Future Technology Devices International Ltd.

Application Note AN_108

Command Processor for MPSSE and MCU Host Bus Emulation Modes

Document Reference No.: FT_000109

Version 1.21

Issue Date: 2010-03-12

This document provides details of the op-codes used to control the Multi Purpose Synchronous Serial Engine (MPSSE) mode of the FT2232D, FT2232H and FT4232H devices.

Future Technology Devices International Limited (FTDI)

Unit1, 2 Seaward Place, Centurion Business Park, Glasgow G41 1HH United Kingdom
Tel.: +44 (0) 141 429 2777 Fax: + 44 (0) 141 429 2758
E-Mail (Support): **support1@ftdichip.com** Web: http://www.ftdichip.com

TABLE OF CONTENTS

1 Overview

The FT2232D, FT2232H and FT4232H incorporate a command processor called the Multi-Protocol Synchronous Serial Engine (MPSSE). The purpose of the MPSSE command processor is to communicate with devices which use synchronous protocols (such as JTAG or SPI) in an efficient manner. The FT2232x and the FT4232H MCU Host Bus Emulation mode also uses the MPSSE technology to make the chip emulate a standard 8048/8051 MCU host bus.
The MPSSE Command Processor unit is controlled using a SETUP command. Various commands are used to clock data out of and into the chip, as well as controlling the other I/O lines.
If disabled the MPSSE is held in reset and will not have any effect on the rest of the chip. When enabled, it will take its commands and data from the OUT data written to the OUT pipe in the chip.
This is done by simply using the normal WRITE command, as if data were being written to a COM port. Any data read will be passed back in the normal IN pipe. This is done using the normal READ command, as if data were being read from a COM port.

NOTE: To ensure that the device driver will not issue IN requests if the buffer is unable to accept data, add a call to FT_SetFlowControl prior to entering MPSSE or MCU Host Bus modes.

2 Data and Clock Definition

2.1 Data bit Definition

The following table shows the pins used and pin functions enabled in MPSSE mode for each device.

Data Bit	Signal	FT2232D Pin	FT2232H Pin		FT4232H Pin		Type	Description
		Channel A	Channel A	Channel B	Channel A	Channel B		
Bit0	TCK/SK	ADBUS0	ADBUS0	BDBUS0	ADBUS0	BDBUS0	output	Clock Signal Output
Bit1	TDI/DO	ADBUS1	ADBUS1	BDBUS1	ADBUS1	BDBUS1	output	Serial Data Out
Bit2	TDO/DI	ADBUS2	ADBUS2	BDBUS2	ADBUS2	BDBUS2	input	Serial Data In
Bit3	TMS/CS	ADBUS3	ADBUS3	BDBUS3	ADBUS3	BDBUS3	output	Select Signal Out
Bit4	GPIOL0	ADBUS4	ADBUS4	BDBUS4	ADBUS4	BDBUS4	(input /output)	General Purpose I/O
Bit5	GPIOL1	ADBUS5	ADBUS5	BDBUS5	ADBUS5	BDBUS5	(input /output)	General Purpose I/O
Bit6	GPIOL2	ADBUS6	ADBUS6	BDBUS6	ADBUS6	BDBUS6	(input /output)	General Purpose I/O
Bit7	GPIOL3	ADBUS7	ADBUS7	BDBUS7	ADBUS7	BDBUS7	(input /output)	General Purpose I/O
Bit8	GPIOH0	ACBUS0	ACBUS0	BCBUS0			(input /output)	General Purpose I/O
Bit9	GPIOH1	ACBUS1	ACBUS1	BCBUS1			(input /output)	General Purpose I/O
Bit10	GPIOH2	ACBUS2	ACBUS2	BCBUS2			(input /output)	General Purpose I/O
Bit11	GPIOH3	ACBUS3	ACBUS3	BCBUS3			(input /output)	General Purpose I/O
Bit12	GPIOH4		ACBUS4	BCBUS4			(input /output)	General Purpose I/O
Bit13	GPIOH5		ACBUS5	BCBUS5			(input /output)	General Purpose I/O
Bit14	GPIOH6		ACBUS6	BCBUS6			(input /output)	General Purpose I/O
Bit15	GPIOH7		ACBUS7	BCBUS7			(input /output)	General Purpose I/O

2.2 Clock Operation

The TCK/CK output pin will do an XOR of the current state of the CLK pin twice. This means that if the clock pin is set low, then the clock will go high then low to be 1 clock cycle. If the clock pin were set high, then the clock will go low then high to be 1 clock cycle.

The implications of this are:

If the clock starts at an idle state of logic 0:

Data can be clocked out on a –ve clock edge.
Data can be clocked in on a +ve clock edge.

If the clock starts at an idle state of logic 1:

Data can be clocked out on a +ve clock edge.
Data can be clocked in on a -ve clock edge.

3 Command Definitions

The following section will define the opcodes required to perform specific functions.

3.1 BadCommands

If the device detects a bad command it will send back 2 bytes to the PC.

0xFA,
followed by the byte which caused the bad command.

If the commands and responses that are read/written have got out of sequence then this will tell you what the first pattern was that it detected an error. The error may have occurred before this, (for example sending the wrong amount of data after a write command) and will only trigger when bit 7 of the rogue command is high.

3.2 Data Shifting Command Overview

The data shifting commands are made up of the following definitions:

Bit 0 : -ve CLK on write
Bit 1 : bit mode = 1 else byte mode
Bit 2 : -ve CLK on read
Bit 3 : LSB first = 1 else MSB first
Bit 4 : Do write TDI
Bit 5 : Do read TDO
Bit 6 : Do writeTMS
Bit 7 : 0

The write commands to TDI take effect when bits 7 and 6 are '0'. Read TDO will operate with TMS output or TDI output or on its own.

3.3 MSB FIRST

The following commands are used when data is transferred with the Most Significant Bit (MSB) first.

OPCODE	Data IN	Data OUT	BITS / BYTES	IN CLK EDGE	OUT CLK EDGE
0x10	-	YES	BYTES	-	+VE
0x11	-	YES	BYTES	-	-VE
0x12	-	YES	BITS	-	+VE
0x13	-	YES	BITS	-	-VE
0x20	YES	-	BYTES	+VE	-
0x24	YES	-	BYTES	-VE	-
0x22	YES	-	BITS	+VE	-
0x26	YES	-	BITS	-VE	-
0x30	YES	YES	BYTES	+VE	+VE
0x31	YES	YES	BYTES	+VE	-VE
0x34	YES	YES	BYTES	-VE	+VE
0x35	YES	YES	BYTES	-VE	-VE
0x32	YES	YES	BITS	+VE	+VE
0x33	YES	YES	BITS	+VE	-VE
0x36	YES	YES	BITS	-VE	+VE
0x37	YES	YES	BITS	-VE	-VE

3.3.1 Clock Data Bytes Out on +ve clock edge MSB first (no read) (use if CLK starts at '1')

0x10,
LengthL,
LengthH,
Byte1
..
Byte65536 (max)

This will clock out bytes on TDI/DO from 1 to 65536 depending on the Length bytes. A length of 0x0000 will do 1 byte and a length of 0xffff will do 65536 bytes. The data is sent MSB first. Bit 7 of the first byte is placed on TDI/D0 then the CLK pin is clocked. The data will change to the next bit on the rising edge of the CLK pin. No data is clocked into the device on TDO/DI.

3.3.2 Clock Data Bytes Out on -ve clock edge MSB first (no read) (use if CLK starts at '0')

0x11,
LengthL,
LengthH,

Byte1
..
Byte65536 (max)

This will clock out bytes on TDI/DO from 1 to 65536 depending on the Length bytes. A length of 0x0000 will do 1 byte and a length of 0xffff will do 65536 bytes. The data is sent MSB first. Bit 7 of the first byte is placed on TDI/DO then the CLK pin is clocked. The data will change to the next bit on the falling edge of the CLK pin. No data is clocked into the device TDO/DI.

3.3.3 Clock Data Bits Out on +ve clock edge MSB first (no read) (use if CLK starts at '1')

0x12,
Length,
Byte1

This will clock out bits on TDI/DO from 1 to 8 depending on the Length byte. A length of 0x00 will do 1 bit and a length of 0x07 will do 8 bits. The data is sent MSB first. Bit 7 of the data byte is placed on TDI/DO then the CLK pin is clocked. The data will change to the next bit on the rising edge of the CLK pin. No data is clocked into the device on TDO/DI.

3.3.4 Clock Data Bits Out on -ve clock edge MSB first (no read) (use if CLK starts at '0')

0x13,
Length,
Byte1

This will clock out bits on TDI/DO from 1 to 8 depending on the Length byte. A length of 0x00 will do 1 bit and a length of 0x07 will do 8 bits. The data is sent MSB first. Bit 7 of the data byte is placed on TDI/DO then the CLK pin is clocked. The data will change to the next bit on the falling edge of the CLK pin. No data is clocked into the device on TDO/DI.

3.3.5 Clock Data Bytes In on +ve clock edge MSB first (no write)

0x20,
LengthL,
LengthH

This will clock in bytes on TDO/DI from 1 to 65536 depending on the Length bytes. A length of 0x0000 will do 1 byte and a length of 0xffff will do 65536 bytes. The first bit in will be the MSB of the first byte and so on. The data will be sampled on the rising edge of the CLK pin. No data is clocked out of the device on TDI/DO.

3.3.6 Clock Data Bytes In on -ve clock edge MSB first (no write)

0x24,
LengthL,
LengthH

This will clock in bytes on TDO/DI from 1 to 65536 depending on the Length bytes. A length of 0x0000 will do 1 byte and a length of 0xffff will do 65536 bytes. The first bit in will be the MSB of the first byte and so on. The data will be sampled on the falling edge of the CLK pin. No data is clocked out of the device on TDI/DO.

3.3.7 Clock Data Bits In on +ve clock edge MSB first (no write) (TDO/DI sampled just prior to rising edge)

0x22,
Length,

This will clock in bits on TDO/DI from 1 to 8 depending on the Length byte. A length of 0x00 will do 1 bit and a length of 0x07 will do 8 bits. The data will be shifted up so that the first bit in may not be in bit 7 but from 6 downwards depending on the number of bits to shift (i.e. a length of 1 bit will have the data bit sampled in bit 0 of the byte sent back to the PC). The data will be sampled on the rising edge of the CLK pin. No data is clocked out of the device on TDI/DO.

3.3.8 Clock Data Bits In on -ve clock edge MSB first (no write) (TDO/DI sampled just prior to falling edge)

0x26,
Length,

This will clock in bits on TDO/DI from 1 to 8 depending on the Length byte. A length of 0x00 will do 1 bit and a length of 0x07 will do 8 bits. The data will be shifted up so that the first bit in may not be in bit 7 but from 6 downwards depending on the number of bits to shift (i.e. a length of 1 bit will have the data bit sampled in bit 0 of the byte sent back to the PC). The data will be sampled on the falling edge of the CLK pin. No data is clocked out of the device on TDI/DO.

3.3.9 Clock Data Bytes In and Out MSB first

The following commands allow for data to be clocked in and out at the same time most significant bit first.

0x30, out on +ve edge, in on +ve edge
LengthL,
LengthH,
Byte1
..
Byte65536 (max)

or

0x31, out on -ve edge, in on +ve edge
LengthL,
LengthH,
Byte1
..
Byte65536 (max)

or

0x34, out on +ve edge, in on -ve edge
LengthL,
LengthH,
Byte1
..
Byte65536 (max)

or

0x35, out on -ve edge, in on -ve edge
LengthL,
LengthH,
Byte1
..
Byte65536 (max)

3.3.10 Clock Data Bits In and Out MSB first

The following commands allow for data to be clocked in and out at the same time most significant bit first.

0x32, out on +ve edge, in on +ve edge
Length
Byte

or

0x33, out on -ve edge, in on +ve edge
Length
Byte

or

0x36, out on +ve edge, in on -ve edge
Length
Byte

or

0x37, out on -ve edge, in on -ve edge
Length
Byte

3.4 LSB FIRST

The following commands are used when data is transferred with the Least Significant Bit (LSB) first.

OPCODE	Data IN	Data OUT	BITS / BYTES	IN CLK EDGE	OUT CLK EDGE
0x18	-	YES	BYTES	-	+VE
0x19	-	YES	BYTES	-	-VE
0x1A	-	YES	BITS	-	+VE
0x1B	-	YES	BITS	-	-VE
0x28	YES	-	BYTES	+VE	-
0x2C	YES	-	BYTES	-VE	-
0x2A	YES	-	BITS	+VE	-
0x2E	YES	-	BITS	-VE	-
0x38	YES	YES	BYTES	+VE	+VE
0x39	YES	YES	BYTES	+VE	-VE
0x3C	YES	YES	BYTES	-VE	+VE
0x3D	YES	YES	BYTES	-VE	-VE
0x3A	YES	YES	BITS	+VE	+VE
0x3B	YES	YES	BITS	+VE	-VE
0x3E	YES	YES	BITS	-VE	+VE
0x3F	YES	YES	BITS	-VE	-VE

3.4.1 Clock Data Bytes Out on +ve clock edge LSB first (no read) (use if CLK starts at '1')

0x18,
LengthL,
LengthH,
Byte1
..
Byte65536 (max)

This will clock out bytes on TDI/DO from 1 to 65536 depending on the Length bytes. A length of 0x0000 will do 1 byte and a length of 0xffff will do 65536 bytes. The data is sent LSB first. Bit 0 of the first byte is placed on TDI/DO then the CLK pin is clocked. The data will change to the next bit on the rising edge of the CLK pin. No data is clocked into the device on TDO/DI.

3.4.2 Clock Data Bytes Out on -ve clock edge LSB first (no read) (use if CLK starts at '0')

0x19,
LengthL,
LengthH,
Byte1
..
Byte65536 (max)

This will clock out bytes on TDI/DO from 1 to 65536 depending on the Length bytes. A length of 0x0000 will do 1 byte and a length of 0xffff will do 65536 bytes. The data is sent LSB first. Bit 0 of the first byte is placed on TDI/DO then the CLK pin is clocked. The data will change to the next bit on the falling edge of the CLK pin. No data is clocked into the device on TDO/DI.

3.4.3 Clock Data Bits Out on +ve clock edge LSB first (no read) (use if CLK starts at '1')

0x1A,
Length,
Byte1

This will clock out bits on TDI/DO from 1 to 8 depending on the Length byte. A length of 0x00 will do 1 bit and a length of 0x07 will do 8 bits. The data is sent LSB first. Bit 0 of the data byte is placed on TDI/DO then the CLK pin is clocked. The data will change to the next bit on the rising edge of the CLK pin. No data is clocked into the device on TDO/DI.

3.4.4 Clock Data Bits Out on -ve clock edge LSB first (no read) (use if CLK starts at '0')

0x1B,
Length,
Byte1

This will clock out bits on TDI/DO from 1 to 8 depending on the Length byte. A length of 0x00 will do 1 bit and a length of 0x07 will do 8 bits. The data is sent LSB first. Bit 0 of the data byte is placed on TDI/DO then the CLK pin is clocked. The data will change to the next bit on the falling edge of the CLK pin. No data is clocked into the device on TDO/DI.

3.4.5 Clock Data Bytes In on +ve clock edge LSB first (no write)

0x28,
LengthL,
LengthH

This will clock in bytes on TDO/DI from 1 to 65536 depending on the Length bytes. A length of 0x0000 will do 1 byte and a length of 0xffff will do 65536 bytes. The first bit in will be the LSB of the first byte and so on. The data will be sampled on the rising edge of the CLK pin. No data is clocked out of the device on TDI/DO.

3.4.6 Clock Data Bytes In on -ve clock edge LSB first (no write)

0x2C,
LengthL,
LengthH

This will clock in bytes on TDO/DI from 1 to 65536 depending on the Length bytes. A length of 0x0000 will do 1 byte and a length of 0xffff will do 65536 bytes. The first bit in will be the LSB of the first byte and so on. The data will be sampled on the falling edge of the CLK pin. No data is clocked out of the device on TDI/DO.

3.4.7 Clock Data Bits In on +ve clock edge LSB first (no write) (TDO/DI sampled just prior to rising edge)

0x2A,
Length,

This will clock in bits on TDO/DI from 1 to 8 depending on the Length byte. A length of 0x00 will do 1 bit and a length of 0x07 will do 8 bits. The data will be shifted down so that the first bit in may not be in bit 0 but from 1 upwards depending on the number of bits to shift (i.e. a length of 1 bit will have the data bit sampled in bit 7 of the byte sent back to the PC). The data will be sampled on the rising edge of the CLK pin. No data is clocked out of the device on TDI/DO.

3.4.8 Clock Data Bits In on -ve clock edge LSB first (no write) (TDO/DI sampled just prior to falling edge)

0x2E,
Length,

This will clock in bits on TDO/DI from 1 to 8 depending on the Length byte. A length of 0x00 will do 1 bit and a length of 0x07 will do 8 bits. The data will be shifted down so that the first bit in may not be in bit 0 but from 1 upwards depending on the number of bits to shift (i.e. a length of 1 bit will have the data bit sampled in bit 7 of the byte sent back to the PC). The data will be sampled on the falling edge of the CLK pin. No data is clocked out of the device on TDI/DO.

3.4.9 Clock Data Bytes In and Out LSB first

The following commands allow for data to be clocked in and out at the same time least significant bit first.

0x38, out on +ve edge, in on +ve edge
LengthL,
LengthH,
Byte1
..
Byte65536 (max)

or

0x39, out on -ve edge, in on +ve edge
LengthL,
LengthH,
Byte1
..
Byte65536 (max)

or

0x3C, out on +ve edge, in on -ve edge
LengthL,
LengthH,
Byte1
..
Byte65536 (max)

or

0x3D, out on -ve edge, in on -ve edge
LengthL,
LengthH,
Byte1
..
Byte65536 (max)

3.4.10 Clock Data Bits In and Out LSB first

The following commands allow for data to be clocked in and out at the same time least significant bit first.

0x3A, Length Byte	out on +ve edge, in on +ve edge
or	
0x3B, Length Byte	out on -ve edge, in on +ve edge
or	
0x3E, Length Byte	out on +ve edge, in on -ve edge
or	
0x3F, Length Byte	out on -ve edge, in on -ve edge

3.5 TMS Commands

The following commands are primarily intended for use in JTAG interfaces where the TMS signal has to be controlled to navigate the JTAG state machine.

OPCODE	Data IN	Data OUT	IN CLK EDGE	OUT CLK EDGE
0x4A	-	YES	-	+VE
0x4B	-	YES	-	-VE
0x6A	YES	YES	+VE	+VE
0x6B	YES	YES	-VE	+VE
0x6E	YES	YES	+VE	-VE
0x6F	YES	YES	-VE	-VE

3.5.1 Clock Data to TMS pin (no read)

0x4A or 0x4B
Length,
Byte1

This will send data bits 6 down to 0 to the TMS pin using the LSB or MSB and -ve or +ve clk , depending on which of the lower bits have been set.

0x4A : TMS with LSB first on +ve clk edge - use if clk is set to '1'
0x4B : TMS with LSB first on -ve clk edge - use if clk is set to '0'

Bit 7 of the Byte1 is passed on to TDI/DO before the first clk of TMS and is held static for the duration of TMS clocking. No read operation will take place.

3.5.2 Clock Data to TMS pin with read

0x6A or 0x6B or 0x6E or 0x6F
Length,
Byte1

This will send data bits 6 down to 0 to the TMS pin using the LSB or MSB and -ve or +ve clk , depending on which of the lower bits have been set.

0x6A : TMS with LSB first on +ve clk edge, read on +ve edge - use if clk is set to '1'
0x6B : TMS with LSB first on -ve clk edge, read on +ve edge - use if clk is set to '0'
0x6E : TMS with LSB first on +ve clk edge, read on -ve edge - use if clk is set to '1'
0x6F : TMS with LSB first on -ve clk edge, read on -ve edge - use if clk is set to '0'

Bit 7 of the Byte1 is passed on to TDI/DO before the first clk of TMS and is held static for the duration of TMS clocking. The TDO/DI pin is sampled for the duration of TMS and a byte containing the data is passed back at the end of TMS clocking.

3.6 Set / Read Data Bits High / Low Bytes

The following commands are used to set the initial direction and logic state of the pins when first entering MPSSE mode. They are also use to set or read GPIO pins. The low byte would be ADBUS 7-0, and the high byte is ACBUS 7-0.

OPCODE	Set O/P	Read I/P	High Byte	Low Byte
0x80	YES	-	-	YES
0x82	YES	-	YES	-
0x81	-	YES	-	YES
0x83	-	YES	YES	-

3.6.1 Set Data bits LowByte

0x80,

0xValue,
0xDirection

This will setup the direction of the first 8 lines and force a value on the bits that are set as output. A 1 in the Direction byte will make that bit an output.

3.6.2 Set Data bits High Byte

0x82,
0xValue,
0xDirection

This will setup the direction of the high 8 lines and force a value on the bits that are set as output. A 1 in the Direction byte will make that bit an output.

3.6.3 Read Data bits LowByte

0x81,

This will read the current state of the first 8 pins and send back 1 byte.

3.6.4 Read Data bits HighByte

0x83,

This will read the current state of the high 8 pins and send back 1 byte.

3.7 Loopback Commands

If loopback is enabled the TDI/DO and TDO/DI pins are internally connected to allow for testing data transfer without an external device.

OPCODE	Loopback Enabled
0x84	YES
0x85	NO

3.7.1 Connect TDI to TDO for Loopback

0x84,

This will connect the TDI/DO output to the TDO/DI input for loopback testing.

3.7.2 Disconnect TDI to TDO for Loopback

0x85,

This will disconnect the TDI output from the TDO input for loopback testing.

3.8 Clock Divisor

The following section defines how to set the speed at which data is clocked in or out of the device.

3.8.1 Set TCK/SK Divisor (FT2232D)

0x86,
0xValueL,
0xValueH

This will set the clock divisor. The TCK/SK always has a duty cycle of 50%, except between commands where it will remain in its initial state. The initial state is set using the Set Data Bits Low Byte command (0x80). For example, to use it in JTAG mode you would issue:-

0x80 Set Data Bits Low Byte
0x08 TCK/SK, TDI/D0 low, TMS/CS high
0x0B TCK/SK, TDI/D0, TMS/CS output, TDO/DI and GPIOL0 -> GPIOL3 input

The clock will then start low. When the MPSSE is sent a command to clock bits (or bytes) it will make the clock go high and then back low again as 1 clock period. For TMS/CS commands, a 0x4B command would be used for no read, and a 0x6B command for TMS/CS with read. For clocking data out on TDI/DO with no read of TDO/DI, a 0x19 command would be used for bytes and 0x1B for bits. To read from TDO/DI with no data sent on TDI/DO a 0x28 command would be used for bytes and 0x2A for bits. To scan in and out at the same time a 0x39 command would be used for bytes and 0x3B for bits.

The TCK/SK frequency can be worked out using the following algorithm:

TCK/SK period = 12MHz / ((1 +[(0xValueH * 256) OR 0xValueL]) * 2)

For example:

Value	TCK/SK Max
0x0000	6 MHz
0x0001	3 MHz
0x0002	2 MHz
0x0003	1.5 MHz
0x0004	1.2 MHz
.....	.....
0xFFFF	91.553 Hz

3.8.2 Set clk divisor (FT2232H/FT4232H)

The TCK/CK clock output pin has a front stage divide by 5 from the 60 MHz internal clock for backward compatibility with the FT2232D device. See command 0x8A for disabling the divide by 5.

0x86,
0xValueL,
0xValueH,

This will set the clock divisor.

The TCK is always 50% duty cycle (except between commands where it will remain in its initial state). The initial state is set using the **Set Data bits LowByte** command. For example for using it in JTAG mode you would issue:

0x80 Set Data Bits Low Byte
0x08 TCK TDI low, TMS high
0x0B TCK, TDI, TMS output, TDO and GPIOL0-> GPIOL3 input

The clock will start low. When the MPSSE is sent a command to clock bits or bytes it will make the clock go high and then back low again as 1 clock period. For TMS commands you would use command 0x4B for no read and 0x6B for TMS with read. For clocking data out on TDI with no read of TDO, you would use command 0x19 for bytes and 0x1B for bits. To read from TDO with no data sent on TDI you would use command 0x28 for bytes and 0x2A for bits. To scan in and out at the same time you would use command 0x39 for bytes and 0x3B for bits.

For example with the divide by 5 set as on:

The TCK frequency can be worked out by the following algorithm :

TCK period = 12MHz / ((1 +[(0xValueH * 256) OR 0xValueL]) * 2)

value TCK max

0x0000 6 MHz
0x0001 3 MHz
0x0002 2 MHz
0x0003 1.5 MHz
0x0004 1.2 MHz
.....
0xFFFF 91.553 Hz

For example with the divide by 5 set as off:

The TCK frequency can be worked out by the following algorithm :

TCK period = 60MHz / ((1 +[(0xValueH * 256) OR 0xValueL]) * 2)

value TCK max

0x0000 30 MHz
0x0001 15 MHz
0x0002 10 MHz
0x0003 7.5 MHz
0x0004 6 MHz
.....
0xFFFF 457.763 Hz

4 Instructions for CPU mode

4.1 Overview

In this mode the chip emulates a CPU style of interface with:

a) a multiplexed 8 bit address and data bus
b) an extended 8 bit address bus
c) CS#, ALE, WR#, RD# and OSC signals
d) 2 I/O lines that can be used as extra I/O or to wait for IRQs

OPCODE	Write	Read	Short Address	Extended Address
0x90	-	YES	YES	-
0x91	-	YES	-	YES
0x92	YES	-	YES	-
0x93	YES	-	-	YES

4.2 CPUMode Read Short Address

0x90,
0xAddrLow

This will read 1 byte from the target device.

4.3 CPUMode Read Extended Address

0x91,
0xAddrHigh
0xAddrLow

This will read 1 byte from the target device.

4.4 CPUMode Write Short Address

0x92,
0xAddrLow,
0xData

This will write 1 byte from the target device.

4.5 CPUMode Write Extended Address

0x93,
0xAddrHigh,
0xAddrLow,
0xData

This will write 1 byte from the target device.

5 Instructions for use in both MPSSE and MCU Host Emulation Modes

5.1 Send Immediate

0x87,

This will make the chip flush its buffer back to the PC.

5.2 Wait On I/O High

0x88,

This will cause the MPSSE controller to wait until GPIOL1 (JTAG) or I/O1 (CPU) is high. Once it is detected as high, it will move on to process the next instruction. The only way out of this will be to disable the controller if the I/O line never goes high.

5.3 Wait On I/O Low

0x89,

This will cause the controller to wait until GPIOL1 (JTAG) or I/O1 (CPU) is low. Once it is detected as low, it will move on to process the next instruction. The only way out of this will be to disable the controller if the I/O line never goes low.

6 FT2232H / FT4232H ONLY

The commands in this section apply only to the FT2232H and FT4232H devices.

OPCODE	Function
0x8A	Disables the clk divide by 5 to allow for a 60MHz master clock.
0x8B	Enables the clk divide by 5 to allow for backward compatibility with FT2232D
0x8C	Enables 3 phase data clocking. Used by I2C interfaces to allow data on both clock edges.
0x8D	Disables 3 phase data clocking.
0x8E	Allows for a clock to be output without transferring data. Commonly used in the JTAG state machine. Clocks counted in terms of numbers of bits
0x8F	Allows for a clock to be output without transferring data. Commonly used in the JTAG state machine. Clocks counted in terms of numbers of bytes
0x94	Allows for a clock to be output without transferring data until a logic 1 input on GPIOL1 stops the clock.
0x95	Allows for a clock to be output without transferring data until a logic 0 input on GPIOL1 stops the clock.
0x96	Enable adaptive clocking
0x97	Disable adaptive clocking
0x9C	Allows for a clock to be output without transferring data until a logic 1 input on GPIOL1 stops the clock or a set number of clock pulses are sent. Clocks counted in terms of numbers of bytes
0x9D	Allows for a clock to be output without transferring data until a logic 0 input on GPIOL1 stops the clock or a set number of clock pulses are sent. Clocks counted in terms of numbers of bytes

6.1 Disable Clk Divide by 5

0x8A

This will turn off the divide by 5 from the 60 MHz clock.

6.2 Enable Clk Divide by 5

0x8B

This will turn on the divide by 5 from the 60 MHz clock to give a 12MHz master clock for backward compatibility with FT2232D designs.

6.3 Enable 3 Phase Data Clocking

0x8C

This will give a 3 stage data shift for the purposes of supporting interfaces such as I2C which need the data to be valid on both edges of the clk. So it will appear as

Data setup for ½ clock period -> pulse clock for ½ clock period -> Data hold for ½ clock period.

6.4 Disable 3 Phase Data Clocking

0x8D

This will give a 2 stage data shift which is the default state. So it will appear as

Data setup for ½ clock period -> Pulse clock for ½ clock period

6.5 Clock For n bits with no data transfer

0x8E
Length,

This will pulse the clock for 1 to 8 times given by length. A length of 0x00 will do 1 clock and a length of 0x07 will do 8 clocks.

6.6 Clock For n x 8 bits with no data transfer

0x8F
LengthL,
LengthH,

This will pulse the clock for 8 to (8 x $10000) times given by length. A length of 0x0000 will do 8 clocks and a length of 0xFFFF will do 524288 clocks

6.7 Clk continuously and Wait On I/O High

0x94,

This will cause the controller to create CLK pulses until GPIOL1 or I/O1 (CPU mode of FT2232H) is low. Once it is detected as high, it will move on to process the next instruction. The only way out of this will be to disable the controller if the I/O line never goes low.

6.8 Clk continuously and Wait On I/O Low

0x95,

This will cause the controller to create CLK pulses until GPIOL1 or I/O1 (CPU mode of FT2232H) is high. Once it is detected as low, it will move on to process the next instruction. The only way out of this will be to disable the controller if the I/O line never goes high.

6.9 Turn On Adaptive clocking

0x96,

Adaptive clocking is required when using the JTAG interface on an ARM processor.

This will cause the controller to wait for RTCK from the ARM processor which should be fed back into GPIOL3 (it is an input). After the TCK output has changed the controller waits until RTCK is sampled to be the same before it changes TCK again. It could be considered as an acknowledgement that the CLK signal was received.

6.10 Turn Off Adaptive clocking

0x97,

This will turn off adaptive clocking.

6.11 Clock For n x 8 bits with no data transfer or Until GPIOL1 is High

0x9C
LengthL,
LengthH,

This will pulse the clock for 8 to (8 x $10000) times given by length. A length of 0x0000 will do 8 clocks and a length of 0xFFFF will do 524288 clocks or until GPIOL1 is high.

6.12 Clock For n x 8 bits with no data transfer or Until GPIOL1 is Low

0x9D
LengthL,
LengthH,

This will pulse the clock for 8 to (8 x $10000) times given by length. A length of 0x0000 will do 8 clocks and a length of 0xFFFF will do 524288 clocks or until GPIOL1 is low.

7 Contact Information

Head Office – Glasgow, UK

Future Technology Devices International Limited
Unit 1, 2 Seaward Place, Centurion Business Park
Glasgow G41 1HH
United Kingdom
Tel: +44 (0) 141 429 2777
Fax: +44 (0) 141 429 2758

E-mail (Sales) sales1@ftdichip.com
E-mail (Support) support1@ftdichip.com
E-mail (General Enquiries) admin1@ftdichip.com
Web Site URL http://www.ftdichip.com
Web Shop URL http://www.ftdichip.com

Branch Office – Taipei, Taiwan

Future Technology Devices International Limited
(Taiwan)
2F, No. 516, Sec. 1, NeiHu Road
Taipei 114
Taiwan , R.O.C.
Tel: +886 (0) 2 8791 3570
Fax: +886 (0) 2 8791 3576

E-mail (Sales) tw.sales1@ftdichip.com
E-mail (Support) tw.support1@ftdichip.com
E-mail (General Enquiries) tw.admin1@ftdichip.com
Web Site URL http://www.ftdichip.com

Branch Office – Hillsboro, Oregon, USA

Future Technology Devices International Limited
(USA)
7235 NW Evergreen Parkway, Suite 600
Hillsboro, OR 97123-5803
USA
Tel: +1 (503) 547 0988
Fax: +1 (503) 547 0987

E-Mail (Sales) us.sales@ftdichip.com
E-Mail (Support) us.support@ftdichip.com
E-Mail (General Enquiries) us.admin@ftdichip.com
Web Site URL http://www.ftdichip.com

Branch Office – Shanghai, China

Future Technology Devices International Limited
(China)
Room 408, 317 Xianxia Road,
Shanghai, 200051
China
Tel: +86 21 62351596
Fax: +86 21 62351595

E-mail (Sales) cn.sales@ftdichip.com
E-mail (Support) cn.support@ftdichip.com
E-mail (General Enquiries) cn.admin@ftdichip.com
Web Site URL http://www.ftdichip.com

Distributor and Sales Representatives

Please visit the Sales Network page of the FTDI Web site for the contact details of our distributor(s) and sales representative(s) in your country.

Appendix A – Revision History

Version 1.00	First Release	21/01/2009
Version 1.1	Changed opcodes 0x98 and 0x99 to 0x9C and 0x9D	03/02/2009
Version 1.2	Corrected opcodes 0x94,0x95,0x96,0x97,0x9C and 0x9D To use the correct GPIOLx. Corrected Taiwan address information	09/06/2009
Version 1.21	Edited section 3.7.1	05/03/2010