FT8U100AX Seven Port USB Hub Controller with Embedded Peripheral Functions

Data Sheet

Future Technology Devices International Limited

St. George’s Studios
93/97 St. George’s Road
Glasgow G3 6JA
UK

Friday, 31 March 2000

High Integration 7- Port USB Hub Controller with embedded peripheral functions

Table of Contents

APPLICATION SCHEMATICS – current application schematics for the FT8U100AX can be found on FTDI's home page at http://www.ftdi.co.uk. follow the links from the Products or Support pages.

High Integration 7- Port USB Hub Controller with embedded peripheral functions

1. Introduction

1.1 Features

- High integration USB hub controller
- 7 x Downstream and 1 x Upstream USB Hub Ports
- Embedded Peripherals Including …
  - RS232 Serial Port
  - PS/2 Keyboard Port
  - PS/2 Mouse Port supporting –
    - 2 Button PS/2 Mouse
    - 3 Button PS/2 Mouse
    - Thumb-Wheel PS/ 2 Mouse
  - IrDA SIR Port
  - Ir Remote Control Port
  - Master/Slave 2-Wire Serial Bus
- 2 Status LEDs per USB Port indicates
  - High / Low Speed Peripheral
  - USB Peripheral Bus Traffic
  - Peripheral Over-Current Error
  - Peripheral Babble / EOF Error
  - USB Reset Condition
  - USB Suspend Condition
- Power Control / Overload detect on each individual USB port
- External Firmware in OTP ROM
- 3.3v Supply Operation
- UHCI / OHCI Compliant
- USB 1.1 Specification Compliant
- 100 Pin PQFP package

1.2 General Description

The FT8U100AX is a 7-Port stand-alone USB hub controller IC with firmware in external OTP ROM for maximum flexibility and product differentiation. It's high integration includes an advanced USB 7-Port Hub controller, EMCU micro-controller core and a variety of embedded peripheral ports allowing a wide variety of legacy and other peripherals to be connected to and controlled by a USB enabled system through a FT8U100AX based hub.

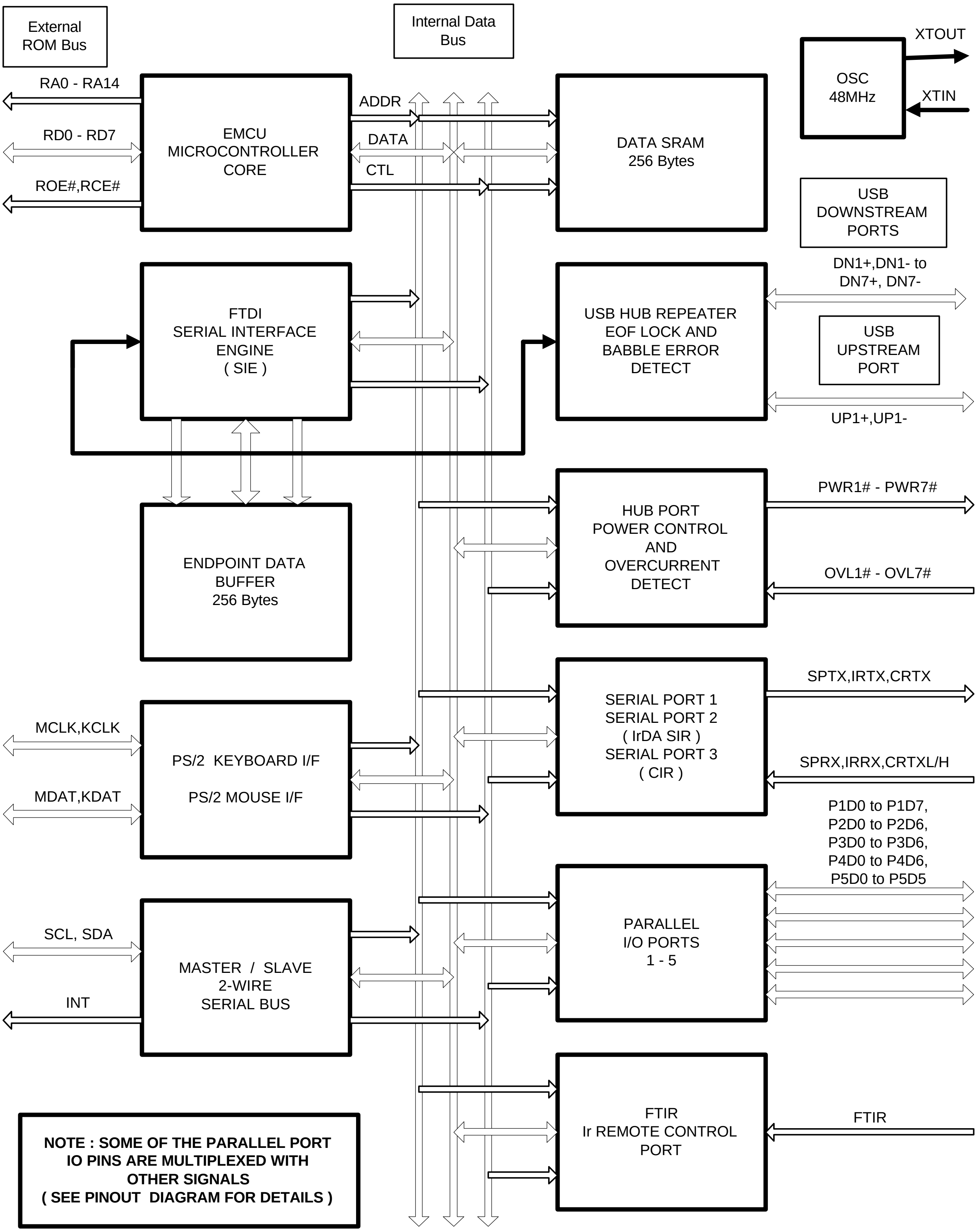
The FT8U100AX has power control and over-current detect on each of the 7 downstream ports and features 2 status LEDs per port providing a useful diagnostic aid for peripherals plugged into the hub ports.

A wide range of pre-tested firmware is available from FTDI including 4 and 7 Port hub solutions with support for legacy keyboards and mice and RS232 serial peripherals. Legacy keyboards and mice plugged into the PS/2 ports are translated by the into USB HID 1.0 compliant protocol whilst firmware and Windows '98 drivers are available for the RS232 serial port function.

A range of tested standard reference designs is available to compliment the firmware and ensure ease of design and fast time to market.

The FT8U100AXintegrates all the above functions into a single cost effective 100 pin PQFP package, requiring only external power switches and a few components to complete the design.

1.3 FT8U100AX Block Diagram



External ROM Bus
Internal Data Bus
XTOUT
OSC 48MHz
XTIN
RA0 - RA14
RD0 - RD7
ROE#,RCE#
EMCU MICROCONTROLLER CORE
ADDR
DATA
CTL
DATA SRAM 256 Bytes
USB DOWNSTREAM PORTS
DN1+,DN1- to DN7+, DN7-
FTDI SERIAL INTERFACE ENGINE (SIE)
USB HUB REPEATER EOF LOCK AND BABBLE ERROR DETECT
USB UPSTREAM PORT
UP1+,UP1-
ENDPOINT DATA BUFFER 256 Bytes
HUB PORT POWER CONTROL AND OVERCURRENT DETECT
PWR1# - PWR7#
OVL1# - OVL7#
SERIAL PORT 1 SERIAL PORT 2 (IrDA SIR) SERIAL PORT 3 (CIR)
SPTX,IRTX,CRTX
SPRX,IRRX,CRTXL/H
MCLK,KCLK
MDAT,KDAT
PS/2 KEYBOARD I/F
PS/2 MOUSE I/F
P1D0 to P1D7, P2D0 to P2D6, P3D0 to P3D6, P4D0 to P4D6, P5D0 to P5D5
PARALLEL I/O PORTS 1 - 5
SCL, SDA
INT
MASTER / SLAVE 2-WIRE SERIAL BUS
FTIR Ir REMOTE CONTROL PORT
FTIR
NOTE : SOME OF THE PARALLEL PORT IO PINS ARE MULTIPLEXED WITH OTHER SIGNALS (SEE PINOUT DIAGRAM FOR DETAILS)

1.4 Block Summary

EMCU Microcontroller Core - FTDI's 8 bit Embedded Microcontroller provides an easy to program , efficient Harvard architecture CPU with 256 bytes of data memory (SRAM). The op-codes are optimised for USB applications and the EMCU includes several standard on board peripherals including serial I/O with modem control and baud rate generator, ISA bus peripheral interface and PS/2 keyboard and mouse interface.

The data sheet for the FTDI EMCU microcontroller core is available on request.

OSC 48MHz – The FTF8U100AX supports the use of either a 48 MHz oscillator or a crystal. , as preferred based on cost versus additional tuning circuit logic. Appendix B contains reference schematics using a 48MHz crystal – including a bandpass filter network and DC blocking capacitor (L1 and C4 with C5 on UH8340 application schematic in Appendix B) advised for reliable operation at 48MHz. Alternatively a 48MHz oscillator can be connected to XTIN using a 10 Ohm series resistor and with the option to add a smoothing capacitor to ground.

Data SRAM - 256 bytes of local variable memory.

FTDI Serial Interface Engine - the FT8U100AX contains an SIE block which provides the following USB functions - integral DPLL Data Separator, NRZI Encode/Decode, Bit Stuff/Unstuff, Sync Packet Generation and Detection, EOP and Bus Reset Detect, Parallel <-> Serial Conversion, CRC checking and generation, PID decode and Validation, Packet Level Handshaking, Suspend/Resume Detection and Signalling

USB Hub Repeater - this block provides the data routing functionality for the FT8U100AX. This provides, on a per-port basis, Connect and Disconnect functions, Full / Low Speed Signal Routing, Power Control and Overcurrent Detect, Suspend and Resume Control as well as Babble detection, End Of Frame Timer and Lock Generation, Low Speed "Keep Alive Strobe" Generation and Low Speed Data Conversion and Pre-Packet Decode.

Hub Port Power Control - this block provides Power Control and Overcurrent Detect functions. The FTDI USB Hub firmware implements several features aimed at improving USB Hub product performance and error reporting in situations where third party peripherals cause power glitches and USB current violations.

Endpoint Data Buffer - 256 byte data buffer with expandable endpoint count, providing support for USB devices within the IC. This provides endpoint buffer management, transmit and receive counters and address to endpoint decoding. The operation of this block is discussed further in section 4.4, USB Device/EndPoint Registers and Appendix A, part 2 – where EndPoint Data Buffer organisation for the UltraHub reference design is discussed.

Parallel I/O Ports - this FT8U100AX provides several General Purpose I/O Ports. These bidirectional ports can be used to tailor the functionality of Hub products using the F8U100AX - providing firmware control of such features as LED status indicator operation and providing pins to support the wide range of legacy peripherals supported by the FT8U100AX.

The FT8U100AX contains hardware to support a range of internal peripherals. The necessary interface signals for these internal peripherals are routed to specific I/O pins under the control of firmware. The Pinout Configuration Diagram provides details on which I/Os can be used to support each of these peripherals.

PS/2 Keyboard and Mouse I/F - provides support for legacy PS/2 keyboard and mouse ports. PS/2 mouse I/F includes support for 2 button and 3 button as well as Thumb Wheel operation.

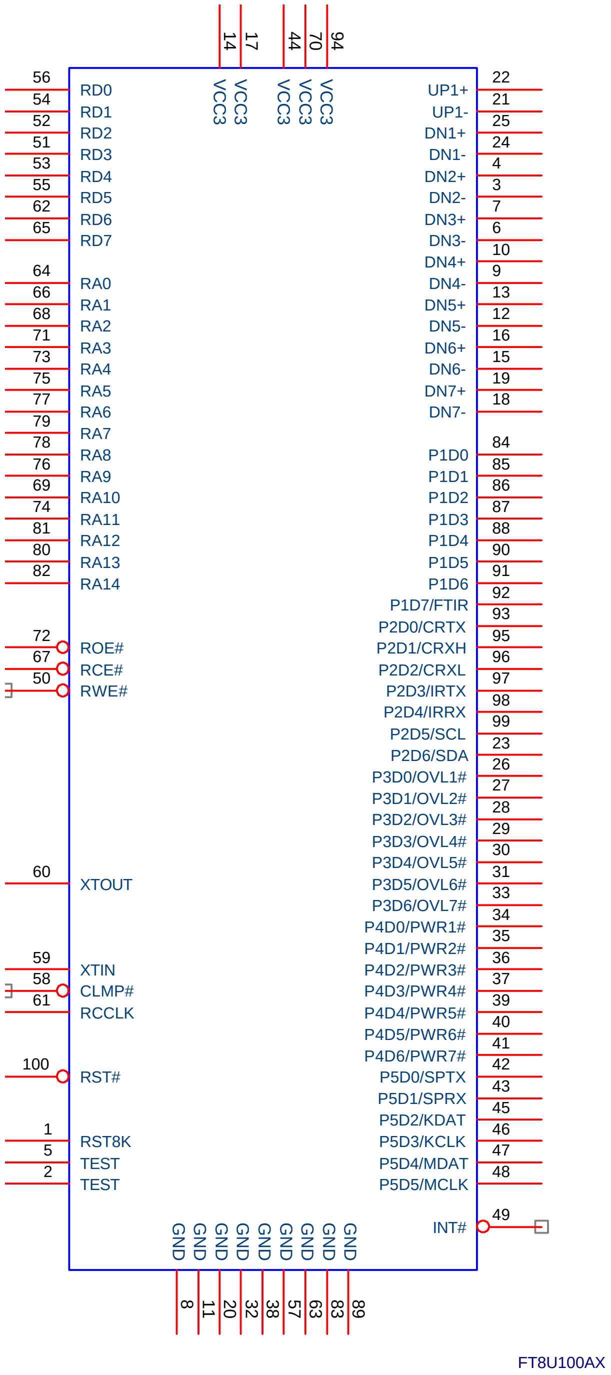
Serial Port 1/2/3 - Serial Port 1 provides an asynchronous serial port for USB to RS232 communications, Serial Port 2 provides support for USB to IrDA SIR port infra red communications and Serial Port 3 provides hardware support for USB to IrDA CIR port infra red communications.

Master/Slave 2-wire serial bus - this provides hardware support for a generic two wire serial bus suitable for monitor control applications.

FTIr Remote Control Port - this provides support hardware for USB to FTIr port for remote control applications.

2. FT8U100AX – Package & Pinout

2.1 FT8U100AX - Pinout Configuration Diagram

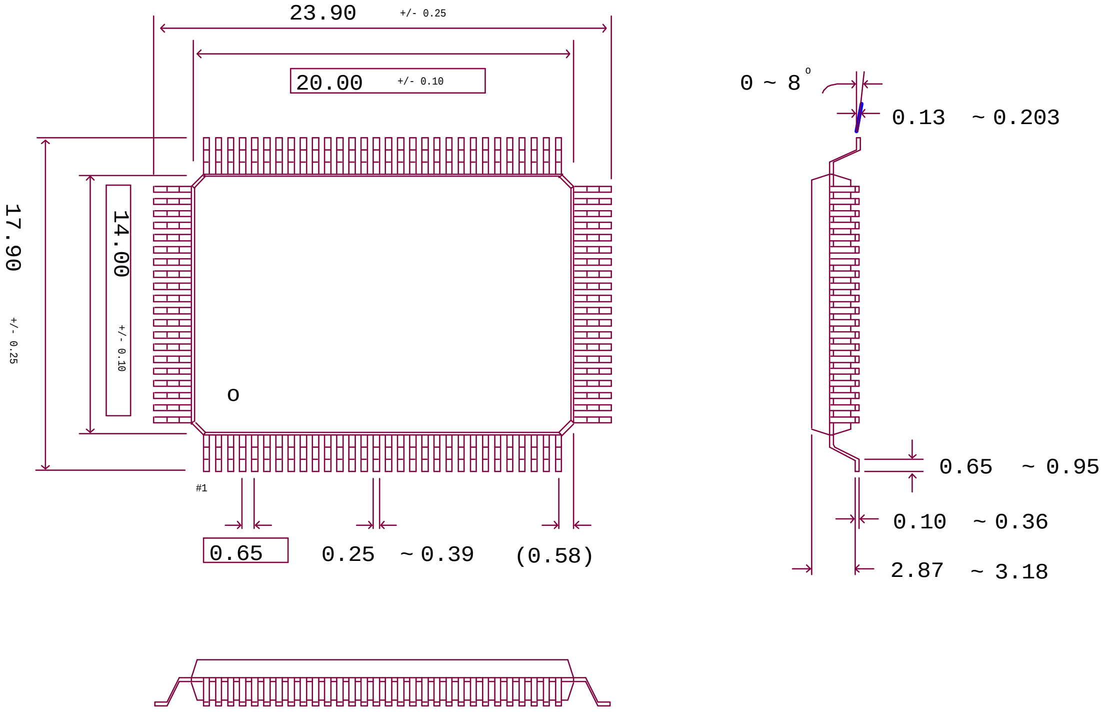


2.2 FT8U100AX - Pinout Description

Pin Number(s)	Signal Name	I/O	Signal Description
22,21	UP1+, UP1-	USB I/O	Differential USB port Connection to Upstream System or Hub
25,24 4,3 7,6 10,9 13,12 16,15 19,18	DN1+, DN1- DN2+, DN2- DN3+, DN3- DN4+, DN4- DN5+, DN5- DN6+, DN6- DN7+, DN7-	USB I/O	Differential USB port Connection to Downstream Peripherals or Hubs
34-37,39,40	PWR1# to PWR7#	I/O	Active Low Power Enable output pin to each USB Port. These signals connect to the gate input of a Power MOSFET or Power Control Switch I.C.
26-31,33	OVL1# to OVL7#	I/O	Active Low Current Overload Detect Input from each USB Port. Pulling these signals low remove the power to the faulty USB device and notifies the system of this event.
84-88, 90-92 93, 95-99,23 26-31,33 34-37,39-41 42,43,45-48	P1D0 to P1D7, P2D0 to P2D6, P3D0 to P3D6, P4D0 to P4D6, P5D0 to P5D5	I/O	General Purpose IO Ports used for controlling LED status indicators etc. Some of these pins are multiplexed with other embedded functions – see Pinout Configuration
42,43	SPTX, SPRX	I/O	Transmit / Receive Data for Serial Port 1
97,98	IRTX, IRRX	I/O	Transmit / Receive Data for Serial Port 2 and IrDA SIR
93,96,95	CRTX, CRXL,CRXH	I/O	Transmit / Receive Data for Serial Port 3 and IrDA CIR
46,45	KCLK, KDAT	I/O	PS/2 Keyboard interface signals
48,47	MCLK, MDAT	I/O	PS/2 Mouse interface signals
99,23 49	SCL,SDA INT#	I/O	Monitor Control port clock and data signals.
92	FTIR	I/O	Ir Remote Control Receiver Port
64,66,68,71,73, 75,77,79,78,76, 69,74,81,80,82	RA0 to RA14	O	Program Address Bus from the internal EMCU micro-controller. These connect to the address pins on an external ROM.
56,54,52,51, 53,55,62,65	RD0 to RD7	I/O	Program Data Bus from the internal EMCU micro-controller. These connect to the address pins on an external ROM.
72,67,50	ROE#, RWE#, RCE#	O	Output Enable, Write Enable and Chip Enable control signals (active low) for the external ROM.
61	RCCLK	O	Oscillator Re-Start Timer RC network. This gates off the oscillator internal to the chip until the Oscillator circuit has become stable.
2,3	TEST	I	For IC Test Purpose Only – Strap to GND
1	RST8K	I	For FTDI In-Circuit Debugger Use Only – Strap to GND for normal operation.
58	CLMP#	I	Strapped to XTIN for Bus powered applications – used to shut down the oscillator circuit during suspend by clamping the input to ground.
59,60	XTIN, XTOUT	I,O	48MHz Crystal Oscillator Input / Output Pins
100	RST#	I	Active Low Chip Reset Pin
14,17,44,70,94	VCC3	VCC	+3.3 Volt Supply Voltage
8,11,20,32,38, 57,63,83,89	GND	GND	Ground Pin

The FT8U100AX contains six general purpose I/O ports which are firmware configurable. These support LED status control, USB overload current detect, per port USB power enable functions as well as internal legacy peripherals including serial port, PS/2 keyboard and mouse. The pinout diagram on the preceding page details the designation of these I/O ports for one specific configuration of the FT8U100AX, as used with the UH8370 USB Hub reference design.

2.3 FT8U100AX – Package Mechanical Drawing



ALL DIMENSIONS IN MILLIMETRES

3. Electrical Specifications

3.1 Absolute Maximum Ratings

Parameter	Min	Max	Units
Storage Temperature	-65	+150	^{o}C
Ambient Temperature (Power Applied)	0	+70	^{o}C
VCC3 Supply Voltage	-0.5	+4.5	V
DC Input Voltage - Inputs	-0.5	VCC3+0.5	V
DC Input Voltage - High Impedance Bidirectionals	-0.5	VCC3+0.5	V
DC Output Current – Outputs		24	mA
DC Output Current – Low Impedance Bidirectionals		24	mA
Power Dissipation		500	mW

3.2 DC Characteristics (Ambient Temperature = 0 .. 70 Degrees C)

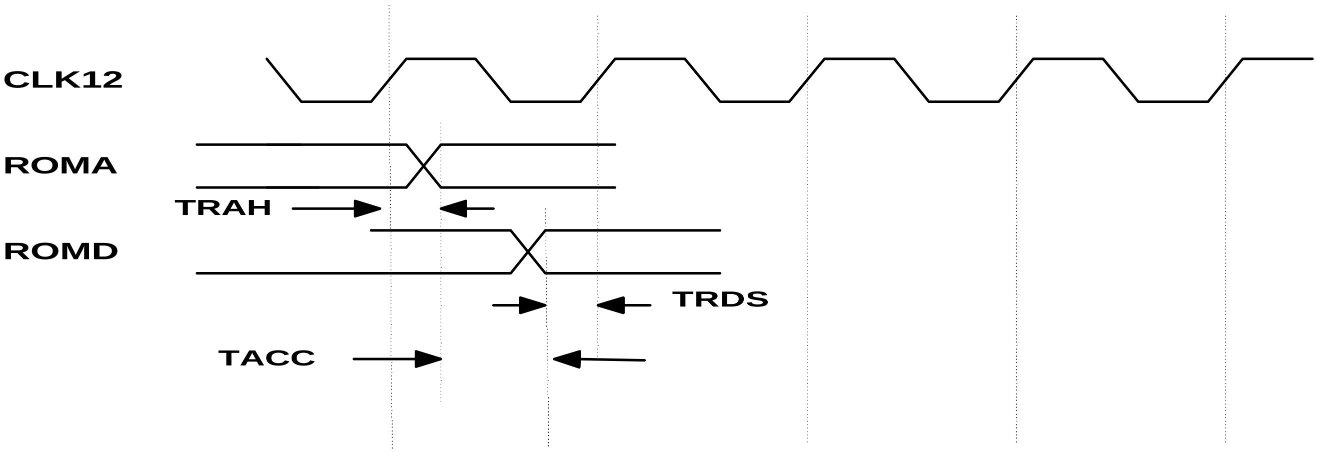
Symbol	Description	Min	Max	Units	Conditions
VCC3	Operating Supply Voltage	3.0	3.6	v	
Icc1	Operating Supply Current		50	mA	Normal Operation
Icc2	Operating Supply Current		250	uA	USB Suspend
Ioh1	Digital IO Pins Source Current	6		mA	Voh = VCC3 – 0.5v
Iol1	Digital IO Pins Sink Current	12		mA	Vol = + 0.5v
Ioh2	Digital Output Pins Source Current	2		mA	Voh = VCC3 – 0.5v
Iol2	Digital Output Pins Sink Current	4		mA	Vol = + 0.5v
Voh1	Input Voltage Threshold (Low)		0.6	v	VDD=3.0V, RL = 1.5K to VDD
Vol1	Input Voltage Threshold (High)	2.7		v	VDD=3.0V, RL = 1.5K to VDD
VDif	USB Differential Input Sensitivity	0.2		v	VDD = 3.0 to 3.6V
VCom	USB Differential Common Mode	0.8	2.5	v	VDD = 3.0 to 3.6V
URxt	USB Single Ended Rx Threshold	0.8	2.0	v	VDD = 3.0 to 3.6V
UVh	USB IO Pins Static Output (Low)		0.3v		Rl = 1.5k to 3.6v
UVl	USB IO Pins Static Output (High)	2.8			Rl = 1.5k to GND
VHYS	Hysteresis on SE receiver inputs	0.1v		0.2v	VDD = 3 to 3.6V
VIHSE	High threshold for SE receiver	1.2	1.5	1.8	VDD = 3 to 3.6V
VILSE	Low threshold for SE receiver	1.0	1.3	1.5	VDD = 3 to 3.6V

3.3 AC Switching Characteristics

The FT8U100AX operates off of a 48MHz input frequecy. This is divided by 4 and passed to the chip's on board microcontroller, FTDI's EMCU, as a 12MHz input clock. The user does not have visibility of this clock however it is useful to relate timings to this clock and as such it is inlcuded.

The key timings for the FT8U100AX relate to ROM access and reset operations.

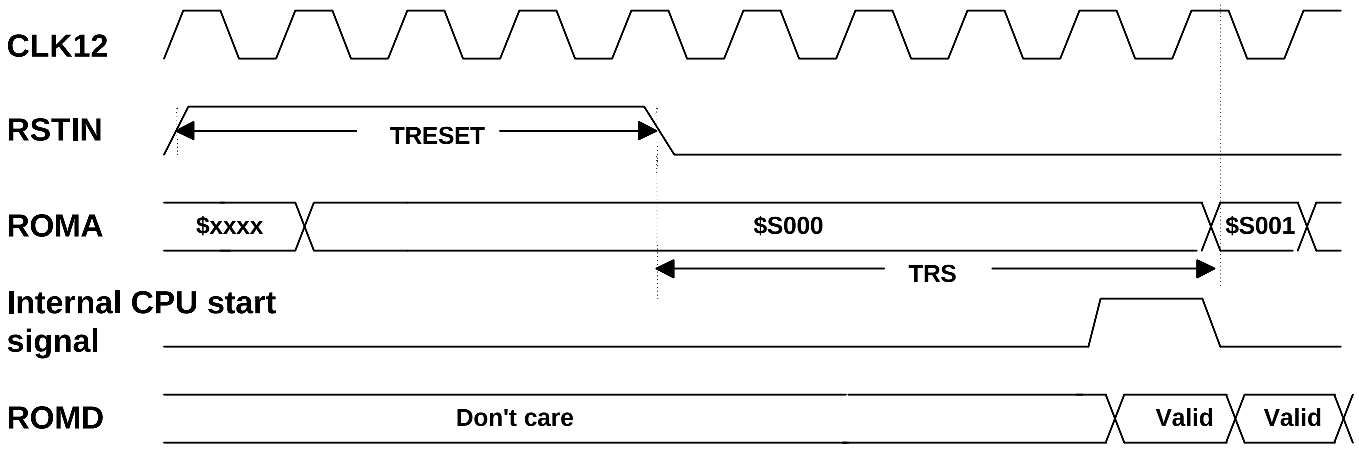
ROM access wait states can be set from 0 to 3 wait states under the control of the Chip Control Register at address CCh.



Program ROM data read (zero wait state)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
TRAH	Address valid from CLK rising	VCC3 = 3.0 to 3.6V			25	nS
TRDS	ROM data setup to CLK	VCC3 = 3.0 to 3.6V			10	nS
TACC	ROM access time	VCC3 = 3.0 to 3.6V			55	nS

The above table shows the case for a zero wait state program ROM read - where the read must take place within 80nS. This requires a ROM access time of 55nS. For One wait state operation 125nS parts can be used.



Reset operation (zero wait states ROM read)

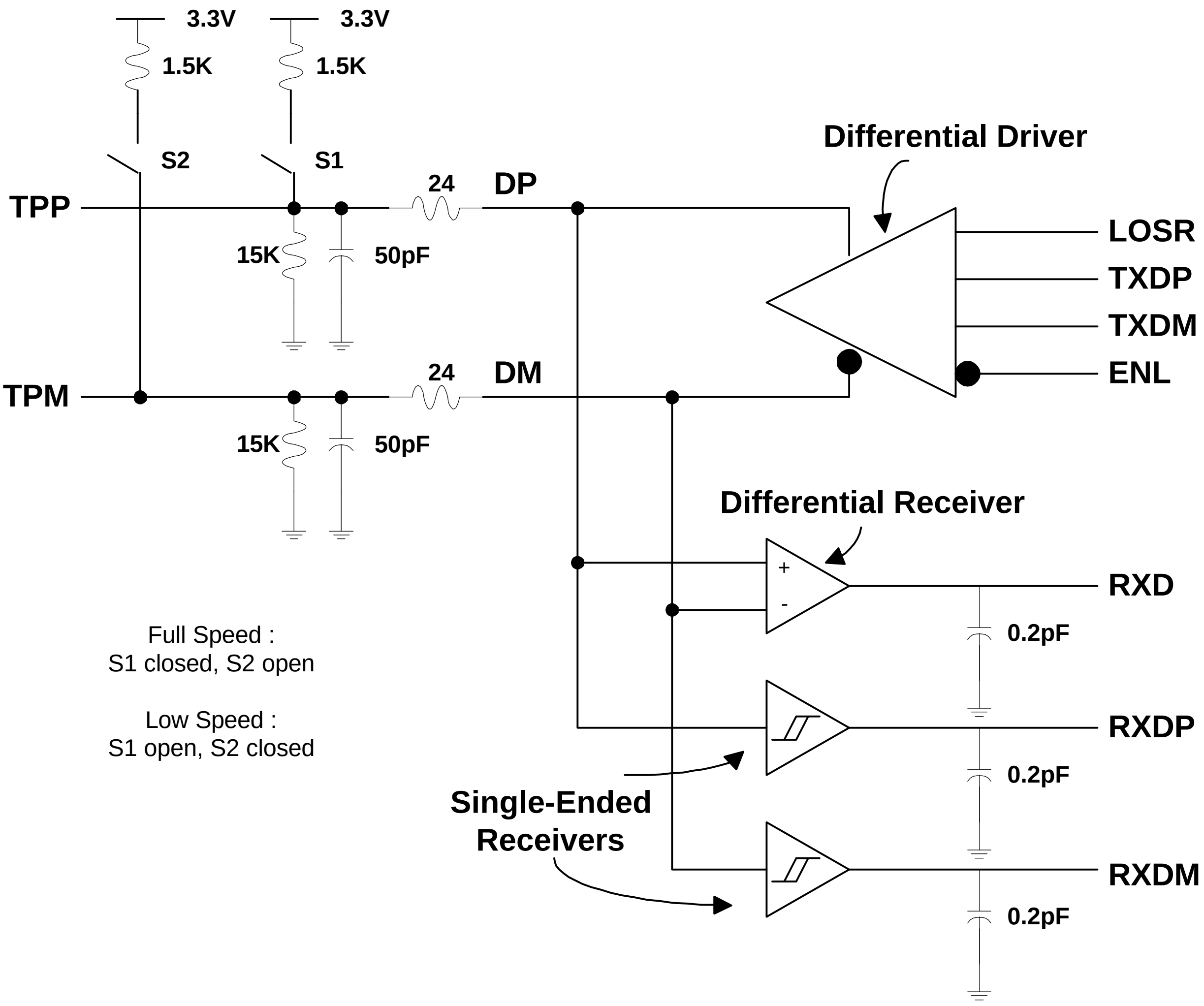
Symbol	Parameter	Conditions	Min	Typ	Max	Units
TRESET	RSTIN pulse width	VCC3 = 3.0 to 3.6V	16			Clk48 cycles
TRS	RSTIN low to ROM address settling	VCC3 = 3.0 to 3.6V	16			Clk48 cycles

The FT8U100AX should be given a reset pulse, TRESET, of at least 16 clock cycles to ensure stability. During reset the 48MHz clock signal must be stable.

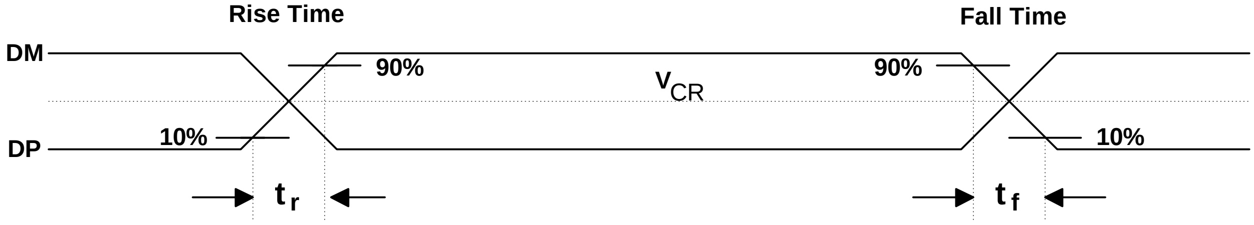
TRS represents the time required for the FT8U100AX to begin stable operation after the removal of the reset pulse – and can be referenced to the setting of ROM address bit 0. This time is set at a minimum of 16 Clk48 cycles – well within the USB specification's reset recovery requirement of 10mS for Hubs to be able to accept requests after the removal of reset.

3.4 USB Transceiver A/C Characteristics

The FT8U100AX contains one upstream and seven downstream USB transceiver I/Os. The figure below includes a functional block diagram of these cells. Each USB I/O comprises a differential driver, a differential receiver and two single-ended receivers.



The following figure describes the key parameters of the differential driver – values for these parameters for high and low speed operation are listed in the tables overleaf. The abovefigure includes details of the loading for this testing.



Differential Driver Characteristics – High Speed

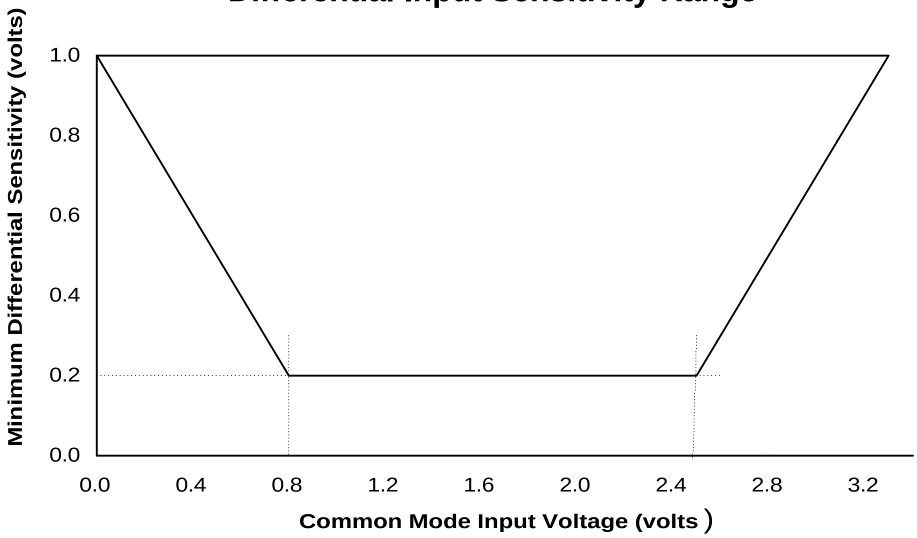
Symbol	Parameter	Conditions	Min	Typ	Max	Units
tr	Rise time	VDD = 3.0 to 3.6V CL = 50pF	8		12	nS
tf	Fall time	VDD = 3.0 to 3.6V CL = 50pF	8		12	nS
VCR	Cross Over Voltage	VDD = 3.0 to 3.6V CL = 50pF		1.8		V
trfmch	Rise/Fall time matching	VDD = 3.0 to 3.6V CL = 50pF	90		110	%

Differential Driver Characteristics – Low Speed

Symbol	Parameter	Conditions	Min	Typ	Max	Units
tr	Rise time	VDD = 3.0 to 3.6V CL = 50pF	85		215	nS
tf	Fall time	VDD = 3.0 to 3.6V CL = 50pF	95		255	nS
VCR	Cross Over Voltage	VDD = 3.0 to 3.6V CL = 50pF		1.6		V
trfmch	Rise/Fall time matching	VDD = 3.0 to 3.6V CL = 50pF	80		120	%

The differential receiver must exhibit input sensitivity of at least 200mV when both differential data inputs are in the differential common mode range of 0.8 to 2.5V as shown above. These values are shown in the DC Characteristics table on page 9.

Differential Input Sensitivity Range



The single ended receivers are standard CMOS buffers with adjusted thresholds and which exhibit hysteresis – necessary to reduce sensitivity to noise (see VHYS on the DC Characteristcs table, page 9).

4. Internal IO Register Summary

The registers contained within the FT8U100AX can be described in terms of the following five main groups

USB Downstream Registers
USB Suspend Registers
USB Upstream Registers
USB Device/EndPoint Registers
Chip Control Registers

A summary of the function of each register group is given followed by tables detailing the register contents. Where necessary a short description of the function of specific registers is provided.

4.1 USB Downstream Registers

The following registers provide indexing, control and status information for up to seven USB downstream ports supported by the FT8U100AX.

This register acts as an index into the downstream port data registers - which provide control and status for even and odd indexes respectively.

USB Hub Downstream Port index register **Address = 90h w/o**
0/1 – Select Port 1
2/3 – Select Port 2
4/5 – Select Port 3
6/7 – Select Port 4
8/9 – Select Port 5
A/B – Select Port 6
C/D – Select Port 7

USB Hub Downstream Port Data register **Address = 91h r/w**
Even index – control
Bit 0 – Set Power On
Bit 1 – Set Port Enable
Bit 2 – Set Port Reset
Bit 3 – Set Port Suspend
Bit 4 – Set Power OK
Bit 5 – Set Reset Activity Detect
Bit 6 – Set Signal Resume
Bit 7 – Set SE0 (set single ended zero)
odd index – status
Bit 0 - Connect Detected
Bit 1 - Port Enabled
Bit 2 - Port Suspended
Bit 3 - Port Over Current detected
Bit 4 - Port Reset
Bit 5 - Enable Power Interrupt
Bit 6 - NOT Full Speed Device
Bit 7 - Resume Signalled

USB Hub Downstream Port Control register **Address = 92h r/w**
Bit 0 - Frame timer enable
Bit 1 - Hub Suspend
Bit 2 - Signal Resume
Bit 3 - Three bit window for seeing SOF
Bit 4 - IRQMask masks Host resume IRQ
Bit 5 - Force EOF2 – End of Frame2 set 10 clocks before the start of frame

Bit 6 - '0'
Bit 7 - Host Resume - read only

USB Hub Downstream Port Activity register **Address = 93h r/o**
Bit 0 - Frame Timer Locked
Bit 1 - Activity Detected on port 1
Bit 2 - Activity Detected on port 2
Bit 3 - Activity Detected on port 3
Bit 4 - Activity Detected on port 4
Bit 5 - Activity Detected on port 5
Bit 6 - Activity Detected on port 6
Bit 7 - Activity Detected on port 7

EOF2 is generated by the Frame Timer 10 clocks prior to the End Of Frame. This is provided as a local frame reference when the Host is in suspend, and EOF is not being generated. This register indicates the state of the USB lines on each downstream port at EOF2.

USB Hub Downstream Port EOF2 Data Line State register **Address = 94h r/o**
Bit 0 - port1 D-
Bit 1 - port1 D+
Bit 2 - port2 D-
Bit 3 - port2 D+
Bit 4 - port3 D-
Bit 5 - port3 D+
Bit 6 - port4 D-
Bit 7 - port4 D+

USB Hub Downstream Port EOF2 Data Line State register **Address = 95h r/o**
Bit 0 - port5 D-
Bit 1 - port5 D+
Bit 2 - port6 D-
Bit 3 - port6 D+
Bit 4 - port7 D-
Bit 5 - port7 D+
Bit 6 - '0'
Bit 7 - '0'

Enable PWR switch lines for I/O port muxing **Address = 96h**
Bit 0 - Enable power on port1
Bit 1 - Enable power on port2
Bit 2 - Enable power on port3
Bit 3 - Enable power on port4
Bit 4 - Enable power on port5
Bit 5 - Enable power on port6
Bit 6 - Enable power on port7
Bit 7 - Force Frame Lock

IRQ Enable for Downstream HUB PORTS **Address = 97h**
Bit 0 - EN PORT 1 IRQ
Bit 1 - EN PORT 2 IRQ
Bit 2 - EN PORT 3 IRQ
Bit 3 - EN PORT 4 IRQ
Bit 4 - EN PORT 5 IRQ
Bit 5 - EN PORT 6 IRQ
Bit 6 - EN PORT 7 IRQ

4.2 USB Suspend Registers

The following registers provide support for suspend and resume. To clarify the terminology used asynchronous resume returns the USB from suspend when a K state on a downstream port is signalled and passed directly to the Host without the intervention of the Hub – this would be the case for example with a bus powered hub in which the clocks are stopped during suspend. Remote wakeup on the other hand requires the active intervention of the Hub.

WakeUp / Resume Control Register 2 **Address = 9Ch**
Command register 2 :
Bit 0 - Enable asynchronous resume
Bit 1 - Enable remote wakeup
Bit 2 -
Bit 3 -
Bit 4 -
Bit 5 -
Bit 6 -
Bit 7 -

Resume detection for async resume **Address = 9Dh**
Bit 0 - Resume on PORT 1
Bit 1 - Resume on PORT 2
Bit 2 - Resume on PORT 3
Bit 3 - Resume on PORT 4
Bit 4 - Resume on PORT 5
Bit 5 - Resume on PORT 6
Bit 6 - Resume on PORT 7
Bit 7 -

4.3 USB Upstream Registers

The following group provides the necessary status and control registers to support the USB upstream port.

USB Main status register **Address = A0h r/o**
bit 0 - IRQ for Start of Frame
bit 1 - IRQ for Receiver full
bit 2 - IRQ for Transmitter empty
bit 3 - Bus Idle
bit 4 - Reset received from USB I/F
bit 5 -
bit 6 -
bit 7 - resume out detected form Host

USB Main Control register **Address = A1h r/w**
bit 0 - Reset Start Of Frame Interrupt when 0, Enable interrupt when 1
bit 1 - Reset Receive Buffer Full Interrupt when 0, Enable interrupt when 1
bit 2 - Reset Transmit Buffer Empty Interrupt when 0, Enable interrupt when 1
bit 3 - Reset Bus Idle status bit when 0
bit 4 - Clock Stop Request
bit 5 - Enable interrupt on Resume
bit 6 - Force Bus to signal Resume
bit 7 – Suspend In (to the SIE)

4.4 USB Device/Endpoint Registers

This group provides device and frame addresses as well as endpoint index registers. The operation of these ties in with the explanation of the EndPoint organisation in Appendix A, part 2.

USB Device register **Address = A2h r/w**
bit 0 – Device Address 0
bit 1 - Device Address 1
bit 2 - Device Address 2
bit 3 - Device Address 3
bit 4 - Device Address 4
bit 5 - Device Address 5
bit 6 - Device Address 6
bit 7 - '0'

USB End-Point Enable register Address = A3h r/w
Bit 0 - Enable for End Point 0 - no effect always enabled
bit 1 - Enable for End Point 1
bit 2 - Enable for End Point 2
bit 3 - Enable for End Point 3
bit 4 - '0'
bit 5 - '0'
bit 6 - '0'
bit 7 - '0'

A sixty four byte buffer is available and can be switched between device 0 and device 1 under the control of bit 6 of the Endpoint Configuration register.

USB Device Endpoint configuration **Address = A6h w/o**
bit 0 - device address bit 0
bit 1 - device address bit 1
bit 2 - '0'
bit 3 - '0'
bit 4 - Enable device 1
bit 5 - Enable device 2
bit 6 - 64 bytes on dev 0 = 1 /64 bytes on dev 1 = 0
bit 7 - '0'

USB EndPoint Buffer Index register **Address = A8h r/w**
Write:
bit 0 - End Point Address bit 0
bit 1 - End Point Address bit 1
bit 2 - Access Receive Region when 0, transmit region when 1
bit 3 - Reset Receive Buffer Full when 0
bit 4 - '0'
bit 5 - '0'
bit 6 - '0'
bit 7 - '0'
Read:
bit 0 - End Point Address bit 0
bit 1 - End Point Address bit 1
bit 2 - CPU Write to SRAM in progress
bit 3 - CPU Read to SRAM in progress
bit 4 - '0'
bit 5 - '0'
bit 6 - '0'
bit 7 - '0'

USB EndPoint Status register **Address = AAh r/w**
Write:
bit 0 -
bit 1 -
bit 2 - End Point Stalled
bit 3 -
bit 4 - Data 0 when 0, Data 1 when 1
bit 5 -
bit 6 -
bit 7 -
Read:
bit 0 - ACK for last transmit
bit 1 - Error for last transfer
bit 2 - ACK for last receive
bit 3 - Setup packet
bit 4 - Data 0 when 0, Data 1 when 1
bit 5 - Receive Buffer Full
bit 6 - Transmit Buffer Empty
bit 7 - Transmit Direction: 0 = out(to us) 1 = in (from us)

USB EndPoint Buffer data register **Address = ABh r/w**
Bit 0 – EP Buffer data Bit 0
Bit 1 – EP Buffer data Bit 1
Bit 2 – EP Buffer data Bit 2
Bit 3 – EP Buffer data Bit 3
Bit 4 – EP Buffer data Bit 4
Bit 5 – EP Buffer data Bit 5
Bit 6 – EP Buffer data Bit 6
Bit 7 – EP Buffer data Bit 7

USB EndPoint Buffer Count Low register **Address = ACh r/w**
Read: read low 8 bits of RX count
Write: write low 8 bits of TX count Writing low count register sets the TXFULL in bit for that end point.
Bit 0 – EP Buffer count Bit 0
Bit 1 – EP Buffer count Bit 1
Bit 2 – EP Buffer count Bit 2
Bit 3 – EP Buffer count Bit 3
Bit 4 – EP Buffer count Bit 4
Bit 5 – EP Buffer count Bit 5
Bit 6 – EP Buffer count Bit 6
Bit 7 – EP Buffer count Bit 7

USB EndPoint Buffer CPU Count Low register **Address = AEh r/w**
Read: read low 8 bits of RX count
Write: write low 8 bits of TX count Writing low count register sets the TXFULL in bit for that end point.
Bit 0 – EP Buffer count Bit 0
Bit 1 – EP Buffer count Bit 1
Bit 2 – EP Buffer count Bit 2
Bit 3 – EP Buffer count Bit 3
Bit 4 – EP Buffer count Bit 4
Bit 5 – EP Buffer count Bit 5
Bit 6 – EP Buffer count Bit 6
Bit 7 – EP Buffer count Bit 7

4.5 Chip Control Registers

This group provides on chip interrupt support (timer count, control and prescaler), internal peripheral control registers, in circuit debugger support registers, IRQ source register and support registers for PS/2 keyboard and mouse functions.

Interrupt Timer – Count Low **Address = B1h r/w**
Bit 0 – Timer count Bit 0
Bit 1 – Timer count Bit 1
Bit 2 – Timer count Bit 2
Bit 3 – Timer count Bit 3
Bit 4 – Timer count Bit 4
Bit 5 – Timer count Bit 5
Bit 6 – Timer count Bit 6
Bit 7 – Timer count Bit 7

Interrupt Timer – Count High **Address = B2h r/w**
Bit 0 – Timer count Bit 8
Bit 1 – Timer count Bit 9
Bit 2 – Timer count Bit10
Bit 3 – Timer count Bit 11
Bit 4 – Timer count Bit 12
Bit 5 – Timer count Bit 13
Bit 6 – Timer count Bit 14
Bit 7 – Timer count Bit 15

Interrupt Timer – Prescaler Divisor **Address = B8h r/w**
Bit 0 – Prescaler Divisor Bit 0
Bit 1 – Prescaler Divisor Bit 1
Bit 2 – Prescaler Divisor Bit 2
Bit 3 – Prescaler Divisor Bit 3
Bit 4 – Prescaler Divisor Bit 4
Bit 5 – Prescaler Divisor Bit 5
Bit 6 – Prescaler Divisor Bit 6
Bit 7 – Prescaler Divisor Bit 7

CHIP CONTROL Register – Low **Address = CCh w/o**
Bit 0 - Enable serial port interrupt on DEBUG interrupt – to speed up operation
Bit 1 - Force No Operation code on ROM data
Bit 2 - Force internal Memory Address bus to appear on I/O ports
Bit 3 – 7

CHIP CONTROL Register – High **Address = CCh w/o**
Bit 0 Enable serial port
Bit 1 Enable PS2 mouse
Bit 2 Enable hub function
Bit 3 Enable SIE
Bit 4 Enable on chip timer
Bit 5 Enable PS2 keyboard
Bit 6 ROM wait state 0, bits 6 and 7 encode the number of ROM wait states (0 to 3)
Bit 7 ROM wait state 1

In-Circuit Debugger Support port Registers **Address = CEh (index)** **Address = CFh (Data R/W port)**
Write mode
00
Bit 0 - Enable ROM address debug
Bit 1 - Enable debug on memory address
Bit 2 - Enable debug on memory data
Bit 3 - Enable debug on write cycle
Bit 4 - Enable debug on read cycle
Bit 5 - Enable debug on IO address
Bit 6 -
Bit 7 - Single Step Enable
Read mode
00
Bit 0 - Enable ROM debug
Bit 1 - Enable debug on memory address
Bit 2 - Enable debug on memory data
Bit 3 - Enable debug on write cycle
Bit 4 - Enable debug on read cycle
Bit 5 - Debug generated on Write
Bit 6 - Debug generated on Read
Bit 7 - Single Step Enable
Read/Write
01 - Data bit MASK for trap on data read or write cycles
02 - Rom address low when Enable Rom address debug active
Memory address when Memory address is enabled
03 - Rom address high when Enable Rom address debug active
Memory data when Memory data is enabled
Optional settings are:
00 - 00h - debug disabled
01h - ROM address debug enabled. Address low in 02
Address high in 03.
12h - Debug on memory read of address in 02.
0Ah - Debug on memory write of address in 02.
1Ah - Debug on memory writes or read of address in 02.
14h - Debug on memory read of data in 03.
01h - contains mask for data bits. A 1 includes
That bit a 0 ignores that bit. The bit value is
Put into 03h.
0Ch - Debug on memory write of data in 03. Mask options
Is the same as for read.
1Ch - Debug on memory read or write of data in 03.
Mask options are the same as for read.
16h - Debug on memory read of address in 02 and data in
03. Mask options are the same as for data read.
0Eh - Debug on memory write of address in 02 and data in
03. Mask options are the same as for data read.
1Eh - Debug on memory read or write of address in 02 and
Data in 03. Mask options are the same as for data
Read.
80h - Single Step main program

IRQ Source Register **Address = DFh r/o**
Bit - 0 timer
Bit - 1 serial 1
Bit - 2 serial 2
Bit - 3 SIE
Bit - 4 HUB
Bit - 5 PS/2 Keyboard
Bit - 6 USB Port
Bit - 7 Generic two wire serial interface

PS/2 Keyboard/mouse Control ports **Address = F4h r/w**
Write:
bit 0 - set output shift register to load
bit 1 - set output shift register to shift
bit 2 - invert keyboard clock coming in
bit 3 - reset input shift register and keyboard clock disable
bit 4 - 0 resets port60wr interrupt signal
bit 5 - 0 resets port64wr interrupt signal
bit 6 - 0 resets keyboard send interrupt signal
bit 7 -
Read:
bit 0 - keyboard clock line in
bit 1 - keyboard data line in
bit 2 - Output Buffer Full status 64 bit 0
bit 3 - Input Buffer Full status 64 bit 1
bit 4 - port60wr interrupt signal
bit 5 - port64wr interrupt signal
bit 6 – keyboard send interrupt signal
bit 7 -

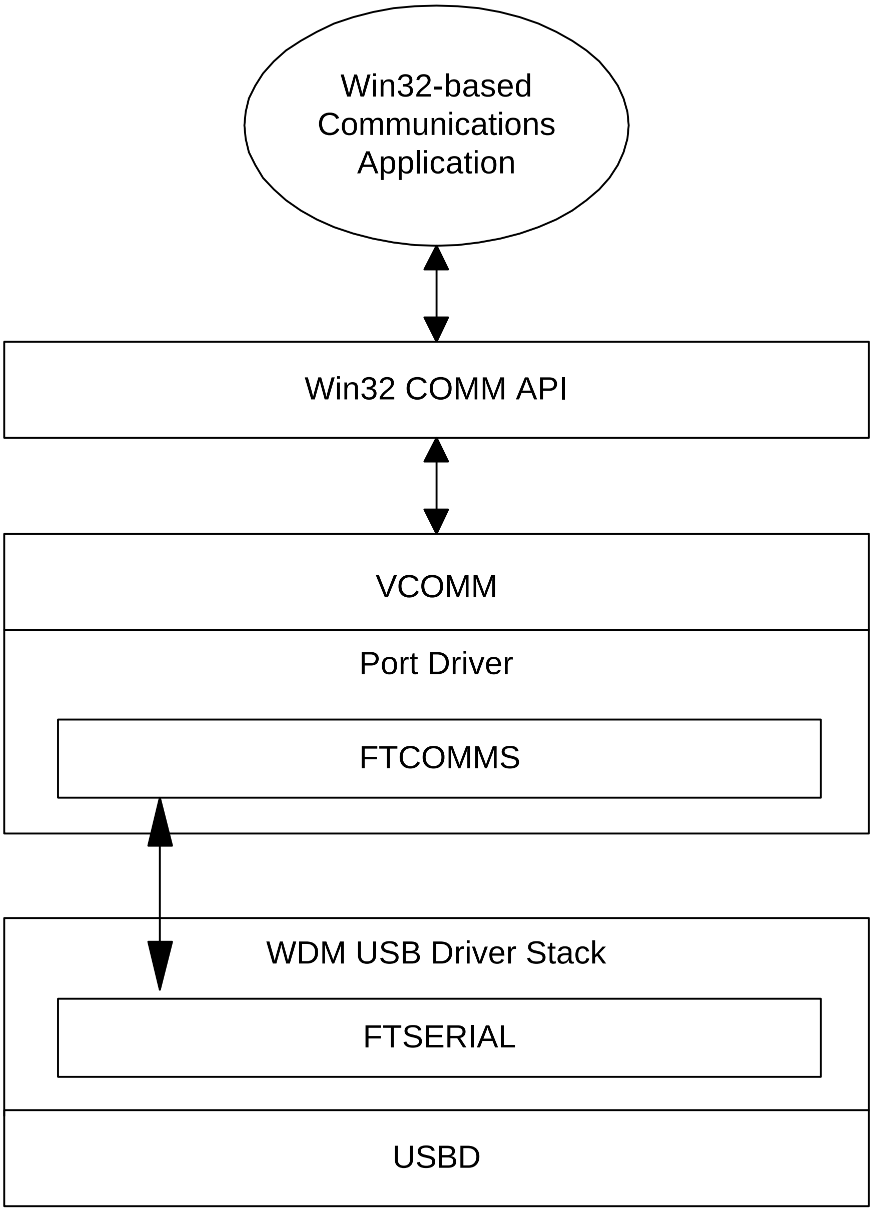
Appendix A - FT8U100AX Internal Peripherals

1. Driver Summary

FTDI USB<->Serial Port Architecture

The USB<-> Serial/Printer Port includes the following modules: -

FTCOMMS.VXD	VCOMM port driver for serial communications.
FTSERIAL.SYS	WDM driver for serial communications.



2. Device End Point Organisation

The following is an example of the organisation of the EndPoint Buffer for legacy devices supported by FTDI firmware. This 256 byte buffer can be organised under the control of USB Device/EndPoint registers – and this example details the devices supported in the UltraHub reference design. The UltraHub comprises PS/2 keyboard and mouse, two serial ports and a printer port.

The devices are controlled by the EndPoint Configuration Register at address A6h.

In the UltraHub the FT8U100AX firmware supports three independent devices – each with separate address registers and EndPoint buffers.

Note all the legacy devices are implemented as full speed, self powered devices.

Device	Endpoint	Transfer Type	Direction	Transfer Size (bytes)	Buffer size (bytes)	Legacy Device
0	0	Control	IN/OUT	8	16	
0	1	Interrupt/Bulk	IN/OUT	8	16	
0	2	Interrupt/Bulk	IN/OUT	8/64	16	

Device	Endpoint	Transfer Type	Direction	Transfer Size (bytes)	Buffer size (bytes)	Legacy Device
1	0	Control	IN/OUT	8	16	
1	1	Interrupt/Bulk	IN/OUT	8	16	Printer
1	2	Interrupt/Bulk	IN/OUT	8/64	128	Serial

Device	Endpoint	Transfer Type	Direction	Transfer Size (bytes)	Buffer size (bytes)	Legacy Device
2	0	Control	IN/OUT	8	16	
2	1	Interrupt	IN/OUT	8	16	PS/2 Kbd & mouse

Note the total Buffer size of 256 bytes – as per the Endpoint Data Buffer in the FT8U100AX Block Diagram of page 4.

3. USB and Legacy Device LED Encoding

The FT8U100AX supports powerful LED-based diagnostic and status features. This provides useful information in an environment where failure isolation can be difficult. FTDI standard firmware supports two LEDs (one yellow, one green) per USB port – in a configuration shown on page 4 of the UltraHub application schematic.

In addition to the USB port LEDs the FT8U100AX supports traffic indicator LEDs for legacy devices. These are switched on when individual legacy devices are configured and flash when there is Bus Traffic from the appropriate device.

The following describes the operation of the LEDs under normal conditions as well as suggesting a basic sequence of steps for locating the likely source of any problems which may be encountered.

On power up

1) All Lights On - hub has been properly reset. LEDs will light up for a short period indicating that the hub has been properly initialised.

2) All Lights Off - USB has enumerated the hub. This indicates that the hub has been succesfully initialised by USB.

On plugging a USB peripheral into a port

3A) Yellow On - LOW Speed USB Peripheral plugged into hub, followed quickly by

4A) Yellow On and Green Twinkles - Bus Traffic from LOW Speed Peripheral connected to hub, OR

3B) Green On - HIGH Speed USB Peripheral plugged into hub, followed quickly by

4B) Green On and Yellow Twinkles - Bus Traffic from HIGH Speed Peripheral connected to hub.

In addition the LEDs indicate specific USB conditions including: USB reset – both LEDs come on continuously and USB suspend – both LEDs go off continuously.

Error conditions

If the lights do not come on when a peripheral is plugged in, the peripheral has not been detected by the USB. Notice that this is different from the USB suspend condition where the peripheral is already functional when the LEDs go off.

Both Green and Yellow Twinkle in Phase - Overcurrent Error Condition. Peripheral has taken too much current and the hub has shut off power to the peripheral.

Both Green and Yellow Twinkle in AntiPhase - Babble Error Condition. Peripheral continues sending data at the end of the USB 1ms Frame interval and the hub has shut off the bus traffic from the peripheral.

Problem Diagnosis - follow this procedure to identify if problem is due to upstream device, hub or downstream device

Step 1) Remove all USB cables (upstream and downstream), leaving only the power cable connected to hub. Switch hub power on. All LEDs should go on as described in 1) on the previous page. If all LEDs go on proceed to step 2. If all the LEDs do not come there may well be a problem with the hub or its power supply.

Step 2) Plug in the upstream USB cable (connected to an upstream device - the system box or another hub). The LEDs should all go out. If all LEDs go off proceed to step 3. If the LEDs do not go out the hub has not been correctly enumerated by the USB. This can be a problem with the upstream device or with the hub.

Replace you hub with a USB peripheral (or another hub if available) to check that the upstream device is functional. If this fails then you may have problem with an upstream hub or with the system box. If this device functions as expected then the hub will need further investigation.

Step 3) If there are no problems with steps 1 and 2 then begin testing you downstream devices by plugging them into the hub one at a time. The LEDs should behave as described in 3) and 4) on the previous page.

If you have a problem with a downstream device try plugging it into another port - in case there is a problem with one port. If you have already found a good device you will have some confidence that the port it used is functioning. If the problem device continues to fail in this port then it is likely that this device is the source of the problem.

Disclaimer & Contact Information

Neither the whole nor any part of the information contained in, or the product described in this manual, may be adapted or reproduced in any material or electronic form without the prior written consent of the copyright holder.

This product and its documentation are supplied on an as is basis and no warranty as to their suitability for any particular purpose is either made or implied. Future Technology Devices will not accept any claim for damages howsoever arising as a result of use or failure of this product. Your statutory rights are not affected.

This product or any variant of it is not intended for use in any medical appliance, device or system in which the failure of the product might reasonably be expected to result in personal injury.

This document provides preliminary information that may be subject to change without notice.

Contact Information

Future Technology Devices International has agents and distributors based in Europe, the US and throughout the Far East. For further information and to obtain local contact details please contact us, preferably by e-mail, at the address below.

Future Technology Devices Intl Limited	Telephone :	+44 (141) 353 2565
St. George’s Studios	Fax :	+44 (141) 353 2656
93/97 St. George’s Road	E-mail :	support@ftdi.co.uk
Glasgow G3 6JA, UK	Internet :	http://www.ftdi.co.uk