# MCS7715

# USB 1.1 to Single Serial and Single Parallel Controller

**Features**

- USB Specification 1.1 Compliant
- Single 5V Operation
- Low Power
- On-Chip Voltage Regulator
- Single Serial Port
  - Supports up to 920 Kbps Serial Data Rate
  - Supports 5, 6, 7, & 8-bit Data Widths
  - Supports 1, 1.5, & 2 Stop Bits
  - Supports Even, Odd, Mark, Space and None parities
- Single Printer Port Interface
- Internal Power-On Reset
- 2KV In-Circuit ESD protection for lower cost of external components
- Available in 48-Pin QFP Package

**Applications**

- Monitoring Equipment
- High-Speed Modems
- Printer Server
- Serial Networking
- Printer Interface

**Application Note**

- AN-7715

**Evaluation Board**

- MCS7715-EVB

**General Description**

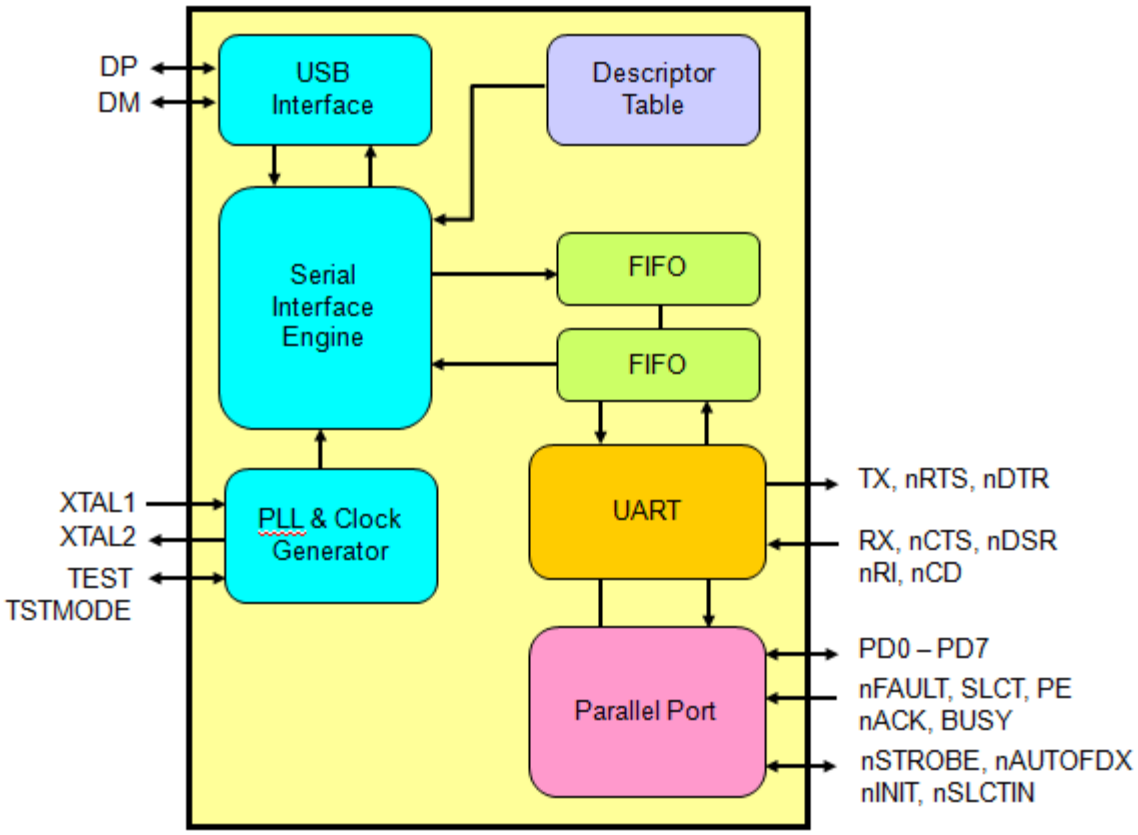
The MCS7715 controller provides bridging between the Universal Serial Bus (USB) port, an enhanced UART, and a Parallel Printer port. This device contains all the necessary logic to communicate with the host computer via the USB bus. It supports Printers with hardware accelerated SPP mode.

The MCS7715 contains a 3.3V regulator, operates in Bus-Powered mode, and has a reduced frequency (6 MHz) crystal oscillator.

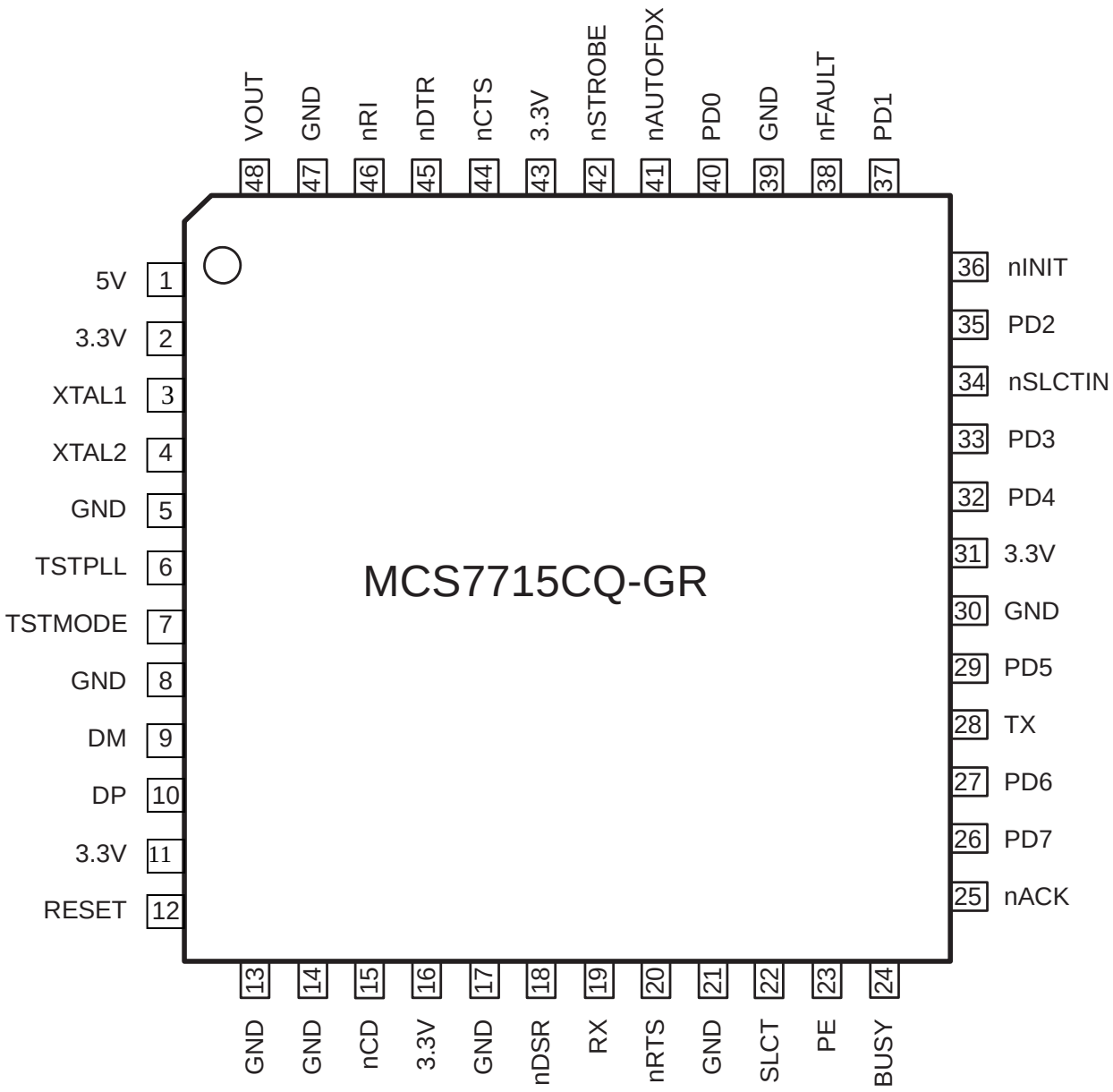
This combination of features allows significant cost savings in system design along with straightforward implementation of Serial Port and Parallel Port functionality into PC peripherals using the host's USB port.

| Ordering Information | | |
|---|---|---|
| Commercial Grade (0° C to +70° C) | | |
| MCS7715CQ-GR | 48-LQFP | RoHS |

## Block Diagram



## Pin-Out



**Pin Assignments**

| Pin Name | Pin | Type | Description |
|---|---|---|---|
| XTAL1 | 3 | I | Crystal Oscillator input, or External Clock input pin (6 MHz). This signal input is used in conjunction with XTAL2 to form a feedback circuit for the internal timing. Two (10 pF) external capacitors connected from each side of the crystal to GND are required to form a Crystal Oscillator. |
| XTAL2 | 4 | O | Crystal Oscillator output. See XTAL1 description. |
| TSTPLL | 6 | I | Test Mode (active low, internal pull-up). When this pin is tied to GND, the internal PLL is bypassed and an external 48 MHz clock is used as the reference clock. |
| TSTMODE | 7 | I | Internal Test Mode (active low, internal pull-up). When this pin is tied to GND, the Internal Test Mode is enabled. |
| DM | 9 | I/O | Upstream USB port differential Data Minus (D-), analog. |
| DP | 10 | I/O | Upstream USB port differential Data Plus (D+), analog. |
| RESET | 12 | I | System Reset (active high). Resets all internal registers, sequencers, and signals to a consistent state. Connect to GND to use the internal Power-On-Reset circuit. |
| nCD | 15 | I | Carrier-Detect signal. When low, this indicates that the modem or data set has detected the Data Carrier. nCD has no effect on the transmitter. |
| nDSR | 18 | I | Data-Set-Ready signal. When low, this indicates that the modem or data set is ready to establish a communication link. nDSR has no effect on the transmitter or receiver. |
| RX | 19 | I | UART Serial Data Input. |
| nRTS | 20 | O | Request-To-Send signal. It is set high (inactive) after a hardware reset or during internal loop-back mode. When low, this indicates that the UART is ready to exchange data. |
| SLCT | 22 | I | Peripheral/Printer Selected (internal pull-up). This pin is set high by the peripheral/printer when it is selected. |
| PE | 23 | I | Paper Empty (internal pull-up). This pin is set high by the peripheral/printer when printer paper is empty. |
| BUSY | 24 | I | Peripheral/Printer Busy (internal pull-up). This pin is set high by the peripheral/printer when the printer or peripheral is not ready to accept data. |
| nACK | 25 | I | Peripheral/Printer data Acknowledge (internal pull-up). This pin is set low by the peripheral/printer to indicate a successful data transfer has taken place. |
| PD7 | 26 | I/O | Parallel printer port Data bit 7. |

| Pin Name | Pin | Type | Description |
|---|---|---|---|
| PD6 | 27 | I/O | Parallel printer port Data bit 6. |
| TX | 28 | O | UART Serial Data Output. |
| PD5 | 29 | I/O | Parallel printer port Data bit 5. |
| PD4 | 32 | I/O | Parallel printer port Data bit 4. |
| PD3 | 33 | I/O | Parallel printer port Data bit 3. |
| nSLCTIN | 34 | I/O | Peripheral/Printer Select (open-drain).<br>Selects the peripheral/printer when it is set low. |
| PD2 | 35 | I/O | Parallel printer port Data bit 2. |
| nINIT | 36 | I/O | Initialize the peripheral/printer (open drain).<br>When set low, the peripheral/printer starts its initialization routine. |
| PD1 | 37 | I/O | Parallel printer port Data bit 1. |
| nFAULT | 38 | I | Peripheral/Printer data error (internal pull-up).<br>This pin is set low by the peripheral/printer during an error condition. |
| PD0 | 40 | I/O | Parallel printer port Data bit 0. |
| nAUTOFDX | 41 | I/O | Peripheral/Printer Auto Feed (open-drain).<br>Continuous autofed paper is selected when this pin is set low. |
| nSTROBE | 42 | I/O | Peripheral/Printer data Strobe (open drain).<br>Data is latched into the peripheral/printer when the nSTROBE is low. |
| nCTS | 44 | I | Clear-To-Send signal.<br>When low, this indicates that the modem or data set is ready to exchange data. nCTS has no effect on the transmitter. |
| nDTR | 45 | O | Data-Terminal-Ready signal.<br>It is set high (inactive) after a hardware reset or during internal loop-back mode. When low, this output indicates to the modem or data set that the UART is ready to establish a communication link. nDTR has no effect on the transmitter or receiver. |
| nRI | 46 | I | Ring-In detect signal. |
| VOUT | 48 | PWR | +3.3V Voltage Regulator output. |
| GND | 5, 8, 13, 14, 17,21, 30, 39, 47 | PWR | Power and Signal Ground. |
| 3.3V | 2, 11, 16, 31, 43 | PWR | Device Supply Inputs.<br>All should be connected to the VOUT pin.<br>VOUT voltage is gated by RESET. |
| 5V | 1 | PWR | Main Power Input.<br>Connect to USB VBUS or local VDD. |

**Note:** All names with “n” prefix are active low.

## USB Description

### Analog Transceivers

The on-chip transceivers are connected directly to USB cables through external series resistors. They transmit and receive serial data at both full-speed (12Mbit/s) and low-speed (1.5Mbit/s) data rates. Slew rates are automatically adjusted according to the speed of the device connected and lie within the range defined in the *USB Specification Rev. 1.1*.

### Serial Interface Engine

This engine implements the complete USB protocol layer including: parallel /serial conversion, synchronization pattern recognition, CRC checking/generation, bit (de)stuffing, packet identifier (PID) verification/generation, address recognition and handshake evaluation/generation.

### Bit Clock Recovery

The bit clock recovery circuit recovers the clock from the incoming USB data stream using 4x over sampling. It is able to track in the presence of jitter and frequency drift as specified by the USB Specification Rev. 1.1.

### 3.3V Source

A 5V to 3.3V DC-DC regulator is integral to the chip relieving the need for a +3.3V source. It supplies the analog transceivers and internal logic and can be used to supply the 1.5kΩ pull-up resistor on the DP line of the upstream connection.

### PLL Clock Multiplier

An integral Phase-Locked Loop (PLL) performs 6 to 48MHz clock multiplication and requires no external components except the crystal. This allows for the use of low-cost 6MHz crystals which reduce high frequency radiated Electro-Magnetic Interference (EMI).

### Host Requests

All standard USB requests from the host are handled via Control Endpoint-0. The Control Endpoint can handle a maximum of 8 Bytes per transfer.

This table shows the supported standard USB requests.
**Note:** the USB data transmission order is Least Significant Bit (LSB) first. In the following tables Multi-Byte variables are displayed least significant Byte first.

### Standard USB Requests

| RequestName | bmRequestType Byte 0 [7:0] (bin) | bRequest Byte 1 (hex) | wValue Byte 2, 3 (hex) | wIndex Byte 4, 5 (hex) | wLength Byte 6, 7 (hex) | Data |
|---|---|---|---|---|---|---|
| ***Address*** | | | | | | |
| Set Address | X000 0000 | 05 | address** | 00,00 | 00,00 | none |
| ***Configuration*** | | | | | | |
| Get Configuration | 1000 0000 | 08 | 00,00 | 00,00 | 01,00 | Configuration Value = 0x01 |
| Set Configuration-0 | X000 0000 | 09 | 00,00 | 00,00 | 00,00 | none |
| Set Configuration-1 | X000 0000 | 09 | 01,00 | 00,00 | 00,00 | none |
| ***Descriptor*** | | | | | | |
| Get Configuration Descriptor | 1000 0000 | 06 | 00,02 | 00,00 | length*** | Configuration, Interface, and Endpoint Descriptors |
| Get Device Descriptor | 1000 0000 | 06 | 00,01 | 00,00 | length*** | Device Descriptor |
| ***Feature*** | | | | | | |
| Clear Feature (REMOTE_WAKEUP) | X000 0000 | 01 | 01,00 | 00,00 | 00,00 | none |
| Clear Endpoint (1) Feature (HALT/STALL) | X000 0010 | 01 | 00,00 | 81,00 | 00,00 | none |
| ***Status*** | | | | | | |
| Get Device Status | 1000 0000 | 00 | 00,00 | 00,00 | 02,00 | Device Status |
| Get Interface Status | 1000 0001 | 00 | 00,00 | 00,00 | 02,00 | zero |
| Get Endpoint (0) Status | 1000 0010 | 00 | 00,00 | 00/80, 00* | 02,00 | Endpoint-0 Status |
| Get Endpoint (1) Status | 1000 0010 | 00 | 00,00 | 81,00 | 02,00 | Endpoint-1 Status |

* The MSB specifies endpoint direction: 0 = OUT, 1 = IN. Either value is accepted.
** Device Address: 0 to 127
*** Returned Value (in Bytes)

### Standard USB Descriptors

These are the supported Standard USB Descriptors:

- Device
- Configuration
- Interface
- Endpoint

**Device Descriptors**

| Offset (Bytes) | Field Name | Size (Bytes) | Value (hex) | Comments |
|---|---|---|---|---|
| 0 | bLength | 1 | 12 | Descriptor Length = 18 Bytes |
| 1 | bDescriptorType | 1 | 01 | Type = DEVICE |
| 2 | bcdUSB | 2 | 10,01 | USB Specification Rev-1.1 |
| 4 | bDeviceClass | 1 | FF | Vendor Class |
| 5 | bDeviceSubClass | 1 | 00 | n/a |
| 6 | bDeviceProtocol | 1 | FF | Vendor Class |
| 7 | bMaxPacketSize0 | 1 | 08 | Packet Size = 8 Bytes |
| 8 | idVendor | 2 | 10,97 | the Vendor-ID (9710) |
| 10 | idProduct | 2 | 15,77 | the MCS7715 Product-ID |
| 12 | bcdDevice | 2 | 00,01 | the silicon revision (1.0) |
| 14 | iManufacturer | 1 | 00 | no manufacturer string |
| 15 | iProduct | 1 | 00 | no product string |
| 16 | iSerialNumber | 1 | 00 | no serial number string |
| 17 | bNumConfigurations | 1 | 01 | one configuration |

**Configuration Descriptors**

| Offset (Bytes) | Field Name | Size (Bytes) | Value (hex) | Comments |
|---|---|---|---|---|
| 0 | bLength | 1 | 09 | Descriptor Length = 9 Bytes |
| 1 | bDescriptorType | 1 | 02 | Type = CONFIGURATION |
| 2 | wTotalLength | 2 | 35,00 | Combined Length of Configuration, + Interface, + all Endpoint Descriptors (53 Bytes) |
| 4 | bNumInterfaces | 1 | 01 | one Interface |
| 5 | bConfiguration | 1 | 01 | Configuration Value = 1 |
| 6 | iConfiguration | 1 | 00 | no configuration string |
| 7 | bmAttributes | 1 | A0 | Bus-Powered with remote wake-up |
| 8 | MaxPower | 1 | 32 | 100mA default |

**Interface Descriptors**

| Offset (Bytes) | Field Name | Size (Bytes) | Value (hex) | Comments |
|---|---|---|---|---|
| 0 | bLength | 1 | 09 | Descriptor Length = 9 Bytes |
| 1 | bDescriptorType | 1 | 04 | Type = INTERFACE |
| 2 | bInterfaceNumber | 1 | 00 | n/a |
| 3 | bAlternateSetting | 1 | 00 | no alternate setting |
| 4 | bNumEndpoints | 1 | 05 | five Endpoints |
| 5 | bInterfaceClass | 1 | FF | Vendor Class |
| 6 | bInterfaceSubClass | 1 | 00 | n/a |
| 7 | bInterfaceProtocol | 1 | FF | Vendor Class |
| 8 | bInterface | 1 | 00 | no interface string |

**Endpoint Descriptions**

There are five end points apart from the Control Endpoint

| Endpoint | Type | Function | Size |
|---|---|---|---|
| 0 | Control Endpoint | Default Functionality | |
| 1 | Bulk-In | Parallel Printer Port | 32Bytes |
| 2 | Bulk-Out | Parallel Printer Port | 32 Bytes |
| 3 | Bulk-In | Serial Port | 64 Bytes |
| 4 | Bulk-Out | Serial Port | 32 Bytes |
| 5 | Interrupt | Status Endpoint | 4 Bytes |

**Endpoint Descriptor #1 Bulk-In (Parallel Port)**

| Offset (Bytes) | Field Name | Size (Bytes) | Value (hex) | Comments |
|---|---|---|---|---|
| 0 | bLength | 1 | 07 | Descriptor Length = 7 Bytes |
| 1 | bDescriptorType | 1 | 05 | Type = ENDPOINT |
| 2 | bEndpointAddress | 1 | 81 | Endpoint-1, direction: IN |
| 3 | bmAddress | 1 | 02 | Bulk Endpoint |
| 4 | wMaxPacketSize | 2 | 20,00 | Packet Size = 32 Bytes |
| 6 | bInterval | 1 | FF | Polling Interval (255mS) |

**Endpoint Descriptor #2 Bulk-Out (Parallel Port)**

| Offset (Bytes) | Field Name | Size (Bytes) | Value (hex) | Comments |
|---|---|---|---|---|
| 0 | bLength | 1 | 07 | Descriptor Length = 7 Bytes |
| 1 | bDescriptorType | 1 | 05 | Type = ENDPOINT |
| 2 | bEndpointAddress | 1 | 02 | Endpoint-2, direction: OUT |
| 3 | bmAddress | 1 | 02 | Bulk Endpoint |
| 4 | wMaxPacketSize | 2 | 20,00 | Packet Size = 32 Bytes |
| 6 | bInterval | 1 | FF | |

**Endpoint Descriptor #3 Bulk-In (Serial Port)**

| Offset (Bytes) | Field Name | Size (Bytes) | Value (hex) | Comments |
|---|---|---|---|---|
| 0 | bLength | 1 | 07 | Descriptor Length = 7 Bytes |
| 1 | bDescriptorType | 1 | 05 | Type = ENDPOINT |
| 2 | bEndpointAddress | 1 | 83 | Endpoint-3, direction: IN |
| 3 | bmAddress | 1 | 02 | Bulk Endpoint |
| 4 | wMaxPacketSize | 2 | 40,00 | Packet Size = 64 Bytes |
| 6 | bInterval | 1 | 01 | Polling Interval (1mS) |

**Endpoint Descriptor #4 Bulk-Out (Serial Port)**

| Offset (Bytes) | Field Name | Size (Bytes) | Value (hex) | Comments |
|---|---|---|---|---|
| 0 | bLength | 1 | 07 | Descriptor Length = 7 Bytes |
| 1 | bDescriptorType | 1 | 05 | Type = ENDPOINT |
| 2 | bEndpointAddress | 1 | 04 | Endpoint-4, direction: OUT |
| 3 | bmAddress | 1 | 02 | Bulk Endpoint |
| 4 | wMaxPacketSize | 2 | 20,00 | Packet Size = 32 Bytes |
| 6 | bInterval | 1 | FF | |

**Endpoint Descriptor #5 Status Endpoint**

| Offset (Bytes) | Field Name | Size (Bytes) | Value (hex) | Comments |
|---|---|---|---|---|
| 0 | bLength | 1 | 07 | Descriptor Length = 7 Bytes |
| 1 | bDescriptorType | 1 | 05 | Type = ENDPOINT |
| 2 | bEndpointAddress | 1 | 87 | Endpoint-7, direction: IN |
| 3 | bmAddress | 1 | 03 | Interrupt Endpoint |
| 4 | wMaxPacketSize | 2 | 04,00 | Packet Size = 4 Bytes |
| 6 | bInterval | 1 | 01 | Polling Interval (1mS) |

**Status Endpoint:**
The Status Endpoint returns 4 Bytes every 1 millisecond. These four Bytes are status information of the Parallel Printer port and the Serial port.

- Byte-1: Interrupt Identification Register of the Serial port (IIR)
- Byte-2: Reserved (Unused)
- Byte-3: Device Status Register of the Parallel Printer port (DSR)
- Byte-4: FIFO status of both the Parallel Printer port and the Serial port.

**Description for Byte-4 of Status Endpoint.**

| Bit[7] | Bit[6] | Bit[5] | Bit[4] | Bit[3] | Bit[2] | Bit[1] | Bit[0] |
|---|---|---|---|---|---|---|---|
| Reserved | Reserved | SerInFifo Status | SerOutFifo Status | ParOutFifo Empty | ParOutFifo Full | ParInFifo Empty | ParInFifo Full |

| Bit | Description | |
|---|---|---|
| ParInFifo Full | 1 Indicates the Bulk-In FIFO is full. | (Parallel Printer Port) |
| ParInFifo Empty | 1 Indicates the Bulk-In FIFO is empty. | (Parallel Printer Port) |
| ParOutFifo Full | 1 Indicates the Bulk-Out FIFO is full. | (Parallel Printer Port) |
| ParOutFifo Empty | 1 Indicates the Bulk-Out FIFO is empty. | (Parallel Printer Port) |
| SerOutFifo Status | 1 Indicates the Bulk-Out FIFO is empty.<br>0 indicates the Bulk-Out FIFO is filled with Tx Data. | (Serial Port) |
| SerInFifo Status | 1 indicates the Bulk-In FIFO is filled with Rx Data.<br>0 indicates the Bulk-In FIFO is empty. | (Serial Port) |

**USB Vendor Specific Commands:**
The purpose of the vendor commands is to configure the Serial and Parallel Printer ports.
The following tables provide information for the Vendor Specific Commands.

**PARALLEL PRINTER PORT**

Get Application Vendor Specific Command (Parallel Printer Port)

| BmrequestType | Brequest | Wvalue | Windex | Wlength | Register Name |
|---|---|---|---|---|---|
| 0xC0 | 0x0D | 0x01*xx* | 0x0000 | 0x0001 | PP_DPR |
| 0xC0 | 0x0D | 0x01*xx* | 0x0001 | 0x0001 | PP_DSR |
| 0xC0 | 0x0D | 0x01*xx* | 0x0002 | 0x0001 | PP_DCR |
| 0xC0 | 0x0D | 0x01*xx* | 0x0008 | 0x0001 | PP_C_FIFO |
| 0xC0 | 0x0D | 0x01*xx* | 0x0008 | 0x0001 | PP_CONF_A |
| 0xC0 | 0x0D | 0x01*xx* | 0x0009 | 0x0001 | PP_CONF_B |
| 0xC0 | 0x0D | 0x01*xx* | 0x000A | 0x0001 | PP_ECR |

Set Application Vendor Specific Command (Parallel Printer Port)

| BmrequestType | Brequest | Wvalue | Windex | Wlength | Register Name |
|---|---|---|---|---|---|
| 0x40 | 0x0E | 0x01*xx* | 0x0000 | 0x0000 | PP_DPR |
| 0x40 | 0x0E | 0x01*xx* | 0x0001 | 0x0000 | PP_DSR |
| 0x40 | 0x0E | 0x01*xx* | 0x0002 | 0x0000 | PP_DCR |
| 0x40 | 0x0E | 0x01*xx* | 0x0008 | 0x0000 | PP_C_FIFO |
| 0x40 | 0x0E | 0x01*xx* | 0x0008 | 0x0000 | PP_CONF_A |
| 0x40 | 0x0E | 0x01*xx* | 0x0009 | 0x0000 | PP_CONF_B |
| 0x40 | 0x0E | 0x01*xx* | 0x000A | 0x0000 | PP_ECR |

**SERIAL PORT**

Get Application Vendor Specific Command (Serial Port)

| BmrequestType | Brequest | Wvalue | Windex | Wlength | Register Name |
|---|---|---|---|---|---|
| 0xC0 | 0x0D | 0x02*xx* | 0x0000 | 0x0001 | SP_RHR |
| 0xC0 | 0x0D | 0x02*xx* | 0x0001 | 0x0001 | SP_IER |
| 0xC0 | 0x0D | 0x02*xx* | 0x0002 | 0x0001 | SP_IIR |
| 0xC0 | 0x0D | 0x02*xx* | 0x0003 | 0x0001 | SP_LCR |
| 0xC0 | 0x0D | 0x02*xx* | 0x0004 | 0x0001 | SP_MCR |
| 0xC0 | 0x0D | 0x02*xx* | 0x0005 | 0x0001 | SP_LSR |
| 0xC0 | 0x0D | 0x02*xx* | 0x0006 | 0x0001 | SP_MSR |
| 0xC0 | 0x0D | 0x02*xx* | 0x0007 | 0x0001 | SP_SPR |
| 0xC0 | 0x0D | 0x02*xx* | 0x0000 | 0x0001 | SP_DLL |
| 0xC0 | 0x0D | 0x02*xx* | 0x0001 | 0x0001 | SP_DLM |

Set Application Vendor Specific Command (Serial Port)

| BmrequestType | Brequest | Wvalue | Windex | Wlength | Register Name |
|---|---|---|---|---|---|
| 0x40 | 0x0E | 0x02*xx* | 0x0000 | 0x0000 | SP_THR |
| 0x40 | 0x0E | 0x02*xx* | 0x0001 | 0x0000 | SP_IER |
| 0x40 | 0x0E | 0x02*xx* | 0x0002 | 0x0000 | SP_FCR |
| 0x40 | 0x0E | 0x02*xx* | 0x0003 | 0x0000 | SP_LCR |
| 0x40 | 0x0E | 0x02*xx* | 0x0004 | 0x0000 | SP_MCR |
| 0x40 | 0x0E | 0x02*xx* | 0x0005 | 0x0000 | SP_LSR |
| 0x40 | 0x0E | 0x02*xx* | 0x0006 | 0x0000 | SP_MSR |
| 0x40 | 0x0E | 0x02*xx* | 0x0007 | 0x0000 | SP_SPR |
| 0x40 | 0x0E | 0x02*xx* | 0x0000 | 0x0000 | SP_DLL |
| 0x40 | 0x0E | 0x02*xx* | 0x0001 | 0x0000 | SP_DLM |

**Vendor Specific Command Parameters**

Brequest: specifies whether to Read or Write
0x0E = write to the application register.
0x0D = read from the application register.

Wvalue: Specifies the Application Number.
0x0100 is the application number for the Parallel Printer port
0x0200 is the application number for the Serial port
0x0000 is the application number provided for accessing the other control registers to control the Parallel Printer port and Serial port (i.e. Enabling the Higher Baud Rates, enabling the Auto hardware Flow Control, Setting the clock frequency, etc.)
*xx* specifies the Byte value to be written into the register.

Windex: Is the offset of the register to Read/Write.

Wlength: Is the length of the data we are going to read or write.

**UART Register Set:**
The UART has 10 registers. Mapping is dependent on the Line Control Register (LCR).

| Register Name | Offset | R/W | Bit[7] | Bit[6] | Bit[5] | Bit[4] | Bit[3] | Bit[2] | Bit[1] | Bit[0] |
|---|---|---|---|---|---|---|---|---|---|---|
| THR | 0 | W | Data to be transmitted (Transmitting Holding Register) | | | | | | | |
| RHR | 0 | R | Data to be received (Receiver Holding Register) | | | | | | | |
| IER | 1 | R/W | Reserved | | | Sleep Mode | Modem Interrupt Mask | Rx Stat Interrupt Mask | THRE Interrupt Mask | RxRdy Interrupt Mask |
| FCR | 2 | W | RHR Trigger Level | | Reserved | | Reserved | Flush THR | Flush RHR | FIFO Enable |
| ISR | 2 | R | FIFOs Enabled | | Reserved | | Interrupt Priority | | | Interrupt Pending |
| LCR | 3 | R/W | DLE | Tx Break | Force Parity | Odd/Even Parity | Parity Enable | Stop Bits | Data Length | |
| MCR | 4 | R/W | Reserved | | RTS/CTS Flow Control | Loop | Unused | | RTS | DTR |
| LSR | 5 | R | Data Error | Tx Empty | THR Empty | Rx Break | Framing Error | Parity Error | Overrun | RxRdy |
| MSR | 6 | R | DCD | RI | DSR | CTS | Delta DCD | TERI | Delta DSR | Delta CTS |
| SPR | 7 | R/W | Scratch Pad Register | | | | | | | |

**Additional Standard Registers - these are accessed when LCR[7] = 1**

| | | | |
|---|---|---|---|
| DLL | 0 | R/W | Divisor Latch bits[7:0] |
| DLM | 1 | R/W | Divisor Latch bits[15:8] |

**Register:** **THR**
**Description:** **Data to be transmitted**
**Offset:** **000**
**Permissions:** **Write**
**Access Condition:** **LCR[7] =0, only write condition can access this register**

| Bit[7] | Bit[6] | Bit[5] | Bit[4] | Bit[3] | Bit[2] | Bit[1] | Bit[0] |
|---|---|---|---|---|---|---|---|
| Data to be transmitted | | | | | | | |

**Register:** **RHR**
**Description:** **Data to be received**
**Offset:** **000**
**Permissions:** **Read**
**Access Condition:** **LCR[7] =0, only read condition can access this register**

| Bit[7] | Bit[6] | Bit[5] | Bit[4] | Bit[3] | Bit[2] | Bit[1] | Bit[0] |
|---|---|---|---|---|---|---|---|
| Data to be received | | | | | | | |

**Interrupt Enable Register:**
Serial channel interrupts are enabled using the Interrupt Enable Register (IER).

**Register:** **IER**
**Description:** **Interrupt Enable Register**
**Offset:** **001**
**Permissions:** **Read/Write**

| Bit[7] | Bit[6] | Bit[5] | Bit[4] | Bit[3] | Bit[2] | Bit[1] | Bit[0] |
|---|---|---|---|---|---|---|---|
| Reserved | | | Sleep Mode | Modem Interrupt Mask | Rx Stat Interrupt Mask | THRE Interrupt Mask | RxRdy Interrupt Mask |

| Bit | Name | Description |
|---|---|---|
| 0 | RxRdy Interrupt Mask | Logic 0 = Disable the Receiver Ready Interrupt<br>Logic 1 = Enable the Receiver Ready Interrupt |
| 1 | THRE Interrupt Mask | Logic 0 = Disable the Transmitter Ready Interrupt<br>Logic 1 = Enable the Transmitter Ready Interrupt |
| 2 | Rx Stat Interrupt Mask | Logic 0 = Disable the Receiver Status Interrupt (Normal Mode)<br>Logic 1 = Enable the Receiver Status Interrupt (Normal Mode) |
| 3 | Modem Interrupt Mask | Logic 0 = Disable the Modem Status Interrupt<br>Logic 1 = Enable the Modem Status Interrupt |
| 4 | Sleep Mode | Logic 0 = Disable Sleep-Mode<br>Logic 1 = Enable Sleep-Mode<br>(the internal clock of the channel is switched off) |
| 5 | Reserved | Reserved |
| 6 | Reserved | Reserved |
| 7 | Reserved | Reserved |

**FIFO Control Register:**
The FCR controls the UART behavior in various modes.

**Register:** **FCR**
**Description:** **FIFO Control Register**
**Offset:** **010**
**Permissions:** **Write**

| Bit[7] | Bit[6] | Bit[5] | Bit[4] | Bit[3] | Bit[2] | Bit[1] | Bit[0] |
|---|---|---|---|---|---|---|---|
| RHR Trigger Level | | Reserved | | Reserved | Flush THR | Flush RHR | Enable FIFO |

| Bit | Name | Description |
|---|---|---|
| 0 | Enable FIFO Mode | Logic 0 = Byte Mode<br>Logic 1 = FIFO Mode |
| 1 | Flush RHR | Logic 0 = No change<br>Logic 1 = Flushes the contents of RHR. This is operative only in FIFO Mode. The RHR is automatically flushed whenever changing between Byte Mode and FIFO Mode. The bit will return to zero after clearing the FIFOs. |
| 2 | Flush THR | Logic 0 = No change<br>Logic 1 = Flushes the content of the THR, in the same manner as FCR[1] does the RHR |
| 3 | Reserved | Reserved |
| 5, 4 | Reserved | Reserved |
| 7, 6 | RHR Trigger Level | See the table below. |

**FCR[7:6] RHR Trigger Level:**
In 550 mode, the receiver FIFO trigger levels are defined using FCR[7:6]. The interrupt trigger level & flow control trigger level where appropriate are defined by L2 in the table below. L1 defines lower flow control trigger levels that introduce a hysteresis element in hardware RTS/CTS flow control.

In Byte Mode (450 Mode) the trigger levels are all set to 1.

| FCR[7:6] | 550 Mode (FIFO = 16) | |
|---|---|---|
| | L1 | L2 |
| 2'b00 | 1 | 1 |
| 2'b01 | 1 | 4 |
| 2'b10 | 1 | 8 |
| 2'b11 | 1 | 14 |

**Interrupt Status Register:**
The source of the highest priority interrupt pending is indicated by the contents of the Interrupt Status Register (ISR). There are five sources of interrupts, and four levels of priority (1 is the highest) as tabulated below.

| Level | Interrupt Source | ISR[5:0] |
|---|---|---|
| - | No interrupt pending | 6'b000001 |
| 1 | Receiver Status Error<br>or address bit detected in 9-bit mode | 6'b000110 |
| 2a | Receiver Data Available | 6'b000100 |
| 2b | Receiver Time Out | 6'b001100 |
| 3 | Transmitter THR Empty | 6'b000010 |
| 4 | Modem Status Change | 6'b000000 |

**Note:** ISR[0] indicates whether any interrupt is pending

**Register:** **ISR**
**Description:** **Interrupt Status Register**
**Offset:** **010**
**Permissions:** **Read**

| Bit[7] | Bit[6] | Bit[5] | Bit[4] | Bit[3] | Bit[2] | Bit[1] | Bit[0] |
|---|---|---|---|---|---|---|---|
| FIFOs Enabled | | Interrupt Priority (Enhanced Mode) | | Interrupt Priority (All Modes) | | | Interrupt Pending |

**Interrupt Descriptions:**
Level1: Receiver Status Error
Normal Mode: This interrupt is active whenever any of the LSR[1], LSR[2], LSR[3] or LSR[4] are set. These flags are cleared following a read of the LSR. The interrupt is masked with IER[2].

Level 2a: Receiver Data Available
The interrupt is active whenever the receiver FIFO level is above the interrupt trigger level.

Level 2b: Receiver Time-Out
A receiver time out event, (which may cause an interrupt) will occur when all of the following conditions are true:

- The UART is in the FIFO Mode.
- There is data in the RHR
- There has been no read of the RHR for a period of time greater than the timeout period. The time-out period of time is greater than the time out period. *The time out period is four times the character period (including start & stop bits) measured from the centre of the first stop bit of the first data item received*.

Reading the first data item in RHR clears this interrupt.

Level 3: Transmitter Empty
This interrupt is set when the transmit FIFO level falls below the trigger level. It is cleared on the ISR read to Level-3 interrupt or by writing more data to the THR so that the trigger level is exceeded.

Level 4: Modem Change
This interrupt is set by the modem change flag (MSR[0], MSR[1], MSR[2] or MSR[3]) becoming active due to changes in the input modem lines. This interrupt is cleared following the read of the MSR register.

**Line Control Register:**
The LCR specifies the data format that is common to both transmitter and receiver.

**Register:** **LCR**
**Description:** **Line Control Register**
**Offset:** **011**
**Permissions:** **Read/Write**
**Access Condition:** **LCR[7] =0**

| Bit[7] | Bit[6] | Bit[5] | Bit[4] | Bit[3] | Bit[2] | Bit[1] | Bit[0] |
|---|---|---|---|---|---|---|---|
| DLE | Tx Break | Force Parity | Odd/Even Parity | Parity Enable | Number of Stop Bits | Data Length | |

LCR[1:0]: Determines the data length of serial characters.

| LCR[1:0] | Data Length |
|---|---|
| 2'b00 | 5 bits |
| 2'b01 | 6 bits |
| 2'b10 | 7 bits |
| 2'b11 | 8 bits |

LCR[2]: Defines the number of stop bits per serial character.

| LCR[2] | Data Length | Stop Bits |
|---|---|---|
| 0 | 5,6,7,8 | 1 |
| 1 | 5 | 1.5 |
| 1 | 6,7,8 | 2 |

LCR[5:3]: The selected parity type will be generated during transmission and checked by the receiver, which may produce a parity error as a result. In 9-bit mode parity is disabled and LCR[5:3] are ignored.

| LCR[5:3] | Parity Type |
|---|---|
| 3'bxx0 | No parity |
| 3'b001 | Odd parity |
| 3'b011 | Even parity |
| 3'b101 | Parity bit forced to 1 |
| 3'b111 | Parity bit forced to 0 |

LCR[6]: Transmission Break
- Logic 0: Break transmission disabled
- Logic 1: Forces the transmitter data output (SOUT) low to alert the communications channel. It is the responsibility of the software driver to ensure that the break duration is longer than the character period for it to be recognized remotely as a break rather than data.

LCR[7]: Divisor Latch Enable
- Logic 0: Accesses to DLL and DLM registers disabled
- Logic 1: Accesses to DLL and DLM registers enabled

**Line Status Register:**
This register provides the status of the data transfer to the CPU.

**Register:** **LSR**
**Description:** **Line Status Register**
**Offset:** **101**
**Permissions:** **Read**
**Access Condition:** **LCR[7] =0, ACR[6] = 0**

| Bit[7] | Bit[6] | Bit[5] | Bit[4] | Bit[3] | Bit[2] | Bit[1] | Bit[0] |
|---|---|---|---|---|---|---|---|
| Data Error | Tx Empty | THR Empty | Rx Break | Framing Error | Parity Error | Overrun | RxRdy |

| Bit | Name | Description |
|---|---|---|
| 0 | RHR Data Available | Logic 0 = RHR is empty<br>Logic 1 = RHR is not empty, data is available to be read |
| 1 | RHR Overrun | Logic 0 = No overrun error<br>Logic 1 = Data was received when the RHR was full, An overrun has occurred. The error is flagged when the data would normally have been transferred to the RHR. |
| 2 | Received Data Parity Error | Logic 0 = No parity error in received data, or 9th bit is “0” in 9-bit mode.<br>Logic 1 = Data has been received that did not have correct parity |
| 3 | Received Data Framing Error | Logic 0 = No framing error<br>Logic 1 = data has been received with an invalid stop bit. |
| 4 | Received Break Error | Logic 0 = No receiver break error<br>Logic 1 = the receiver received a break error |
| 5 | THR Empty | Logic 0 = Transmitter FIFO is not empty<br>Logic 1 = Transmitter FIFO is empty |
| 6 | Transmitter & THR Empty | Logic 0 = The transmitter is not idle<br>Logic 1 = THR is empty & the transmitter has completed the character in the shift register and is in the idle mode |
| 7 | Receiver Data Error | Logic 0 = Either there are no receiver data errors in the FIFO, or it was cleared by earlier read of LSR<br>Logic 1 = At least one parity error, framing error or break indication in the FIFO. |

**Note :** A break condition occurs when the SIN line goes low and stays low through out the start, data, parity & first stop bits. One zero character associated with break flag set will be transferred to the RHR and the receiver will then wait until the SIN line returns high. The LSR[4] flag break flag is set when this data item gets to the top of the RHR and it is cleared following the read to the LSR.

**Modem Control Register:**

**Register:** **MCR**
**Description:** **Modem Control Register**
**Offset:** **100**
**Permissions:** **Read/Write**

| Bit[7] | Bit[6] | Bit[5] | Bit[4] | Bit[3] | Bit[2] | Bit[1] | Bit[0] |
|---|---|---|---|---|---|---|---|
| 550 Mode | | | | | | | |
| Unused | | CTS/RTS Flow Control | Internal Loop Back Enable | Out2 (Interrupt Enable) | Out1 | RTS | DTR |

| Bit | Name | Description |
|---|---|---|
| 0 | DTR | Logic 0 = Forces DTR# output to inactive (high)<br>Logic 1 = Forces DTR# output to active (low) |
| 1 | RTS | Logic 0 = Forces RTS# output to inactive (high)<br>Logic 1 = Forces RTS# output to active (low) |
| 2 | Out1 | Unused |
| 3 | Out2 | Unused |
| 4 | Internal Loop Back Enable | Logic 0 = Normal operating mode<br>Logic 1 = Enable Local Loop-Back Mode |
| 5 | CTS/RTS flow control | Logic 0 = CTS/RTS flow control Disabled in 550-Mode<br>Logic 1 = CTS/RTS flow control Enabled in 550-Mode |
| 6 | Unused | Unused |
| 7 | Unused | Unused |

**Modem Status Register:**
This register provides the status of the modem control lines to CPU.

**Register:** **MSR**
**Description:** **Modem Status Register**
**Offset:** **110**
**Permissions:** **Read**

| Bit[7] | Bit[6] | Bit[5] | Bit[4] | Bit[3] | Bit[2] | Bit[1] | Bit[0] |
|---|---|---|---|---|---|---|---|
| DCD | RI | DSR | CTS | Delta DCD | Trailing Edge RI | Delta DSR | Delta CTS |

| Bit | Name | Description |
|---|---|---|
| 0 | Delta CTS | Logic 0 = no change in the CTS signal<br>Logic 1 = indicates that the CTS input has changed since the last time the MSR was read |
| 1 | Delta DSR | Logic 0 = no change in the DSR signal<br>Logic 1 = indicates that the DSR input has changed since the last time the MSR was read |
| 2 | Trailing Edge RI | Logic 0 = no change in the RI signal<br>Logic 1 = indicates that the RI input has changed from low to high since the last time the MSR was read |
| 3 | Delta DCD | Logic 0 = no change in the DCD signal<br>Logic 1 = indicates that the DCD input has changed since the last time the MSR was read |
| 4 | CTS | Logic 0 = CTS# line is 1<br>Logic 1 = CTS# line is 0 |
| 5 | DSR | Logic 0 = DSR# line is 1<br>Logic 1 = DSR# line is 0 |
| 6 | RI | Logic 0 = RI# line is 1<br>Logic 1 = RI# line is 0 |
| 7 | DCD | Logic 0 = DCD# line is 1<br>Logic 1 = DCD# line is 0 |

**Scratch Pad Register:**
The scratch pad register does not effect operation of the rest of the UART in any way and can be used for the temporary data storage.

**Register:** **SPR**
**Description:** **Scratch Pad Register**
**Offset:** **111**
**Permissions:** **Read/Write**

| Bit[7] | Bit[6] | Bit[5] | Bit[4] | Bit[3] | Bit[2] | Bit[1] | Bit[0] |
|---|---|---|---|---|---|---|---|
| Scratch Pad Register | | | | | | | |

**Divisor Latch Registers:**
The divisor latch registers (DLL & DLM) are used to program the baud rate divisor. This is a value between 1 and 65535 by which the input clock is divided in order to generate serial Baud Rates. After a hardware Reset, the Baud Rate used by the transmitter & receiver is given by:

Baud Rate = Input Clock / 16 * Divisor

where divisor is given by: (256 * DLM ) + DLL

**Note:** More flexible Baud Rate generation options are also available. These require the use of Advanced Features in other registers however.

**Register:** **DLL**
**Description:** **Divisor Latch Register**
**Offset:** **000**
**Permissions:** **Read/Write**
**Access Condition:** **LCR[7] =1, Address = 000**

| Bit[7] | Bit[6] | Bit[5] | Bit[4] | Bit[3] | Bit[2] | Bit[1] | Bit[0] |
|---|---|---|---|---|---|---|---|
| Least significant Byte for divisor latch | | | | | | | |

**Register:** **DLM**
**Description:** **Divisor Latch Register**
**Offset:** **001**
**Permissions:** **Read/Write**
**Access Condition:** **LCR[7] =1, Address = 001**

| Bit[7] | Bit[6] | Bit[5] | Bit[4] | Bit[3] | Bit[2] | Bit[1] | Bit[0] |
|---|---|---|---|---|---|---|---|
| Most significant Byte for divisor latch | | | | | | | |

**Baud Rate Generator Programming Table**

| Baud Rate | DLM (Hex) | DLL (Hex) |
|---|---|---|
| 115.2K | 00 | 01 |
| 57.6K | 00 | 02 |
| 38.4K | 00 | 03 |
| 19.2K | 00 | 06 |
| 9600 | 00 | 0C |
| 2400 | 00 | 30 |
| 1200 | 00 | 60 |
| 600 | 00 | C0 |
| 300 | 01 | 80 |
| 150 | 03 | 00 |
| 50 | 09 | 00 |

**Master Reset Values**

| Register | BIT-7 | BIT-6 | BIT-5 | BIT-4 | BIT-3 | BIT-2 | BIT-1 | BIT-0 |
|---|---|---|---|---|---|---|---|---|
| RHR | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| THR | X | X | X | X | X | X | X | X |
| IER | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| FCR | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| IIR | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| LCR | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| MCR | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| LSR | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| MSR | X | X | X | X | 0 | 0 | 0 | 0 |
| SPR | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Parallel Printer Port Register Descriptions**

**Data Register**
The Data Register is cleared at initialization by RESET. During a write operation, the contents of this register are buffered and output onto the PD7-PD0 ports. During a read operation PD7-PD0 ports are buffered and output to the host CPU.

**Register:** **PP_DPR**
**Description:** **Parallel Printer Data Register**
**Offset:** **000**
**Permissions:** **Read/Write**
**Access Condition:** **Application Number = 0x0100**

| Bit[7] | Bit[6] | Bit[5] | Bit[4] | Bit[3] | Bit[2] | Bit[1] | Bit[0] |
|---|---|---|---|---|---|---|---|
| Parallel Printer Data | | | | | | | |

**Device Status Register**
The bits of the status register are defined as follows:

**Register:** **DSR**
**Description:** **Device Status Register**
**Offset:** **001**
**Permissions:** **Read**
**Access Condition:** **Application Number = 0x0100**

| Bit[7] | Bit[6] | Bit[5] | Bit[4] | Bit[3] | Bit[2] | Bit[1] | Bit[0] |
|---|---|---|---|---|---|---|---|
| nBUSY | nACK | PE | SLCT | nFAULT | Unused | Unused | Unused |

| Bit | Name | Description |
|---|---|---|
| 0 | Unused | Not used, set to 0 |
| 1 | Unused | Not used, set to 0 |
| 2 | Unused | Not used, set to 0 |
| 3 | nFAULT | Logic 0 = Printer reports an error condition exists.<br>Logic 1 = Normal operation. |
| 4 | SLCT | Logic 0 = Printer is off line.<br>Logic 1 = Printer is on line. |
| 5 | PE | Logic 0 = Normal operation<br>Logic 1 = Paper end/empty is detected |
| 6 | nACK | Logic 0 = State of the nACK pin (nACK = Low).<br>Logic 1 = State of the nACK pin (nACK = High). |
| 7* | nBUSY | Logic 0 = BUSY pin is high, printer is not ready to take data.<br>Logic 1 = BUSY pin is low, printer is ready to take data. |

**Note:** Bit-7 (nBUSY) in this register is logically inverted from the state of the electrical signal appearing at the physical device pin. When the printer is BUSY, this bit will read back as a zero.

**Device Control Register**

**Register:** **DCR**
**Description:** **Device Control Register**
**Offset:** **002**
**Permissions:** **Read/Write**
**Access Condition:** **Application Number = 0x0100**

| Bit[7] | Bit[6] | Bit[5] | Bit[4] | Bit[3] | Bit[2] | Bit[1] | Bit[0] |
|---|---|---|---|---|---|---|---|
| Unused | Unused | DIR | Unused | SLCTIN | nINIT | AUTOFD | STROBE |

| Bit | Name | Description |
|---|---|---|
| 0* | STROBE | Logic 0 = Sets the nSTROBE pin to high.<br>Logic 1 = Sets the nSTROBE pin to low. PD7-PD0 data are latched into printer |
| 1* | AUTOFD | Logic 0 = Sets the nAUTOFDX pin to high. No auto feed function.<br>Logic 1 = Sets the nAUTOFDX pin to low. Printer generates auto line feed after each line is printed. |
| 2 | nINIT | Logic 0 = Peripheral/printer starts its initialization routine.<br>Logic 1 = Normal Operation. |
| 3* | SLCTIN | Logic 0 = Sets the nSLCTIN pin to high. Printer is not selected.<br>Logic 1 = Sets the nSLCTIN pin to low. Selects the printer. |
| 4 | Unused | Not uset set to 0 |
| 5 | DIR | Logic 0 = PD7-PD0 pins are configured for output mode.<br>Logic 1 = PD7-PD0 pins are configured for input mode. |
| 6 | Unused | Not used, set to 0. |
| 7 | Unused | Not used, set to 0. |

**Note:** Three bits (0, 1, & 3) of this register are logically inverted from the state of the electrical signals appearing at the physical device pins they control. The physical pins for these three bits are all Active-Low signals, so writing a “one” in this register will enable or activate the desired function.

The physical pin associated with Bit-2 (nINIT) of this register is also an Active-Low electrical signal. This bit is not inverted however, so in order to start the Initialization process, this bit must be set LOW.

**Extended Control Register (ECR)**
This register controls the Parallel Port mode selection operation.

**Register:** **PP_ECR**
**Description:** **Parallel Printer Port Extended Control Register**
**Offset:** **0x0A**
**Permissions:** **Read/Write**
**Access Condition:** **Application Number = 0x0100**

| Bit[7] | Bit[6] | Bit[5] | Bit[4] | Bit[3] | Bit[2] | Bit[1] | Bit[0] |
|---|---|---|---|---|---|---|---|
| Parallel Port Mode | | | Reserved | Reserved | Reserved | Reserved | Reserved |

| Bit-7 | Bit-6 | Bit-5 | Operating Mode |
|---|---|---|---|
| 0 | 0 | 0 | SPP |
| 0 | 0 | 1 | Nibble |
| 0 | 1 | 0 | CB-FIFO mode |
| 0 | 1 | 1 | Not Used |
| 1 | 0 | 0 | Not Used |
| 1 | 0 | 1 | Not used |
| 1 | 1 | 0 | Not Used |
| 1 | 1 | 1 | Not Used |

**Mode Changes**
After a hardware reset Nibble Mode is selected as the default mode. It is required to select mode 000 or 001 before any other mode configuration.

**Mode "000"**
**SPP/Centronics/Compatible Mode**
Forward direction only. The direction bit is forced to "0" and PD7-PD0 are set to the output direction. The Parallel Port is under software control. This mode defines the protocol used by most PCs to transfer data to a printer. It is commonly called the "Centronics" mode and is the method utilized with the Standard Parallel Printer port. Data is placed on the PD7-PD0 ports, and the printer status is checked via the DSR register. If no error condition is flagged and the printer is not busy, software toggles the nSTROBE pin to latch the PD7-PD0 data into the printer. This operating cycle continues when the printer/peripheral issues the data acknowledge signal (pulses the ACK and nBUSY pins).

**Mode "001"**
**Nibble Mode**
The Nibble Mode is the most common way to get reverse channel data from the printer or peripheral. This mode is usually combined with the SPP Mode or a proprietary forward channel mode to create a Bi-Directional channel. In this mode printer status bits are used as Nibble bits.

| Pin | Data Bit |
|---|---|
| BUSY | Bit-7 |
| PE | Bit-6 |
| SLCT | Bit-5 |
| nFAULT | Bit-4 |
| BUSY | Bit-3 |
| PE | Bit-2 |
| SLCT | Bit-1 |
| nFAULT | Bit-0 |

**Bit Order for Nibble Mode**

**Mode "010"**
**FIFO Output Mode**
In this mode, Bytes written to the FIFO are transmitted automatically using the SPP/Centronics standard protocol.

**Vendor Specific Command Registers:**
There are three vendor specific registers which are used to tune the behavior and performance of the UART and Parallel Port. They are as follows

- SP_REG
- PP_REG
- SP_CONTROL_REG

**SP_REGISTER:**
This register is used for internal debugging of UDC controller & bridge circuitry. This enables the designer to pin point the problem in the design. This register also enables the hardware flow control. This register has bits which clear the Bulk-In & Bulk-Out FIFOs. There is a bit which resets the UART. There are bits which control the input clock fed to the UART, providing options for higher Baud Rates.

**Register:** **SP_REG**
**Description:** **Serial Port Register**
**Offset:** **0x01**
**Permissions:** **Read/Write**
**Access Condition:** **Application Number = 0x0000**

| Bit[7] | Bit[6] | Bit[5] | Bit[4] | Bit[3] | Bit[2] | Bit[1] | Bit[0] |
|---|---|---|---|---|---|---|---|
| UART_Reset | Clk_UART_Select | | | Sp_bi_clear | Sp_bo_clear | Ser_line_err_ctl_en | Udc_loop |

| Bit | Description |
|---|---|
| Udc_loop | When enabled, loops the data from the Bulk-Out FIFO to the Bulk-In FIFO. |
| Ser_line_err_ctl_en | When enabled, will not allow the data from the UART to be written into the Bulk-In FIFO if there are any errors in the received data. |
| Sp_bo_clear | Reset the Bulk-Out FIFO |
| Sp_bi_clear | Reset the Bulk-In FIFO |
| Clk_UART_Select | Changes the clock fed to UART as shown in the table below |
| UART_Reset | Resets the UART |

**Clk_UART_Select**: Changes the clock fed to the Serial Port as shown in the table below.

| Option | Input Clock Frequency |
|---|---|
| 3'b000 | 12 MHz |
| 3'b001 | 6 MHz |
| 3'b010 | 3 MHz |
| 3'b011 | 1.5 MHz |
| 3'b100 | 0.75 MHz |

**PP_REGISTER:**
This register is used for internal debugging of UDC controller & bridge circuitry. There is a bit which resets the Parallel Printer port. There are bits which control the input clock fed to the Parallel Printer port.

**Register:** **PP_REG**
**Description:** **Parallel Printer Port Register**
**Offset:** **0x04**
**Permissions:** **Read/Write**
**Access Condition:** **Application Number = 0x0000**

| Bit[7] | Bit[6] | Bit[5] | Bit[4] | Bit[3] | Bit[2] | Bit[1] | Bit[0] |
|---|---|---|---|---|---|---|---|
| pp_reset | clk_freq_select | | | pp_bi_clear | pp_bo_clear | Reserved | udc_loop |

| Bit | Description |
|---|---|
| udc_loop | When enabled, loops the data from the Bulk-Out FIFO to the Bulk-In FIFO. |
| Reserved | Reserved |
| pp_bo_clear | Reset the Bulk-Out FIFO |
| pp_bi_clear | Reset the Bulk-In FIFO |
| clk_freq_select | Changes the clock fed to the Parallel Printer port. |
| pp_reset | Resets the Parallel Printer port |

**pp reset:** This bit will reset the Parallel Printer port. When this bit is a 1 the Parallel Printer port is under the Reset process. Whenever the frequency is to be changed, first set the Reset bit, change the clk_ frequency, and finally clear the Reset bit to 0.

**clk_freq_select:** Changes the clock fed to the Parallel Printer port as shown in the table below.

| Option | Input Clock Frequency |
|---|---|
| 3'b000 | 12 MHz |
| 3'b001 | 6 MHz |
| 3'b010 | 3 MHz |
| 3'b011 | 1.5 MHz |
| 3'b100 | 0.75 MHz |

**SP_CONTROL_REGISTER:**
The Control register is used for controlling the Auto RTS/CTS flow control of the Serial Port.

**Register:** **SP_CONTROL_REG**
**Description:** **Control Register for Serial Port**
**Offset:** **0x08**
**Permissions:** **Read/Write**
**Access Condition:** **Application Number = 0x0000**

| Bit[7] | Bit[6] | Bit[5] | Bit[4] | Bit[3] | Bit[2] | Bit[1] | Bit[0] |
|---|---|---|---|---|---|---|---|
| Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | sp_autoflow_ctrl_en |

| Bit | Description |
|---|---|
| sp_autoflow_ctrl_en | This bit is used for enabling the hardware flow control for the Serial port. |
| Reserved | Reserved |

## Electrical Characteristics

### Absolute Maximum Ratings

| | |
|---|---|
| Supply Voltage | 6 Volts |
| Input Voltage (I/O) | -0.3 to $V_{CC}$ +0.3 |
| Storage Temperature | -60° C to +150° C |

### Recommended Operating Conditions

| | |
|---|---|
| Supply Voltage | 4.5 to 5.5 Volts |
| Input Voltage (I/O) | 0 to 5.5 Volts |
| Ambient Operating Temperature (free air) | 0° C to +70° C |
| Junction Operating Temperature | 0° C to +115° C |

### Static Characteristics (Supply Pins)

$V_{CC}$ = 4.5V to 5.5V; GND = 0V; Temp = 0 to +70° C; unless otherwise specified

| Symbol | Parameter | Conditions | Min | Typical | Max | Unit |
|---|---|---|---|---|---|---|
| $V_{reg}$ (3.3V) | Regulated Supply Voltage | | 3.0 | 3.3 | 3.6 | V |
| $I_{CC}$ | Operating Supply Current | | - | 18 | - | mA |

### Static Characteristics

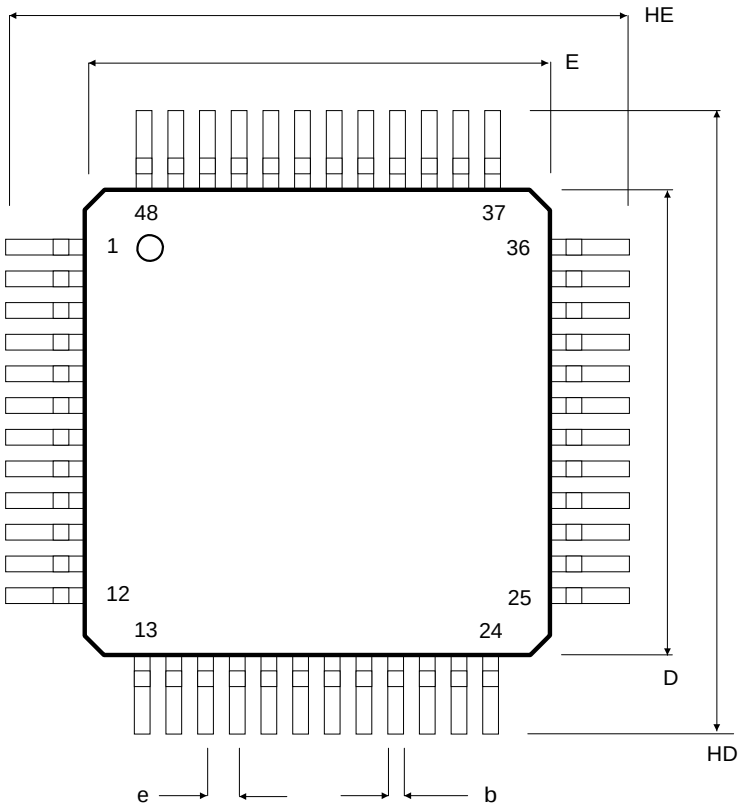
$V_{CC}$ = 4.5V to 5.5V; GND = 0V; Temp = 0 to +70° C; unless otherwise specified

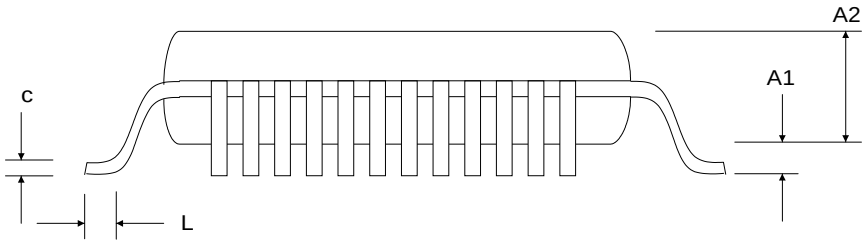
| Symbol | Parameter | Conditions | Min | Typical | Max | Unit |
|---|---|---|---|---|---|---|
| $V_{IL}$ | LOW Level Input Voltage | | - | - | 0.3*Vcc | V |
| $V_{IH}$ | HIGH Level Input Voltage | | 0.7*Vcc | - | - | V |
| $V_{th}$(LH) | Positive going Threshold Voltage | | - | 3.22 | - | V |
| $V_{th}$(HL) | Negative going Threshold Voltage | | - | 1.84 | - | V |
| $I_{LI}$ | Input Leakage Current | | - | - | ±1 | µA |
| $I_{OZ}$ | Tri-State Leakage Current | | - | - | ±10 | µA |
| $V_{OL}$ | Output Voltage (Low ) | | - | - | 0.4 | V |
| $V_{OH}$ | Output Voltage (High ) | | 3.5 | - | - | V |

### Dynamic Characteristics – Analog I/O Pins (DP, DM); Full-Speed Mode

$V_{CC}$ = 4.5V to 5.5V; GND = 0V; Temp = 0 to +70° C; unless otherwise specified

| Symbol | Parameter | Condition | Min | Typical | Max | Unit |
|---|---|---|---|---|---|---|
| $T_{FR}$ | Rise Time | $C_L$ = 50pF<br>10% to 90% of $\|V_{OH} - V_{OL}\|$ | 4 | - | 20 | nS |
| $T_{FF}$ | Fall Time | $C_L$ = 50pF<br>10% to 90% of $\|V_{OH} - V_{OL}\|$ | 4 | - | 20 | nS |




**48-Pin QFP**
**Package Dimensions**

| SYMBOL | MILLIMETERS | | | INCHES | | |
|---|---|---|---|---|---|---|
| | MIN | TYPICAL | MAX | MIN | TYPICAL | MAX |
| **A1** | 0.05 | | 0.15 | 0.002 | | 0.006 |
| **A2** | 1.35 | | 1.45 | 0.053 | | 0.057 |
| **b** | 0.17 | | 0.27 | 0.007 | | 0.011 |
| **c** | 0.09 | | 0.20 | 0.004 | | 0.008 |
| **e** | | 0.50 | | | 0.0197 | |
| **L** | 0.45 | | 0.75 | 0.018 | | 0.030 |
| **HD** | 8.80 | | 9.20 | 0.346 | | 0.362 |
| **D** | 7.20 | | 6.80 | 0.283 | | 0.268 |
| **HE** | 8.80 | | 9.20 | 0.346 | | 0.362 |
| **E** | 7.20 | | 6.80 | 0.2.83 | | 0.268 |

**IMPORTANT NOTICE**



**DISCLAIMER**

No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose, without the express written permission of ASIX. ASIX may make changes to the product specifications and descriptions in this document at any time, without notice.

ASIX provides this document "as is" without warranty of any kind, either expressed or implied, including without limitation warranties of merchantability, fitness for a particular purpose, and non-infringement.

Designers must not rely on the absence or characteristics of any features or registers marked "reserved", "undefined" or "NC". ASIX reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. Always contact ASIX to get the latest document before starting a design of ASIX products.

**TRADEMARKS**

ASIX, the ASIX logo are registered trademarks of ASIX Electronics Corporation. All other trademarks are the property of their respective owners.

## Revision History

| Revision | Date | Comment |
|---|---|---|
| 1.0 | 6-Nov-2002 | Preliminary Release |
| 1.1 | 7-Feb-2005 | Datasheet updated for Serial and Printer port related details |
| 1.2 | 25-Oct-2005 | Revised Data Sheet |
| 1.3 | 17-Nov-2005 | Minor Layout Changes |
| 1.4 | 28-Nov-2005 | Changes to Serial and Parallel Port Descriptors |
| 1.5 | 30-Nov-2005 | Corrected Polling Intervals |
| 1.6 | 6-Jan-2006 | Corrected Electrical Characteristics and Pin Descriptions |
| 2.00 | 2011/08/05 | 1. Changed to ASIX Electronics Corp. logo, strings and contact information.<br>2. Added ASIX copyright legal header information.<br>3. Modified the Revision History table format.<br>4. Updated the block diagram. |
| 2.01 | 2011/11/01 | 1. Updated the ordering information. |

ASIX Electronics Corporation.
4F, No. 8, Hsin Ann Rd., HsinChu Science Park,
HsinChu, Taiwan, R.O.C.

TEL: 886-3-5799500
FAX: 886-3-5799558

Sales Email: sales@asix.com.tw
Support Email: support@asix.com.tw
Web: http://www.asix.com.tw