREQ output pin is active high.	Yes	Yes	1
0	**DMA Endpoint Select.** This field determines which endpoint is being accessed during a DMA channel transfer. **Value** **Endpoint used during DMA** 0 Endpoint A 1 Endpoint B	Yes	Yes	0

8.4.7 (Address 1Dh; SCRATCH) Scratchpad Register

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

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

8.4.8 (Address 20h; IRQENB0) Interrupt Enable Register (low byte)

Bits	Description	Read	Write	Default Value
7	**SOF Interrupt Enable.** When set, this bit enables a local interrupt to be generated when a start-of-frame packet is received by the NET2272.	Yes	Yes	0
6	**DMA Done Interrupt Enable.** When set, this bit enables a local interrupt to be generated when an EOT signal is received from the DMA controller, or when the EP_TRANSFER counter reaches 0zero during a DMA writes to an IN endpoint.	Yes	Yes	0
5	**Setup Packet Interrupt Enable.** When set, this bit enables a local interrupt to be generated when a setup packet has been received from the host.	Yes	Yes	0
4	**Virtualized Endpoint Interrupt Enable.** When set, this bit enables a local interrupt to be generated when an IN or OUT token to a virtualized endpoint is detected.	Yes	Yes	0
3	**Endpoint C Interrupt Enable.** When set, this bit enables a local interrupt to be set when an interrupt is active on this endpoint.	Yes	Yes	0
2	**Endpoint B Interrupt Enable.** When set, this bit enables a local interrupt to be set when an interrupt is active on this endpoint.	Yes	Yes	0
1	**Endpoint A Interrupt Enable.** When set, this bit enables a local interrupt to be set when an interrupt is active on this endpoint.	Yes	Yes	0
0	**Endpoint 0 Interrupt Enable.** When set, this bit enables a local interrupt to be set when an interrupt is active on this endpoint.	Yes	Yes	0

8.4.9 (Address 21h; IRQENB1) Interrupt Enable Register (high byte)

Bits	Description	Read	Write	Default Value
7	**Reserved.**	0	No	0
6	**Root Port Reset Interrupt Enable.** When set, this bit enables a local interrupt to be generated when a root port reset is detected.	Yes	Yes	0
5	**Resume Interrupt Enable.** When set, this bit enables a local interrupt to be generated when a device resume has been detected.	Yes	Yes	0
4	**Suspend Request Change Interrupt Enable.** When set, this bit enables a local interrupt to be generated when a change in the Suspend Request Interrupt state is detected.	Yes	Yes	0
3	**Suspend Request Interrupt Enable.** When set, this bit enables a local interrupt to be generated when a USB Suspend Request from the host is detected.	Yes	Yes	0
2	**VBUS Interrupt Enable.** When set, this bit enables a local interrupt to be generated when a change has been detected on the VBUS pin.	Yes	Yes	0
1	**Control Status Interrupt Enable.** When set, this bit enables a local interrupt to be generated when an IN or OUT token indicating Control Status has been received.	Yes	Yes	0
0	**Reserved.**	0	No	0

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

8.4.10 (Address 22h; LOCCTL) Local Bus Control Register

Bits	Description	Read	Write	Default Value
7:6	**Buffer Configuration**. Buffer configuration for endpoints A and B. For example, if value 01 is selected, Endpoint A will be allocated a single buffer with buffer size set to 1024 bytes, while Endpoint B will be allocated a double buffer with each buffer size set to 512 bytes. Actual packets sent/received can be less than or equal to the **EP_MAXPKT** size for the selected endpoint. The **EP_MAXPKT** size must be less than or equal to the allocated buffer size. **Value** **EP_A** **EP_B** 00 512 db * 512 db 01 1024 512 db 10 1024 1024 11 1024 db Disabled * "db" means double buffered. All values shown in bytes.	Yes	Yes	00
5	**Byte Swap.** When clear, local data bus LD[15:0] is connected to the endpoint buffer with no byte swapping. When set, the two bytes of a 16-bit data bus are swapped before connecting to the endpoint buffer.	Yes	Yes	0
4	**DMA Split Bus Mode**. When clear, I/O and DMA accesses share the same data bus. When set, I/O accesses to the configuration registers or buffers use LD[7:0], and DMA accesses to the buffers use LD[15:8], thus splitting the data bus for CPU and DMA accesses.	Yes	Yes	0
3:1	**Local Clock Output.** This field controls the frequency of the LCLKO pin. Value Frequency 000 0 (off) 001 3.75 MHz **010 7.5 MHz (default)** 011 15 MHz 100 30 MHz 101 60 MHz 110 Reserved 111 Reserved	Yes	Yes	2
0	**Data Width.** This field controls the width of the local data bus for **EP_DATA** accesses to endpoint buffers. Write to this register using the lower 8 bits of the data bus to switch to 16-bit mode. This bit does not affect accesses to any other registers. Value Width 0 8 bits 1 16 bits	Yes	Yes	0

8.4.11 (Address 23h; CHIPREV_LEGACY) Legacy Silicon Revision Register

Bits	Description	Read	Write	Default Value
7:0	**Legacy Chip Revision.** This register returns a legacy silicon revision number for use by Net2270 firmware.	Yes	No	‘h40

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.

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

8.4.12 (Address 24h; LOCCTL1) Local Bus Control Register 1

Bits	Description	Read	Write	Default Value
7:3	**Reserved**	Yes	No	0
2	**DMA DACK Enable.** When clear, the NET2272 does not recognize the DACK input pin. When set, the DACK input pin is enabled. This bit is automatically set when the *DMA Request Enable* bit in the **DMAREQ** register is set. In split DMA mode, this bit should not be cleared if there is a possibility that DACK will be asserted.	Yes	Yes	0
1:0	**DMA Mode**. This field determines the behavior of DREQ during DMA transactions. Value Description 00 **Slow DREQ**. DREQ is de-asserted several clock periods after the start of a DMA transaction, and is re-asserted several clock periods after the end of the DMA transactions. This mode is compatible with the Net2270. 01 **Fast DREQ.** DREQ is de-asserted at the beginning of a DMA transaction, and is re-asserted soon after the end of the DMA transaction. This mode provides higher DMA performance. 10 **Burst Mode.** DREQ is asserted when the *DMA Request Enable* bit is set and there is space/data available in the endpoint buffer. DREQ remains asserted until either the buffer becomes empty/full, the *DMA Request Enable* bit is cleared, EOT is asserted, or EP_TRANSFER counts to 0 for an IN endpoint. 11 **Reserved.**	Yes	Yes	0

8.4.13 (Address 25h; CHIPREV_2272) Net2272 Silicon Revision Register

Bits	Description	Read	Write	Default Value
7:0	**Chip Revision.** This register returns the Net2272 silicon revision number.	Yes	No	‘h11

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.

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

8.5 USB Control Registers

8.5.1 (Address 18h; USBCTL0) USB Control Register (low byte)

Bits	Description	Read	Write	Default Value
7:6	**Reserved**	Yes	No	11
5	**USB Root Port Wakeup Enable.** When clear, the wake-up condition is not detected. When set, the root port wake-up condition is detected when activity is detected on the USB.	Yes	Yes	1
4	**Reserved.**	Yes	Yes	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. This bit should not be set until the configuration registers have been programmed. When operating as a bus-powered device, the registers should be programmed and this bit should be set promptly after VBUS has been detected.	Yes	Yes	0
2	**Reserved.**	Yes	No	0
1	**I/O Wakeup Enable.** When clear, asserting CS# will not cause a device remote wakeup. When set, this bit enables the assertion of CS# to initiate a device remote wakeup.	Yes	Yes	0
0	**Reserved.**	Yes	No	0

8.5.2 (Address 19h; USBCTL1) USB Control Register (high byte)

Bits	Description	Read	Write	Default Value
7:5	**Reserved.**	0	No	0
4	**Virtual Endpoint Enable.** When set, this bit enables the virtual endpoint feature of the Net2272. A NAK is returned to the USB host if an IN or TOKEN is received for an endpoint that is virtualized.	Yes	Yes	0
3	**Generate Resume.** Writing a 1 to this bit causes a Resume sequence to be initiated to the host if device remote wakeup is enabled. This bit should be written after a device remote wakeup has been generated (CS# pin asserted). This bit is self-clearing, and reading always returns a 0.	No	Yes/ Resume	0
2	**USB High Speed.** When set, this bit indicates that the transceiver is operating in high speed (480 Mbits/sec) mode.	Yes	No	0
1	**USB Full Speed.** When set, this bit indicates that the transceiver is operating in full speed (12 Mbits/sec) mode.	Yes	No	0
0	**VBUS pin.** This bit indicates the state of the VBUS pin. When set, this bit indicates that the NET2272 is connected to the USB.	Yes	No	0

8.5.3 (Address 1Ah; FRAME0) Frame Counter (low byte)

Bits	Description	Read	Write	Default Value
7:0	**FRAME[7:0].** This field contains the frame counter from the most recent start-of-frame packet.	Yes	No	0

8.5.4 (Address 1Bh; FRAME1) Frame Counter (high byte)

Bits	Description	Read	Write	Default Value
7:3	**Reserved.**	0	No	0
2:0	**FRAME[10:8].** This field contains the frame counter from the most recent start-of-frame packet.	Yes	No	0

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

8.5.5 (Address 30h; OURADDR) Our Current USB Address

Bits	Description	Read	Write	Default Value
7	**Force Immediate.** If this bit is set when this register is being written, the NET2272 USB address is updated immediately, without waiting for a valid status phase from the USB host.	0	Yes/Force	0
6:0	**Our Address.** This field contains the current USB address of the device. This field is cleared when a root port reset is detected. After this field is written, the register isn't actually updated until the corresponding status phase of the control write transfer completes successfully. This feature allows the firmware to write this field as soon as the Setup packet is received, rather than waiting for a successful status phase. Refer to sections 9.2.6.3 and 9.4.6 of the USB 2.0 specification.	Yes	Yes	0

8.5.6 (Address 31h; USBDIAG) USB Diagnostic Register

Bits	Description	Read	Write	Default Value
7:5	**Reserved**	Yes	No	0
4	**Fast Times**. When this bit is set, the frame counter operates at a fast speed for factory chip testing purposes only.	Yes	Yes	0
3	**Reserved.**	Yes	No	0
2	**Force Receive Error.** When this bit is set, an error is forced on the next received data packet. As a result, the packet will not be acknowledged. This bit is automatically cleared at the end of the next packet.	Yes	Yes/Set	0
1	**Prevent Transmit Bit-Stuff.** When this bit is set, normal bit-stuffing is suppressed during the next transmitted data packet. This will cause a bit-stuffing error when six or more consecutive bits of '1' are in the data stream. This bit is automatically cleared at the end of the next packet.	Yes	Yes/Set	0
0	**Force Transmit CRC Error.** When this bit is 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. This bit is automatically cleared at the end of the next packet.	Yes	Yes/Set	0

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

8.5.7 (Address 32h; USBTEST) USB Test Modes

Bits	Description	Read	Write	Default Value
7:3	**Reserved.**	Yes	No	0
2:0	**Test Mode Select.** See sections 7.1.20 and 9.4.9 of the USB 2.0 specification. **Value** **Test** 000 Normal Operation 001 Test_J 010 Test_K 011 Test_SE0_NAK 100 Test_Packet 101 Test_Force_Enable 110 Reserved 111 Reserved	Yes	Yes	0

8.5.8 (Address 33h; XCVRDIAG) Transceiver Diagnostic Register

Bits	Description	Read	Write	Default Value
7:6	**Linestate.** This field indicates the state of the DM (Linestate[1]) and DP (Linestate[0]) USB data signals. [DM,DP] Description 00 SE0 (Single-ended zero) 01 "J" state (idle) 10 "K" state (resume) 11 SE1 (Single-ended one)	Yes	No	-
5:4	**Opmode.** This field indicates the operational state of the transceiver. Value Description 00 normal operation 01 non-driving 10 disable bit stuffing and NRZI endcoding 11 reserved	Yes	No	--
3	**Force High Speed.** When this bit is high, the transceiver is forced into high-speed mode (480 Mbps).	Yes	Yes	0
2	**Force Full Speed.** When this bit is high, the transceiver is forced into full-speed mode (12 Mbps).	Yes	Yes	0
1:0	**Reserved.**	Yes	No	-

8.5.9 (Address 34h; VIRTOUT0) Virtual OUT 0

Bits	Description	Read	Write	Default Value
7:1	**Virtual OUT Interrupts.** These bits are set when the *Virtual Endpoint Enable* bit is set, and an OUT token is received by a virtual endpoint 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 a 1 to a bit clears that bit.	Yes	Yes/Clr	0
0	**Reserved.**	Yes	No	0

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

8.5.10 (Address 35h; VIRTOUT1) Virtual OUT 1

Bits	Description	Read	Write	Default Value
7:0	**Virtual OUT Interrupts.** These bits are set when the *Virtual Endpoint Enable* bit is set, and an OUT token is received by a virtual endpoint 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 a 1 to a bit clears that bit.	Yes	Yes/Clr	0

8.5.11 (Address 36h; VIRTIN0) Virtual IN 0

Bits	Description	Read	Write	Default Value
7:1	**Virtual IN Interrupts.** These bits are set when the *Virtual Endpoint Enable* bit is set, and an IN token is received by a virtual endpoint 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 a 1 to a bit clears that bit.	Yes	Yes/Clr	0
0	**Reserved.**	Yes	No	0

8.5.12 (Address 37h; VIRTIN1) Virtual IN 1

Bits	Description	Read	Write	Default Value
7:0	**Virtual IN Interrupts.** These bits are set when the *Virtual Endpoint Enable* bit is set, and an IN token is received by a virtual endpoint 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 a 1 to a bit clears that bit.	Yes	Yes/Clr	0

8.5.13 (Address 40h; SETUP0) Setup Byte 0

Bits	Description	Read	Write	Default Value
7:0	**Setup Byte 0.** This register provides byte 0 of the last setup packet received. For a Standard Device Request, the following bmRequestType information is returned. Refer to section 9.3 of the USB 2.0 specification. Bit Description 7 Direction: 0 = host to device; 1 = device to host 6:5 Type: 0 = Standard, 1 = Class, 2 = Vendor, 3 = Reserved 4:0 Recipient: 0 = Device, 1 = Interface, 2 = Endpoint, 3 = Other, 4-31 = Reserved	Yes	No	0

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

8.5.14 (Address 41h; SETUP1) Setup Byte 1

Bits	Description	Read	Write	Default Value
7:0	**Setup Byte 1.** This register provides byte 1 of the last setup packet received. For a Standard Device Request, the following bRequest Code information is returned. Refer to section 9.4 of the USB 2.0 specification. **Code** **Description** 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	0

8.5.15 (Address 42h; SETUP2) Setup Byte 2

Bits	Description	Read	Write	Default Value
7:0	**Setup Byte 2.** This register 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 section 9.3.3 of the USB 2.0 specification.	Yes	No	0

8.5.16 (Address 43h; SETUP3) Setup Byte 3

Bits	Description	Read	Write	Default Value
7:0	**Setup Byte 3.** This register 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 section 9.3.3 of the USB 2.0 specification.	Yes	No	0

8.5.17 (Address 44h; SETUP4) Setup Byte 4

Bits	Description	Read	Write	Default Value
7:0	**Setup Byte 4.** This register 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 section 9.3.4 of the USB 2.0 specification.	Yes	No	0

8.5.18 (Address 45h; SETUP5) Setup Byte 5

Bits	Description	Read	Write	Default Value
7:0	**Setup Byte 5.** This register 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 section 9.3.4 of the USB 2.0 specification.	Yes	No	0

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

8.5.19 (Address 46h; SETUP6) Setup Byte 6

Bits	Description	Read	Write	Default Value
7:0	**Setup Byte 6.** This register 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 section 9.3.3 of the USB 2.0 specification.	Yes	No	0

8.5.20 (Address 47h; SETUP7) Setup Byte 7

Bits	Description	Read	Write	Default Value
7:0	**Setup Byte 7.** This register 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 section 9.3.3 of the USB 2.0 specification.	Yes	No	0

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

8.6 Endpoint Registers

There are 4 sets of endpoint registers, one for each endpoint. To access an endpoint, set the *Page Select* field of the **PAGESEL** register to the desired endpoint, then read or write to an endpoint register as defined below. Status bits associated with an endpoint packet buffer (Buffer Full, Buffer Empty, etc), are only valid for the currently visible buffer. The currently visible buffer is the one that is currently being written to or read from by the local bus.

8.6.1 (Address 05h; EP_DATA) Endpoint Data

Note: If DMA Request is enabled, then this register accesses the endpoint buffer selected by *DMA Endpoint Select*, rather than *Page Select.*

Bits	Description	Read	Write	Default Value
15:8	**Endpoint Data (High Order byte).** When operating with a bus width of 16 bits, bits [15:8] of this register provide the high order byte.	Yes	Yes	0
7:0	**Endpoint Data (Low Order byte).** When operating with a bus width of 8 bits, bits [7:0] of this register provide the data for the buffer transaction (read or write). When operating with a bus width of 16 bits, bits [7:0] of this register provide the low order byte.	Yes	Yes	0

8.6.2 (Address 06h; EP_STAT0) Endpoint Status Register (low byte)

Note 1: 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*.
Note 2: The *Buffer Full* and *Buffer Empty* bits take up to 100 nsec to become valid after an endpoint buffer is written or read.

Bits	Description	Read	Write	Default Value
7	**Buffer Full.** This bit is 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 side for writing (no space to write). For an OUT endpoint, there is a buffer available on the local side, and there are MaxPkt bytes available to read (entire packet is available for reading).	Yes	No	0
6	**Buffer Empty.** For an IN endpoint, a buffer is available to the local side for writing up to MaxPkt bytes. This bit is set when the endpoint buffer is empty. For an OUT endpoint, the currently selected buffer has a count of 0, or no buffer is available on the local side (nothing to read).	Yes	No	1
5	**NAK OUT Packets.** This bit is set when a short data packet is received from the host by this endpoint, and the *NAK OUT Packets Mode* bit of the **EP_RSPSET** register is set. Writing a 1 clears this status bit. If this bit is set and another OUT token is received, a NAK is returned to the host if another OUT packet is sent to this endpoint. This bit can also be controlled by the **EP_RSPCLR** and **EP_RSPSET** registers.	Yes	Yes/CLR	0
4	**Short Packet Transferred Interrupt.** This bit is set when the length of the last packet was less than the Maximum Packet Size (**EP_MAXPKT**). Writing a 1 clears this bit.	Yes	Yes/CLR	0
3	**Data Packet Received Interrupt.** This bit is set when a data packet is received from the host by this endpoint. Writing a 1 clears this bit.	Yes	Yes/CLR	0
2	**Data Packet Transmitted Interrupt.** This bit is set when a data packet is transmitted from the endpoint to the host. Writing a 1 clears this bit.	Yes	Yes/CLR	0
1	**Data OUT Token Interrupt.** This bit is set when a Data OUT token has been received from the host. This bit is also set by PING tokens (in high-speed only). Writing a 1 clears this bit.	Yes	Yes/CLR	0
0	**Data IN Token Interrupt.** This bit is set when a Data IN token has been received from the host. Writing a 1 clears this bit.	Yes	Yes/CLR	0

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

8.6.3 (Address 07h; EP_STAT1) -- Endpoint Status Register (high byte)

Bits	Description	Read	Write	Default Value
7	**Buffer Flush.** Writing a 1 to this bit causes the packet buffer to be flushed and the corresponding **EP_AVAIL** register to be cleared. This bit is self-clearing. This bit should always be written after an endpoint configuration (direction, address, etc.) has been changed. This bit should not be asserted during a split-mode DMA if *Page Select* is selecting another endpoint.	0	Yes/Flush	0
6	**Local OUT ZLP.** When set, this bit indicates that the current local buffer contains a zero length packet.	Yes	No	0
5	**USB STALL Sent.** The last USB packet could not be accepted or provided because the endpoint was stalled, and was acknowledged with a STALL. Writing a 1 clears this bit.	Yes	Yes/CLR	0
4	**USB IN NAK Sent.** The last USB IN packet could not be provided, and was acknowledged with a NAK. Writing a 1 clears this bit.	Yes	Yes/CLR	0
3	**USB IN ACK Rcvd.** The last USB IN data packet transferred was successfully acknowledged with an ACK from the host. Writing a 1 clears this bit.	Yes	Yes/CLR	0
2	**USB OUT NAK Sent.** The last USB OUT data packet could not be accepted, and was acknowledged with a NAK to the host. Writing a 1 clears this bit.	Yes	Yes/CLR	0
1	**USB OUT ACK Sent.** The last USB OUT data packet transferred was successfully acknowledged with an ACK to the host. Writing a 1 clears this bit.	Yes	Yes/CLR	0
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 will expect the same packet to be retransmitted in response to the next IN token. For an OUT endpoint, the last USB packet received had a CRC or bit-stuffing error, and was not acknowledged by the NET2272. The Host PC will retransmit the packet. Writing a 1 clears this bit.	Yes	Yes/CLR	0

8.6.4 (Address 08h; EP_TRANSFER0) Transfer Count Register (Byte 0)

Bits	Description	Read	Write	Default Value
7:0	**EP_TRANSFER[7:0].** For IN endpoints, this field determines the total number of bytes to be sent to the host. This field should be written before any packet data is written to the buffer. When the count reaches zero, any remaining data in the buffer is validated. Note that validation takes about 100 nsec. Writing zero to **EP_TRANSFER0** when **EP_TRANSFER1** and **EP_TRANSFER2** have a value of 0 validates the contents of this IN endpoint buffer regardless of the state of the *Auto Validate* bit**;** if the buffer is empty, writing zero to **EP_TRANSFER0** validates a Zero Length Packet. For OUT endpoints, this 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 will only increment by 1.	Yes	Yes	0

8.6.5 (Address 09h; EP_TRANSFER1) Transfer Count Register (Byte 1)

Bits	Description	Read	Write	Default Value
7:0	**EP_TRANSFER[15:8].**	Yes	Yes	0

8.6.6 (Address 0Ah; EP_TRANSFER2) Transfer Count Register (Byte 2)

Bits	Description	Read	Write	Default Value
7:0	**EP_TRANSFER[23:16]**.	Yes	Yes	0

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

8.6.7 (Address 0Bh; EP_IRQENB) Endpoint Interrupt Enable Register

Bits	Description	Read	Write	Default Value
7:5	**Reserved.**	0	No	0
4	**Short Packet Transferred Interrupt Enable.** When set, this bit enables a local interrupt to be set when a short data packet has been transferred to/from the host.	Yes	Yes	0
3	**Data Packet Received Interrupt Enable.** When set, this bit enables a local interrupt to be set when a data packet has been received from the host.	Yes	Yes	0
2	**Data Packet Transmitted Interrupt Enable.** When set, this bit enables a local interrupt to be set when a data packet has been transmitted to the host.	Yes	Yes	0
1	**Data OUT Token Interrupt Enable.** When set, this bit enables a local interrupt to be set when a Data OUT token has been received from the host.	Yes	Yes	0
0	**Data IN Token Interrupt Enable.** When set, this bit enables a local interrupt to be set when a Data IN token has been received from the host.	Yes	Yes	0

8.6.8 (Address 0Ch: EP_AVAIL0) Endpoint Available Count (low byte)

Note: If DMA Request is enabled, then the value in this register corresponds to the endpoint buffer selected by *DMA Endpoint Select*, rather than *Page Select*.

Bits	Description	Read	Write	Default Value
7:0	**EP_AVAIL[7:0].** For an OUT endpoint, this register returns the number of valid bytes in the endpoint packet buffer. Values range from 0 (empty) to 1024 (full). For an IN endpoint, this register returns the number of empty bytes in the packet buffer. Values range from 0 (full) to 1024 (empty). This field is updated either after 2 bytes have been written to the buffer, or when the buffer has been validated. If only the low byte of this field is read, the entire 11-bit field is frozen until the upper byte is read.	Yes	No	0

8.6.9 (Address 0Dh: EP_AVAIL1) Endpoint Available Count (high byte)

Bits	Description	Read	Write	Default Value
7:3	**Reserved.**	Yes	No	0
2:0	**EP_AVAIL[10:8].**	Yes	No	0

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

8.6.10 (Address 0Eh; EP_RSPCLR) Endpoint Response Register Clear

Note: Writing a 1 to bits 7:0 clears the corresponding register bits.

Bits	Description	Read	Write	Default Value
7	**Alt NAK OUT Packets.** This bit is set when a short data packet is received from the host by this endpoint, and the *NAK OUT Packets Mode* bit is set. If this bit is set and another OUT token is received, a NAK is returned to the host if another OUT packet is sent to this endpoint. This bit can also be cleared by a bit in the **EP_STAT0** register.	Yes	Yes/Clr	0
6	**Hide Status Phase.** When set, the *DATA Packet Received* and *Data Packet Transmitted* interrupts for status phase packets are not set.	Yes	Yes/Clr	0
5	**Auto Validate.** When set, this bit allows automatic validation of maximum length packets. Automatic validation means that if there are **EP_MAXPKT** bytes in the endpoint buffer, 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 mode of operation for endpoint transactions and is the default state for this bit. When this bit is clear, packets must be manually validated. Writing zero to **EP_TRANSFER0** when **EP_TRANSFER1** and **EP_TRANSFER2** have a value of 0 validates the contents of this IN endpoint buffer regardless of the state of the *Auto Validate* bit**;** if the buffer is empty, writing zero to **EP_TRANSFER0** validates a Zero Length Packet.	Yes	Yes/Clr	1
4	**Interrupt Mode.** This bit is only used for INTERRUPT endpoints. For normal interrupt data, this bit should be set to zero and standard data toggle protocol is followed. When this interrupt endpoint is used for isochronous rate feedback information, this bit should be set high. In this mode the data toggle bit is changed after each packet is sent to the host without regard to handshaking. No packet retries are performed in the rate feedback mode.	Yes	Yes/Clr	0
3	**Control Status Phase Handshake.** This bit is only used for endpoint 0. This bit is automatically set when a setup packet is detected. While the bit is set, a control status phase will be acknowledged with a NAK. Once cleared, the proper response will be returned to the host (ACK for Control Reads and zero-length packets for Control Writes).	Yes	Yes/Clr	0
2	**NAK OUT Packets Mode.** This bit is only used for OUT endpoints. When *NAK OUT Packets Mode* is true, the *NAK OUT Packets* bit is set whenever a short packet is received by this endpoint.	Yes	Yes/Clr	1
1	**Endpoint Toggle.** This bit is used to clear the endpoint data toggle bit. Reading this bit returns the current state of the endpoint data toggle bit. Under normal operation, the toggle bit is controlled automatically, so the local CPU does not need to use this bit.	Yes	Yes/Clr	0
0	**Endpoint Halt.** This bit is used to clear the endpoint stall bit. When an Endpoint Set Feature Standard Request to the halt bit is detected by the local CPU, it must write a 1 to this bit. Reading this bit returns the current state of the endpoint halt bit. For Endpoint 0, the halt bit is automatically cleared when another Setup packet is received.	Yes	Yes/Clr	0

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

8.6.11 (Address 0Fh; EP_RSPSET) Endpoint Response Register Set

Note: Writing a 1 to bits 7:0 sets the corresponding register bits.

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

8.6.12 (Address 28h; EP_MAXPKT0) Max Packet Size (low byte)

Bits	Description	Read	Write	Default Value
7:0	**EP_MAXPKT[7:0].** This field determines the Endpoint Maximum Packet Size.	Yes	Yes	EP0 = 64 EPA = 512 EPB = 512 EPC = 512

8.6.13 (Address 29h; EP_MAXPKT1) Max Packet Size (high byte)

Bits	Description	Read	Write	Default Value
7:5	**Reserved.**	Yes	No	0
4:3	**Additional Transaction Opportunities.** This field determines the number of additional transaction opportunities per microframe for high-speed isochronous and interrupt endpoints. 00 = None (1 transaction per microframe) 01 = 1 additional (2 per microframe) 10 = 2 additional (3 per microframe) 11 = Reserved	Yes	Yes	0
2:0	**EP_MAXPKT[10:8].** This field determines the Endpoint Maximum Packet Size.	Yes	Yes	EP0 = 64 EPA = 512 EPB = 512 EPC = 512

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

8.6.14 (Address 2Ah; EP_CFG) Endpoint Configuration Register

NOTE: For Endpoint 0, all fields in this register, except *Endpoint Direction*, are assigned to fixed values, and are **RESERVED**.

Bits	Description	Read	Write	Default Value
7	**Endpoint Enable.** When set, this bit enables this endpoint. This bit has no effect on Endpoint 0, which is always enabled. When this bit is cleared, it will not read back as a zero until all pending USB transactions on the endpoint have completed.	Yes	Yes	0
6:5	**Endpoint Type.** This field selects the type of this endpoint. Endpoint 0 is forced to a Control type. Value Description 0 **Reserved** 1 Isochronous 2 Bulk 3 Interrupt	Yes	Yes	0
4	**Endpoint Direction.** This bit selects the direction of the endpoint selected by Page Select. EP_DIR = 0 means Host OUT to Device, while EP_DIR = 1 means Host IN from Device. Endpoint 0 is bi-directional, and uses this bit for a test mode. When set, endpoint packet buffers can be read back for diagnostics. Note that a maximum of one OUT and IN endpoint is allowed for each endpoint number. For endpoint 0, this bit is dynamic, and depends on the direction bit in the last Setup packet.	Yes	Yes	0
3:0	**Endpoint Number.** This field selects the number of the endpoint. Valid numbers are 0 to 15. This field has no effect on Endpoint 0, which always has an endpoint number of 0.	Yes	Yes	0

8.6.15 (Address 2Bh: EP_HBW) Endpoint High Bandwidth

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

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

8.6.16 (Address 2Ch: EP_BUFF_STATES) Endpoint Buffer States

Bits	Description	Read	Write	Default Value
7:4	**Reserved.**	Yes	No	0
3:2	**Buffer B State.** This field provides the current state of the endpoint buffer B. Value State 00 Buff_Free; buffer is empty, free to assign 01 Buff_Valid; buffer has valid packet, waiting to move to local or USB side 10 Buff_Lcl; buffer is assigned to the local side 11 Buff_Usb; buffer is assigned to the USB side	Yes	No	--
1:0	**Buffer A State.** This field provides the current state of the endpoint buffer A. Value State 00 Buff_Free; buffer is empty, free to assign 01 Buff_Valid; buffer has valid packet, waiting to move to local or USB side 10 Buff_Lcl; buffer is assigned to the local side 11 Buff_Usb; buffer is assigned to the USB side	Yes	No	--

IN packets move: Lcl -> Valid -> Usb -> Free
OUT packets move: 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 to the Buff_Free state.

8.7 Register Changes from Net2270

- Add *Virtualized Endpoint Interrupt Enable* to **IRQENB0**[4].
- Add *Virtualized Endpoint Interrupt Status* to **IRQSTAT0**[4].
- Add new **LOCCTL1** register. Add *DMA Mode* bit to **LOCCTL1**[1:0].
- Add *Virtual Endpoint Enable* bit to **USBCTL**[4].
- Add Virtual OUT Interrupt 0 register (**VIRTOUT0)**.
- Add Virtual OUT Interrupt 1 register (**VIRTOUT1)**.
- Add Virtual IN Interrupt 0 register (**VIRTIN0)**.
- Add Virtual IN Interrupt 1 register (**VIRTIN1)**.
- Add *Local OUT ZLP* status bit to **EP_STAT1[6].**
- Add new **EP_HBW** register.
- Add new **EP_BUFF_STATES** register.
- Remove *Force Bi-directional to Inputs* bit from **USBDIAG** register.

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

9 USB Standard Device Requests

Standard device requests must be supported by Endpoint 0. See also chapter 9, USB specification. The local bus CPU decodes the setup packets for Endpoint 0 and generates a response based on the following tables

Table 9-1: Standard Request Codes

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

Table 9-2. Descriptor Types

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

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

9.1 Control 'Read' Transfers

9.1.1 Get Device Status

Offset	Number of Bytes	Description	Suggested Value
0	2	bits 15:2 = Reserved bit 1 = Device Remote Wakeup enabled bit 0 = Power supply is good in Self-Powered mode.	Determined by local CPU

9.1.2 Get Interface Status

Offset	Number of Bytes	Description	Suggested Value
0	2	bits 15:0 = Reserved	0000h

9.1.3 Get Endpoint Status

Offset	Number of Bytes	Description	Suggested Value
0	2	bits 15:1 = Reserved bit 0 = Endpoint is halted	Determined by local CPU

9.1.4 Get Device Descriptor (18 Bytes)

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

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

9.1.5 Get Device Qualifier (10 Bytes)

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

9.1.6 Get Other_Speed_Configuration Descriptor

The structure of the other_speed_cofiguration is identical to a configuration descriptor, except that the bDescriptorType is 7 instead of 2.

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

9.1.7 Get Configuration Descriptor

The NET2272 can support a variety of configurations, interfaces, and endpoints, each of which is defined by the descriptor data returned to the host. The local CPU has the responsibility of providing this data to the NET2272 when the host requests it.

This example has one configuration, and two interfaces. The first interface defines one Bulk OUT endpoint at address 1 with maximum packet size of 512 and one Interrupt IN endpoint at address 82h (endpoint number = 2) with a maximum packet size of 8. The second interface defines one Bulk OUT endpoint at address 3 with maximum packet size of 512.

Note that all interface and endpoint descriptors are returned in response to a Get Configuration Descriptor request, and for this example, 48 bytes are returned.

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

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

Get Configuration Descriptor (continued)

Offset	Number of Bytes	Description	Suggested Value
Bulk OUT Endpoint 1 Descriptor			
0	1	Size of this descriptor	07h
1	1	Descriptor Type (endpoint)	05h
2	1	Endpoint Address bit 7 = direction (1 = IN, 0 = OUT) bits 6:4 = reserved bits 3:0 = endpoint number	01h
3	1	Endpoint Attributes bits 7:2 = reserved bits 1:0 00 = Control 01 = Isochronous 10 = Bulk 11 = Interrupt	02h
4	2	Maximum packet size of this endpoint	0200h
6	1	Maximum NAK rate of the endpoint.	Determined by Local CPU
Interrupt IN Endpoint 2 Descriptor			
0	1	Size of this descriptor	07h
1	1	Descriptor Type (endpoint)	05h
2	1	Endpoint Address bit 7 = direction (1 = IN, 0 = OUT) bits 6:4 = reserved bits 3:0 = endpoint number	82h
3	1	Endpoint Attributes bits 7:2 = reserved bits 1:0 00 = Control 01 = Isochronous 10 = Bulk 11 = Interrupt	03h
4	2	bits 10:0 = Maximum packet size of this endpoint. (Determined by the **EP_MAXPKT** registers). bits 12:11 = Number of additional transaction opportunities per microframe: 00 = None (1 transaction per microframe) 01 = 1 additional (2 per microframe) 10 = 2 additional (3 per microframe) 11 = Reserved Bits 15:13 = reserved	0008h
6	1	Interval for polling endpoint	Determined by Local CPU

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

Get Configuration Descriptor (continued)

Offset	Number of Bytes	Description	Suggested Value
Interface 1 Descriptor			
0	1	Size of this descriptor in bytes	09h
1	1	Type (interface)	04h
2	1	Number of this interface	01h
3	1	Alternate Interface	00h
4	1	Number of endpoints used by this interface (excluding Endpoint 0)	01h
5	1	Class Code	FFh
6	1	Sub Class Code	00h
7	1	Device Protocol	00h
8	1	Index of string descriptor describing this interface	00h
Bulk OUT Endpoint 3 Descriptor			
0	1	Size of this descriptor	07h
1	1	Descriptor Type (endpoint)	05h
2	1	Endpoint Address bit 7 = direction (1 = IN, 0 = OUT) bits 6:4 = reserved bits 3:0 = endpoint number	03h
3	1	Endpoint Attributes bits 7:2 = reserved bits 1:0 00 = Control 01 = Isochronous 10 = Bulk 11 = Interrupt	02h
4	2	Maximum packet size of this endpoint for bulk mode	0200h
6	1	Maximum NAK rate of the endpoint.	Determined by Local CPU

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

9.1.8 Get String Descriptor 0

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

9.1.9 Get String Descriptor 1

Offset	Number of Bytes	Description	Suggested Value
0	1	Size of this descriptor in bytes	26h
1	1	Descriptor type (string)	03h
2	36	Manufacturer Descriptor. The text string is encoded in UNICODE.	"NetChip Technology"

9.1.10 Get String Descriptor 2

Offset	Number of Bytes	Description	Suggested Value
0	1	Size of this descriptor in bytes	42h
1	1	Descriptor type (string)	03h
2	64	Product Descriptor. The text string is encoded in UNICODE.	"NET2272 USB Peripheral Controller"

9.1.11 Get String Descriptor 3

Offset	Number of Bytes	Description	Suggested Value
0	1	Size of this descriptor in bytes	0Ah
1	1	Descriptor type (string)	03h
2	8	Serial Number Descriptor. The text string is encoded in UNICODE.	"1001"

9.1.12 Get Configuration

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

9.1.13 Get Interface

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

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

9.2 Control 'Write' Transfers

9.2.1 Set Address

Note: The local CPU must write the new device address into the **USBADDR** configuration register

Offset	Number of Bytes	Description	Suggested Value
--	0	Sets USB address of device wValue = device address, wIndex = 0, wLength = 0	--

9.2.2 Set Configuration

Note: The local CPU must keep track of the configuration value.

Offset	Number of Bytes	Description	Suggested Value
--	0	Sets the device configuration wValue = Configuration value, wIndex = 0, wLength = 0	--

9.2.3 Set Interface

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

Offset	Number of Bytes	Description	Suggested Value
--	0	Selects alternate setting for specified interface wValue = Alternate setting, wIndex = specified interface, wLength = 0	--

9.2.4 Device Clear Feature

Note: The local CPU must keep track of the state of the Device Remote Wakeup enable.

Offset	Number of Bytes	Description	Suggested Value
--	0	Clear the selected device feature wValue = feature selector, wIndex = 0, wLength = 0 FS = 1 → Device Remote Wakeup (disable)	--

9.2.5 Device Set Feature

Offset	Number of Bytes	Description	Suggested Value
--	0	Set the selected device feature wValue = feature selector, wLength = 0 FS = 1 → Device Remote Wakeup (enable), wIndex = 0 FS = 2 → Test Mode, wIndex = specifies test mode	--

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

9.2.6 Endpoint Clear Feature

Note: The local CPU must clear the endpoint halt bit by writing to the *Endpoint Halt* bit in the **EP_RSPCLR** register.

Offset	Number of Bytes	Description	Suggested Value
--	0	Clear the selected endpoint feature wValue = feature selector, wIndex = endpoint number, wLength = 0 FS = 0 → Endpoint halt (clears halt bit)	--

9.2.7 Endpoint Set Feature

Note: The local CPU must set the endpoint halt bit by writing to the *Endpoint Halt* bit in the **EP_RSPSET** register.

Offset	Number of Bytes	Description	Suggested Value
--	0	Set the selected endpoint feature wValue = feature selector, wIndex = endpoint number, wLength = 0 FS = 0 → Endpoint halt (sets halt bit)	--

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

10 Electrical Specifications

10.1 Absolute Maximum Ratings

Conditions that exceed the Absolute Maximum limits may destroy the device.

Symbol	Parameter	Conditions	Min	Max	Unit
V_{DDC}, V_{DD25}, P_{VDD}, A_{VDD}	2.5V Supply Voltages	With Respect to Ground	-0.5	3.6	V
V_{DDIO}, V_{DD33}	3.3V Supply Voltages	With Respect to Ground	-0.5	4.6	V
V_I	DC input voltage	3.3 V buffer	-0.5	4.6	V
		5 V Tolerant buffer	-0.5	6.6	V
I_{OUT}	DC Output Current, per pin	3mA Buffer	-10	10	mA
		6mA Buffer	-20	20	mA
		12mA Buffer	-40	40	mA
T_{STG}	Storage Temperature	No bias	-65	150	° C
T_{AMB}	Ambient temperature	Under bias	-40	85	° C
V_{ESD}	ESD Rating	R = 1.5K, C = 100pF		2	KV

10.2 Recommended Operating Conditions

Conditions that exceed the Operating limits may cause the device to function incorrectly.

Symbol	Parameter	Conditions	Min	Max	Unit
V_{DDC}, V_{DD25}, P_{VDD}, A_{VDD}	2.5V Supply Voltages		2.2	2.6	V
V_{DDIO}, V_{DD33}	3.3V Supply Voltages		3.2	3.5	V
V_N	Negative trigger voltage	3.3 V buffer	0.8	1.7	V
		5 V tolerant buffer	0.8	1.7	V
V_P	Positive trigger voltage	3.3 V buffer	1.3	2.4	V
		5 V tolerant buffer	1.3	2.4	V
V_{IL}	Low Level Input Voltage	3.3 V buffer	0	0.7	V
		5 V tolerant buffer	0	0.8	V
V_{IH}	High Level Input Voltage	3.3 V buffer	0.5* V_{DDIO}	V_{DDIO}	V
		5 V 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		0	70	° C
t_R	Input rise times	Normal input	0	200	ns
t_F	Input fall time	Normal input	0	200	ns
t_R	Input rise times	Schmitt input	0	10	ms
t_F	Input fall time	Schmitt input	0	10	ms

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

10.3 DC Specifications

10.3.1 Core DC Specifications

Operating Conditions: V_{DDC}: 2.2-2.6V, V_{DDIO}: 3.2-3.5V, T_A = 0°C to 70°C
All typical values are at V_{DDC} = 2.5V, V_{DDIO} = 3.3V and T_A = 25°C

10.3.1.1 Disconnected from USB

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
I_{VDD33}	3.3V Supply Current	V_{DDC} = 3.3V		1.4	1.6	mA
I_{VDD25}	2.5V Supply Current	V_{DDIO} = 2.5V		36.3	40	mA

10.3.1.2 Connected to USB (High-Speed)

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
I_{VDD33}	3.3V Supply Current	V_{DDC} = 3.3V		3.3	3.7	mA
I_{VDD25}	2.5V Supply Current	V_{DDIO} = 2.5V		51.6	58	mA

10.3.1.3 Active (High-Speed)

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
I_{VDD33}	3.3V Supply Current	V_{DDC} = 3.3V		4	4.4	mA
I_{VDD225}	2.5V Supply Current	V_{DDIO} = 2.5V		52.7	58	mA

10.3.1.4 Connected to USB (Full-Speed)

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
I_{VDD33}	3.3V Supply Current	V_{DDC} = 3.3V		2.8	3.1	mA
I_{VDD25}	2.5V Supply Current	V_{DDIO} = 2.5V		35.3	40	mA

10.3.1.5 Active (Full-Speed)

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
I_{VDD33}	3.3V Supply Current	V_{DDC} = 3.3V		6	6.6	mA
I_{VDD25}	2.5V Supply Current	V_{DDIO} = 2.5V		35.9	40	mA

10.3.1.6 Suspended

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
I_{VDD33}	3.3V Supply Current	V_{DDC} = 3.3V		0.1*	10	uA
I_{VDD25}	2.5V Supply Current	V_{DDIO} = 2.5V		0.1	10	uA

* Disconnected from USB. When connected to USB, 200uA is added for the 1.5K pull-up resistor.
16-bit data bus (LD[15:0]) should not float when suspended to prevent leakage current.

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

10.3.2 USB Full Speed DC Specifications

Operating Conditions: V_{DDC}: 2.2-2.6V, V_{DDIO}: 3.2-3.5V, T_A = 0°C to 70°C
All typical values are at V_{DDC} = 2.5V, V_{DDIO} = 3.3V and T_A = 25°C

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 VDI 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 to GND			20	pF

10.3.3 USB High Speed DC Specifications

Operating Conditions: V_{DDC}: 2.2-2.6V, V_{DDIO}: 3.2-3.5V, T_A = 0°C to 70°C
All typical values are at V_{DDC} = 2.5V, V_{DDIO} = 3.3V and T_A = 25°C

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}	Hip-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_{CHIPRJ}	Chirp J level (differential voltage)		700		1100	mV
V_{CHIPRK}	Chirp K level (differential voltage)		-900		-500	mV
C_{IO}	I/O Capacitance	Pin to GND			20	pF

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

10.3.4 Local Bus DC Specifications

Operating Conditions: V_{DDC}: 2.2-2.6V, V_{DDIO}: 3.2-3.5V, T_A = 0°C to 70°C
All typical values are at V_{DDC} = 2.5V, V_{DDIO} = 3.3V and T_A = 25°C

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		0.8	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 Output High Voltage	I_{OUT} = -12mA	2.4			V
V_{OL}	5.0V Tolerant Output Low Voltage	I_{OUT} = 12mA			0.4	V
C_{IN}	Input Capacitance	Pin to GND			10	pF
C_{CLK}	CLK Pin Capacitance	Pin to GND	5		12	pF
C_{IDSEL}	IDSEL Pin Capacitance	Pin to GND			8	pF

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

10.4 AC Specifications

10.4.1 USB Full Speed Port AC Specifications

Operating Conditions: V_{DDC}: 2.2-2.6V, V_{DDIO}: 3.2-3.5V, T_A = 0°C to 70°C
All typical values are at V_{DDC} = 2.5V, V_{DDIO} = 3.3V and T_A = 25°C

Symbol	Parameter	Conditions	Waveform	Min	Typ	Max	Unit
T_{FR}	Rise & Fall Times	C_L = 50 pF,	Figure 8-1	4		20	ns
T_{FF}		Note 16		4		20	
T_{FRFM}	Rise/Fall time matching	(T_{FR}/ T_{FF}), Note 10	Figure 8-1	90		110	%
Z_{DRV}	Driver Output Resistance	Steady State Drive		10		15	Ω
$T_{FDRATHS}$	Full-speed Data Rate			11.994	12	12.006	Mbs
T_{DJ1}	Source Differential Driver Jitter to Next Transition	Notes 7,8,10,12	Figure 8-2	-2	0	2	ns
T_{DJ2}	Source Differential Driver Jitter for Paired Transitions	Notes 7,8,10,12	Figure 8-2	-1	0	1	ns
T_{FDEOP}	Source Jitter for Differential Transition to SE0 Transition	Note 8, 11	Figure 8-3	-2	0	5	ns
T_{JR1}	Receiver Data Jitter Tolerance to Next Transition	Note 8	Figure 8-4	-18.5	0	18.5	ns
T_{JR2}	Receiver Data Jitter Tolerance for Paired Transitions	Note 8	Figure 8-4	-9	0	9	ns
T_{EOPT}	Source SE0 interval of EOP		Figure 8-3	160	167	175	ns
T_{FEOPR}	Receiver SE0 interval of EOP	Note 13	Figure 8-3	82			ns
T_{FST}	Width of SE0 interval during differential transition			14			ns

10.4.2 USB High Speed Port AC Specifications

Operating Conditions: V_{DDC}: 2.2-2.6V, V_{DDIO}: 3.2-3.5V, T_A = 0°C to 70°C
All typical values are at V_{DDC} = 2.5V, V_{DDIO} = 3.3V and T_A = 25°C

Symbol	Parameter	Conditions	Waveform	Min	Typ	Max	Unit
T_{HSR}	Rise & Fall Times	Note 16		500			ps
T_{HSF}				500			
Z_{DRV}	Driver Output Resistance	Steady State Drive		10		15	Ω
T_{HSDRV}	High-speed Data Rate			479.760	480	480.240	Mbs
	Data source jitter	Specified by eye pattern templates					
	Receiver jitter tolerance	Specified by eye pattern templates					

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

10.4.3 USB Full Speed Port AC Waveforms

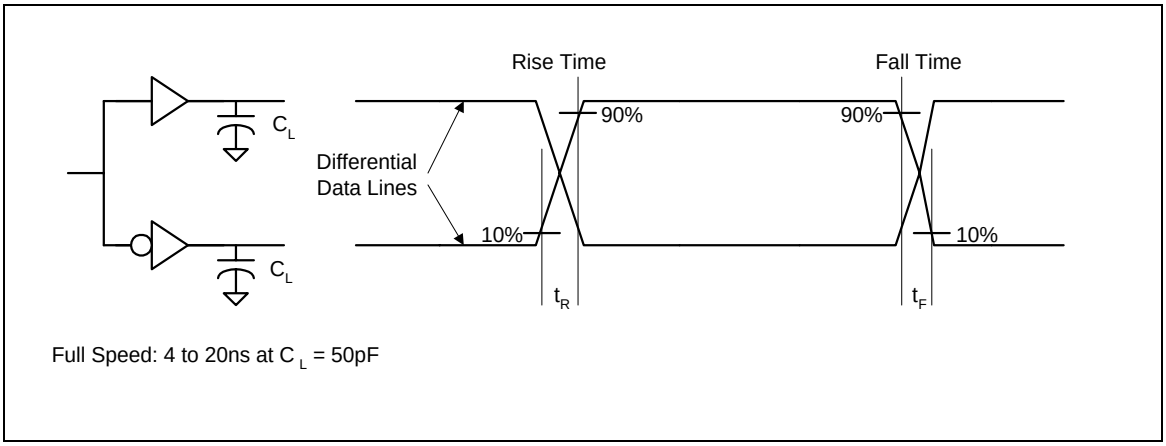


Figure 10-1. Data Signal Rise and Fall Time

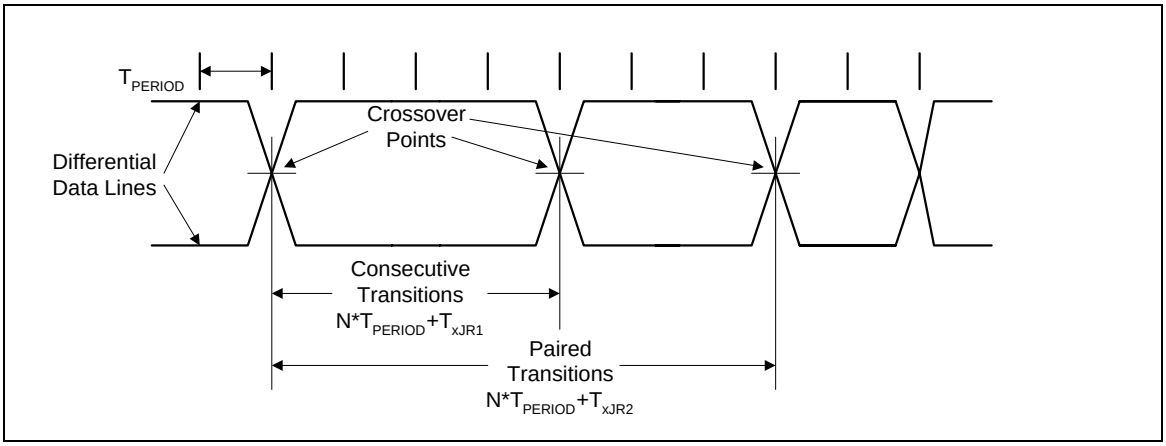


Figure 10-2. Differential Data Jitter

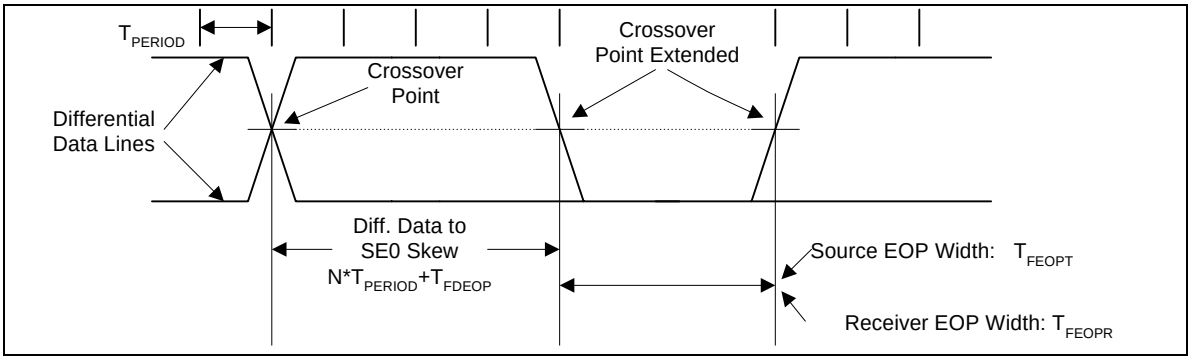


Figure 10-3. Differential to EOP Transition Skew and EOP Width

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

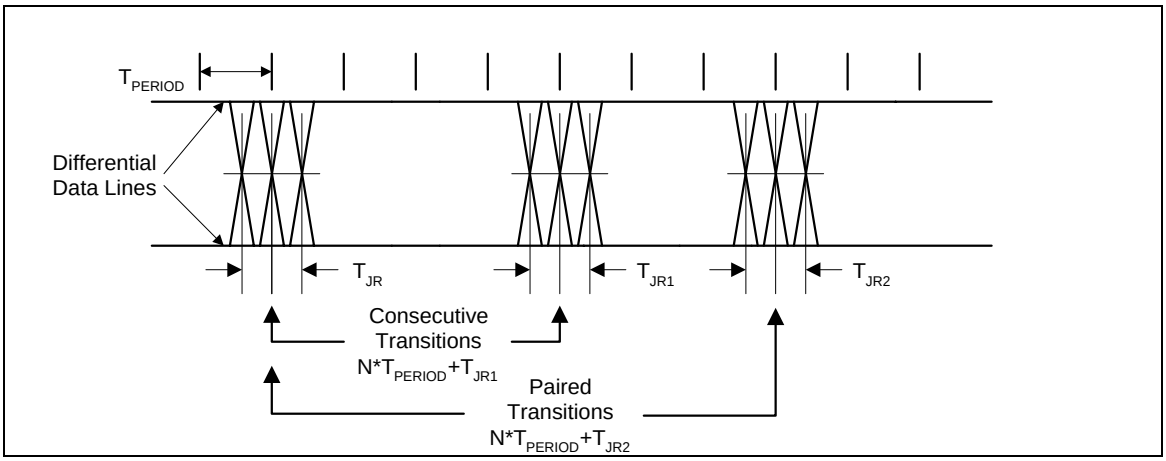


Figure 10-4. Receiver Jitter Tolerance

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

10.4.4 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 1.425KΩ to 3.6V.
6. Measured with RL of 14.25KΩ to GND.
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 V_{BUS} droop 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, GND, 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 remote wakeup.
16. Measured from 10% to 90% of the data signal.

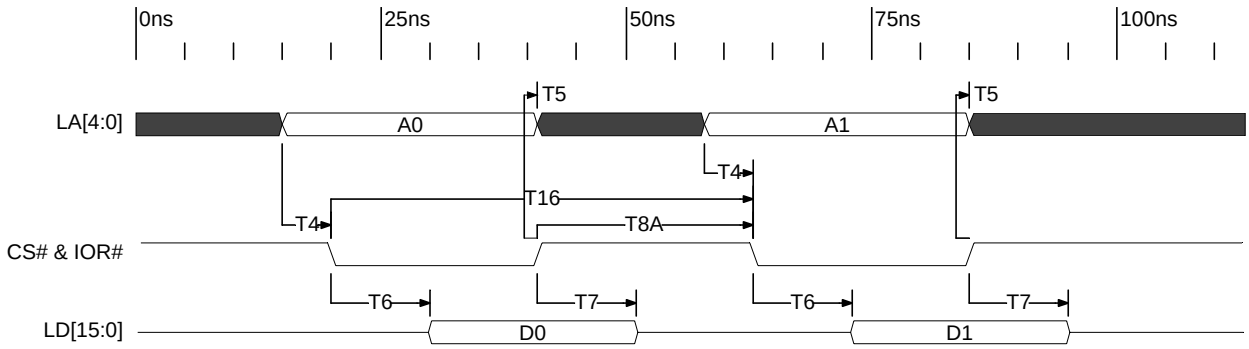
335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

10.4.5 Local Bus Non-Multiplexed Read

Operating Conditions: V_{DDC}: 2.2-2.6V, V_{DDIO}: 3.2-3.5V, T_A = 0°C to 70°C, Output Load = 25pF

NAME	DESCRIPTION	MIN	MAX	UNIT
T4	Address setup to read enable	-1		ns
T5	Address hold from end of read enable	-2		ns
T6	Data access time from LA valid or read enable asserted (1), whichever is later		18	ns
T7	Data tri-state time from end of read enable	2	11	ns
T8A	Recovery Time to next read (2)	19		ns
T8B	Recovery Time to next write	19		ns
T16	Read Cycle Time	35		ns

(1) Read enable is the occurrence of both CS# and IOR#.
(2) Since reading and writing to **EP_DATA** cause **EP_AVAIL** and **EP_TRANSFER** to change values, it is necessary to increase the recovery time to 37 nsec between a read or write to **EP_DATA** and a read from **EP_AVAIL** or **EP_TRANSFER**.



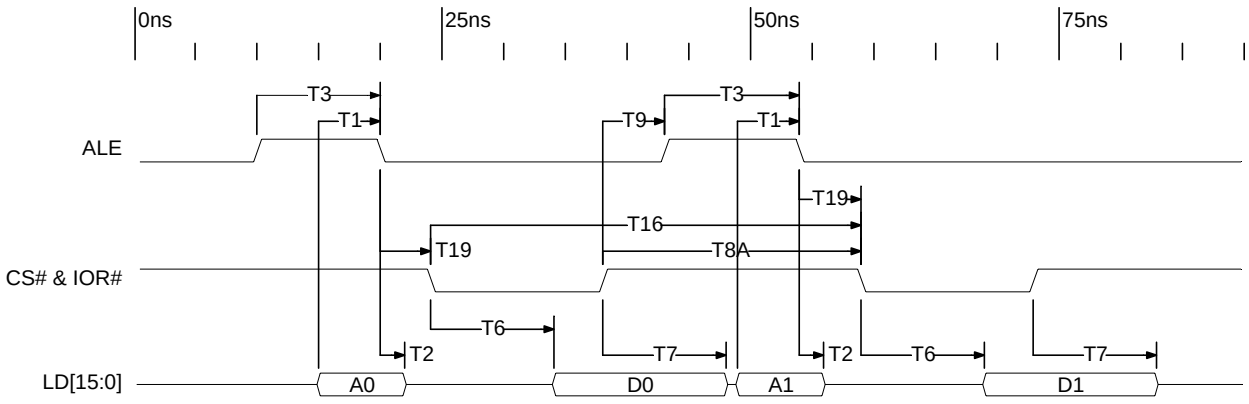
335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

10.4.6 Local Bus Multiplexed Read

Operating Conditions: V_{DDC}: 2.2-2.6V, V_{DDIO}: 3.2-3.5V, T_A = 0°C to 70°C, Output Load = 25pF

NAME	DESCRIPTION	MIN	MAX	UNIT
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
T19	ALE falling edge to read enable	1		ns
T6	Data access time from read enable (1)		18	ns
T7	Data tri-state time from end of read enable	2	11	ns
T8A	Recovery Time to next read (2)	19		ns
T8B	Recovery Time to next write	19		ns
T9	Recovery Time to next ALE	5		ns
T16	Read Cycle Time	35		ns

(1) Read enable is the occurrence of both CS# and IOR#.
(2) Since reading and writing to **EP_DATA** cause **EP_AVAIL** and **EP_TRANSFER** to change values, it is necessary to increase the recovery time to 37 nsec between a read or write to **EP_DATA** and a read from **EP_AVAIL** or **EP_TRANSFER**.



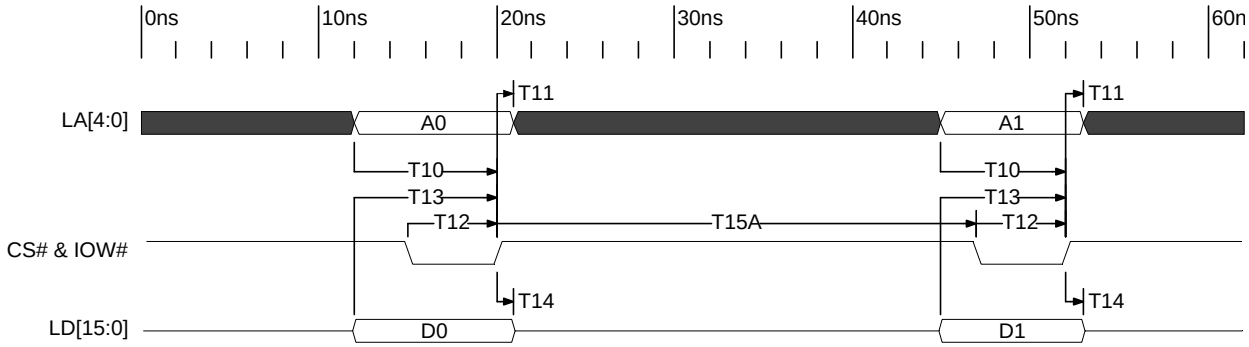
335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

10.4.7 Local Bus Non-Multiplexed Write

Operating Conditions: V_{DDC}: 2.2-2.6V, V_{DDIO}: 3.2-3.5V, T_A = 0°C to 70°C, Output Load = 25pF

NAME	DESCRIPTION	MIN	MAX	UNIT
T10	Address setup to end of write enable	5		ns
T11	Address hold from end of write enable	0		ns
T12	Write enable width (1)	5		ns
T13	Data setup to end of write enable	5		ns
T14	Data hold time from end of write enable	0		ns
T15A	Recovery Time to next write	28		ns
T15B	Recovery Time to next read (2)	52		ns

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



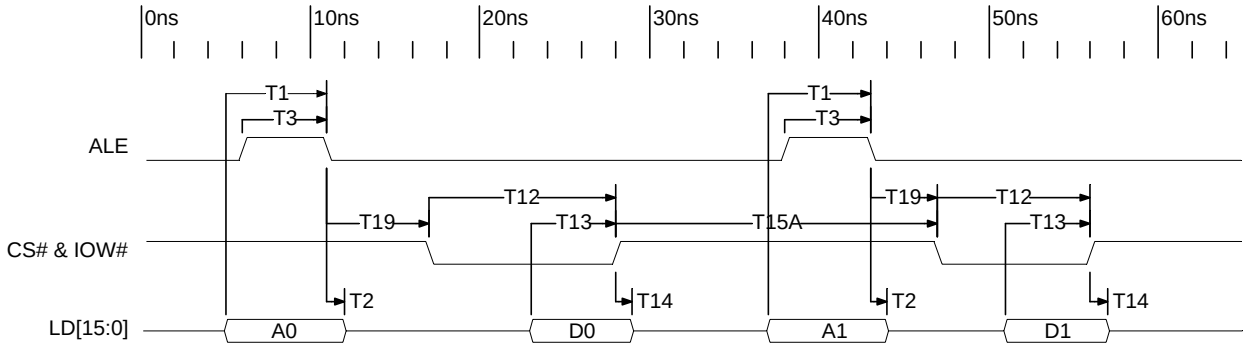
335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

10.4.8 Local Bus Multiplexed Write

Operating Conditions: V_{DDC}: 2.2-2.6V, V_{DDIO}: 3.2-3.5V, T_A = 0°C to 70°C, Output Load = 25pF

NAME	DESCRIPTION	MIN	MAX	UNIT
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
T19	ALE falling edge to write enable	1		ns
T12	Write enable width (1)	5		ns
T13	Data setup to end of write enable	5		ns
T14	Data hold time from end of write enable	0		ns
T15A	Recovery Time to next write	28		ns
T15B	Recovery Time to next read (2)	52		ns

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



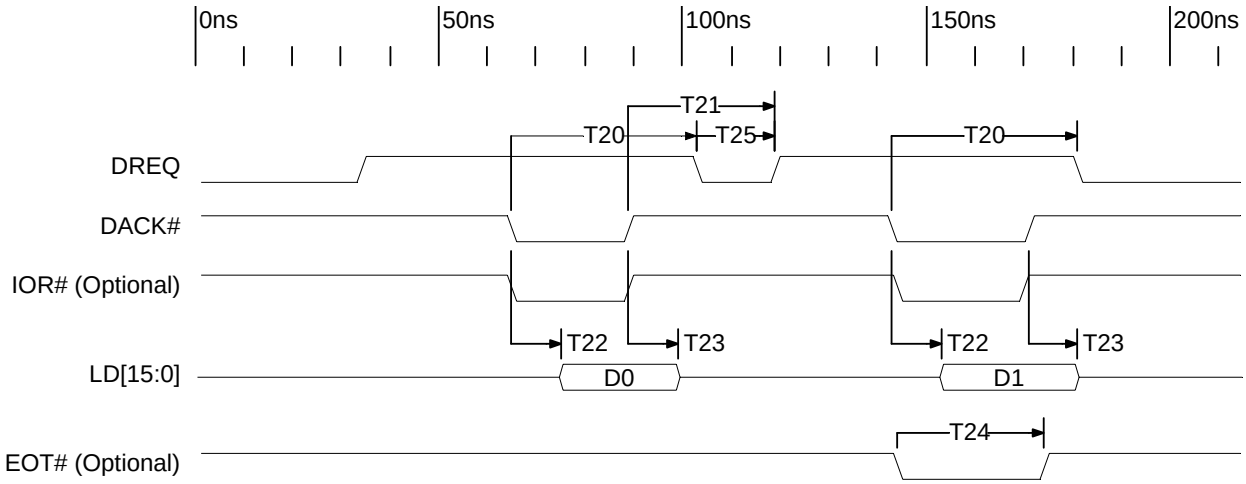
335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

10.4.9 Local Bus DMA Read; Slow Mode

Operating Conditions: V_{DDC}: 2.2-2.6V, V_{DDIO}: 3.2-3.5V, T_A = 0°C to 70°C, Output Load = 25pF

NAME	DESCRIPTION	MIN	TYP	MAX	UNIT
T20	Read enable true to DREQ false (2)	25	50	60	ns
T21	Read enable false to DREQ true	16	50	68	ns
T22	Data access time from read enable (1)			16	ns
T23	Data tri-state time from end of read enable	2		12	ns
T24	Width of EOT# pulse (3)	10			ns
T25	DREQ false to DREQ true	15	25		ns

(1) For non-split DMA mode, read enable is the occurrence of DACK# and optionally, IOR#. For split DMA mode, read enable is the occurrence of DACK# and optionally, DMARD#.
(2) The minimum value is only guaranteed if the *DMA Request Enable* bit in the **DMAREQ** register 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 nsec is required between the de-assertion of DMA read enable and the assertion of an I/O read enable.



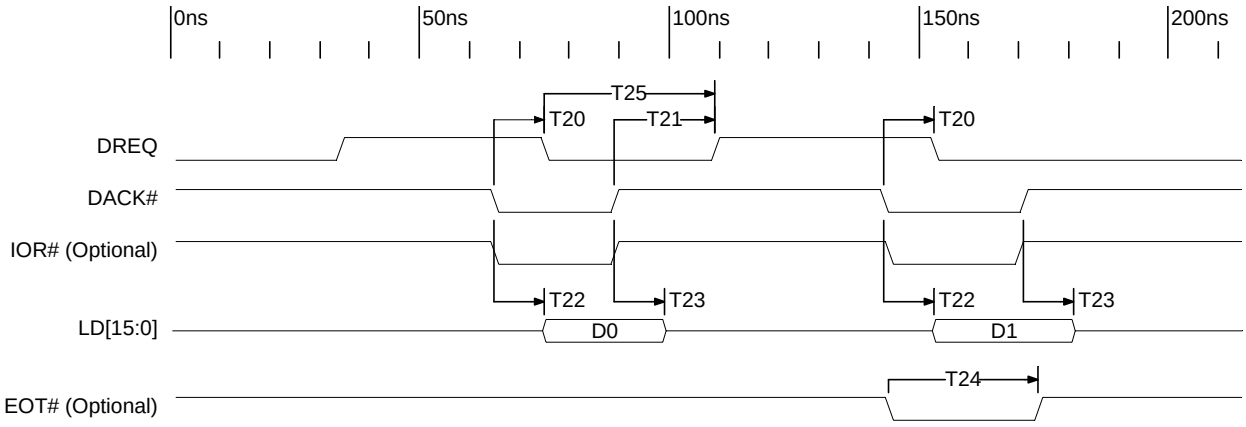
335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

10.4.10 Local Bus DMA Read; Fast Mode

Operating Conditions: V_{DDC}: 2.2-2.6V, V_{DDIO}: 3.2-3.5V, T_A = 0°C to 70°C, Output Load = 25pF

NAME	DESCRIPTION	MIN	TYP	MAX	UNIT
T20	Read enable true to DREQ false (2)	4		19	ns
T21	Read enable false to DREQ true	4	35	45	ns
T22	Data access time from read enable (1)			16	ns
T23	Data tri-state time from end of read enable	2		12	ns
T24	Width of EOT# pulse (3)	10			ns
T25	DREQ false to DREQ true	15	35		ns

(1) For non-split DMA mode, read enable is the occurrence of DACK# and optionally, IOR#. For split DMA mode, read enable is the occurrence of DACK# and optionally, DMARD#.
(2) The minimum value is only guaranteed if the *DMA Request Enable* bit in the **DMAREQ** register 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 nsec is required between the de-assertion of DMA read enable and the assertion of an I/O read enable.



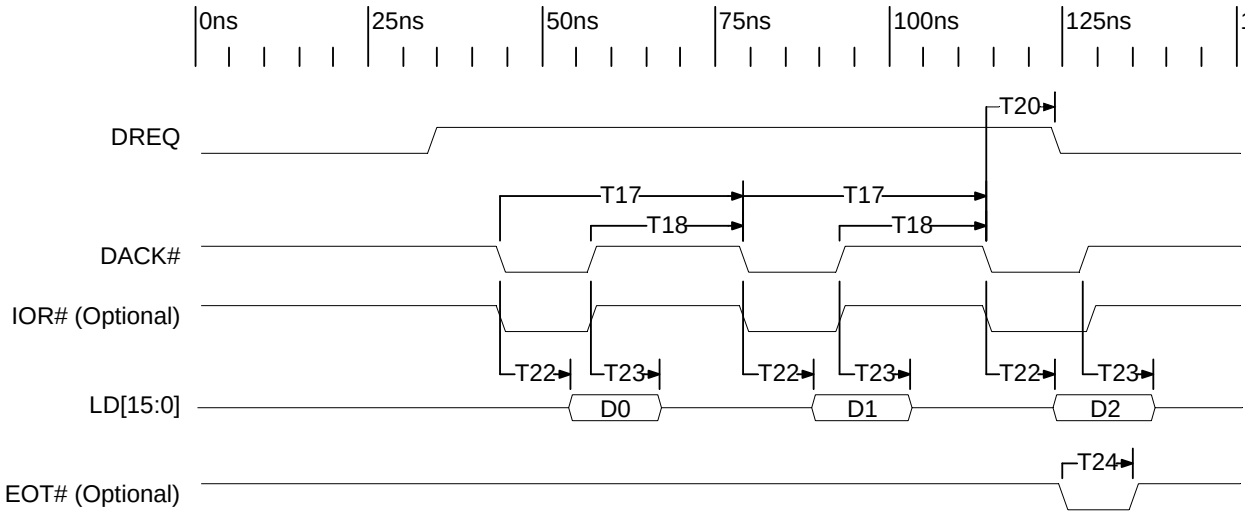
335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

10.4.11 Local Bus DMA Read; Burst Mode

Operating Conditions: V_{DDC}: 2.2-2.6V, V_{DDIO}: 3.2-3.5V, T_A = 0°C to 70°C, Output Load = 25pF

NAME	DESCRIPTION	MIN	TYP	MAX	UNIT
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 (1)			16	ns
T23	Data tri-state time from end of read enable	2		12	ns
T24	Width of EOT# pulse (3)	10			ns

(1) For non-split DMA mode, read enable is the occurrence of DACK# and optionally, IOR#. For split DMA 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 nsec is required between the de-assertion of DMA read enable and the assertion of an I/O read enable.



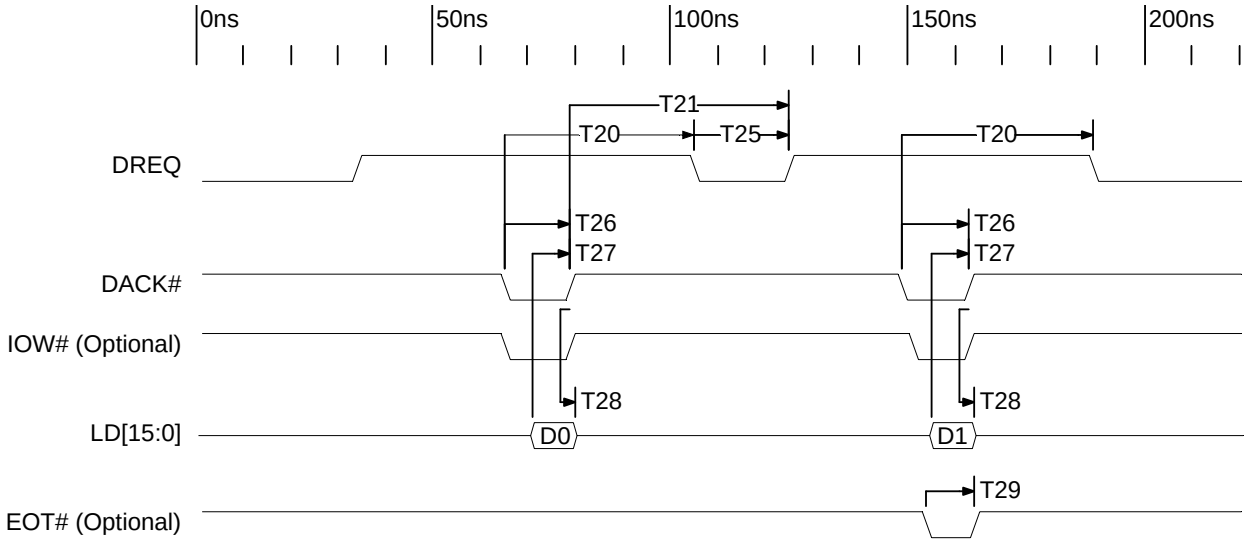
335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

10.4.12 Local Bus DMA Write; Slow Mode

Operating Conditions: V_{DDC}: 2.2-2.6V, V_{DDIO}: 3.2-3.5V, T_A = 0°C to 70°C, Output Load = 25pF

NAME	DESCRIPTION	MIN	TYP	MAX	UNIT
T20	Write enable true to DREQ false (2)	17	50	60	ns
T21	Write enable false to DREQ true	23	50	72	ns
T25	DREQ false to DREQ true	15	25		ns
T26	Write enable width (1)	5			ns
T27	Data setup to end of write enable	5			ns
T28	Data hold time from end of write enable	0			ns
T29	Width of EOT# pulse (3)	5			ns

(1) For non-split DMA mode, write enable is the occurrence of DACK# and optionally, IOW#. For split DMA mode, write enable is the occurrence of DACK# and optionally, DMAWR#.
(2) The minimum value is only guaranteed if the *DMA Request Enable* bit in the **DMAREQ** register 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.



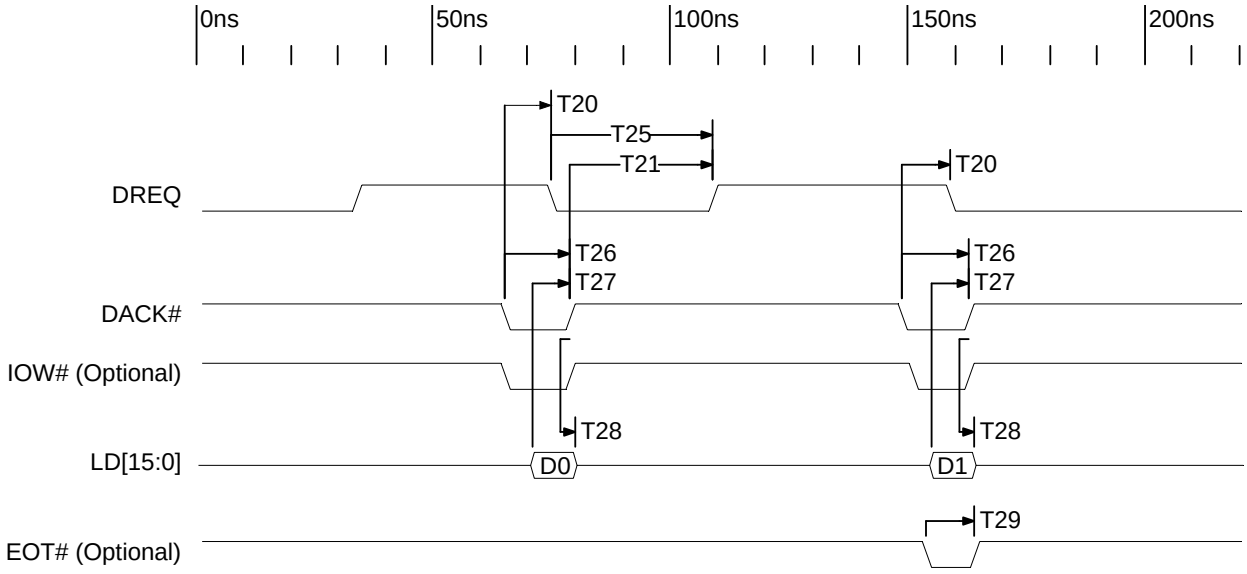
335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

10.4.13 Local Bus DMA Write; Fast Mode

Operating Conditions: V_{DDC}: 2.2-2.6V, V_{DDIO}: 3.2-3.5V, T_A = 0°C to 70°C, Output Load = 25pF

NAME	DESCRIPTION	MIN	TYP	MAX	UNIT
T20	Write enable true to DREQ false (2)	4		19	ns
T21	Write enable false to DREQ true	16	45	55	ns
T25	DREQ false to DREQ true	15	40		ns
T26	Write enable width (1)	5			ns
T27	Data setup to end of write enable	5			ns
T28	Data hold time from end of write enable	0			ns
T29	Width of EOT# pulse (3)	5			ns

(1) For non-split DMA mode, write enable is the occurrence of DACK# and optionally, IOW#. For split DMA mode, write enable is the occurrence of DACK# and optionally, DMAWR#.
(2) The minimum value is only guaranteed if the *DMA Request Enable* bit in the **DMAREQ** register 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.



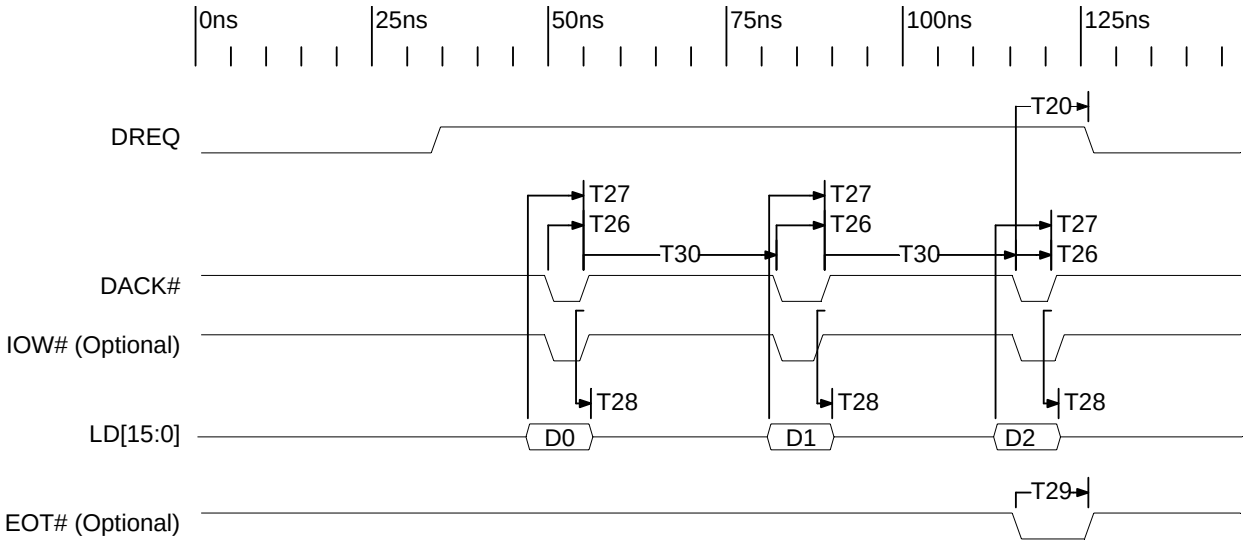
335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

10.4.14 Local Bus DMA Write; Burst Mode

Operating Conditions: V_{DDC}: 2.2-2.6V, V_{DDIO}: 3.2-3.5V, T_A = 0°C to 70°C, Output Load = 25pF

NAME	DESCRIPTION	MIN	TYP	MAX	UNIT
T20	Write enable true to DREQ false	4		20	ns
T26	Write enable width (1)	5			ns
T27	Data setup to end of write enable	5			ns
T28	Data hold time from end of write enable	0			ns
T29	Width of EOT# pulse (3)	5			ns
T30	DMA Write Recovery	28			ns

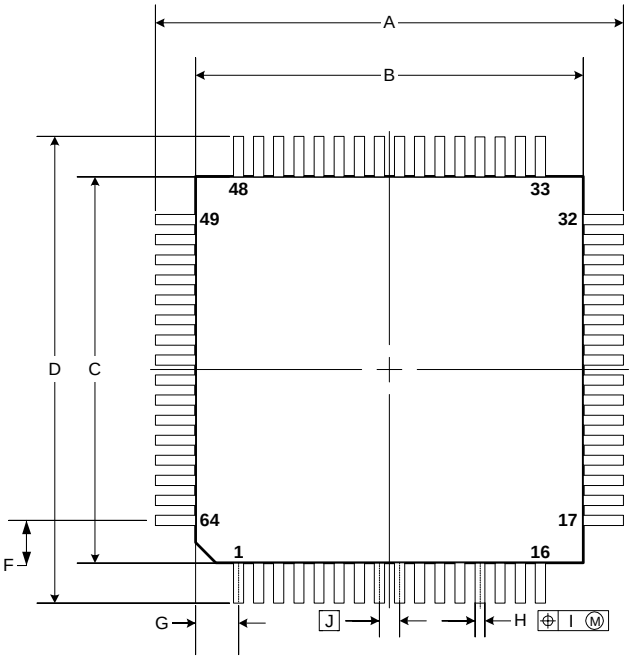
(1) For non-split DMA mode, write enable is the occurrence of DACK# and optionally, IOW#. For split DMA mode, write enable is the occurrence of DACK# and optionally, DMAWR#.
(3) EOT#, DACK#, and optionally, IOW# or DMAWR# must all be true for at least T29 for proper recognition of the EOT# pulse.



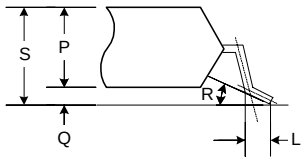
335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com

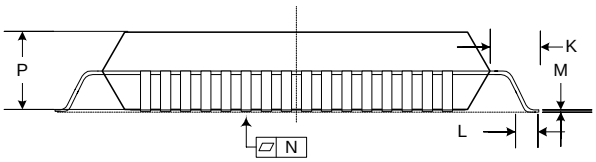
11 Mechanical Drawing

64-PIN PLASTIC TQFP (10x10)



detail of lead end





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

ITEM	MILLIMETERS	INCHES
A	12.0±0.2	0.472± 0.009 / 0.008
B	10.0±0.2	0.394± 0.008 / 0.009
C	10.0±0.2	0.394± 0.008 / 0.009
D	12.0±0.2	0.472± 0.009 / 0.008
F	1.25	0.049
G	1.25	0.049
H	0.22± 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± 0.009 / 0.008
L	0.5±0.2	0.020± 0.008 / 0.009
M	0.145± 0.055 / 0.045	0.006±0.002
N	0.10	0.004
P	1.0±0.1	0.039± 0.005 / 0.004
Q	0.1±0.05	0.004±0.002
R	3°± 7° / 3°	3°± 7° / 3°
S	1.27 MAX	0.050 MAX

S64GB-50-9EU-1

335 Pioneer Way, Mountain View, California 94041
TEL (650) 526-1490 FAX (650) 526-1494
http://www.netchip.com