7I76E/7I76ED
ETHERNET STEP/DIR PLUS I/O DAUGHTERCARD

V1.11

Table of Contents

Table of Contents

Table of Contents

Table of Contents

GENERAL

DESCRIPTION

The 7I76E/7I76ED are remote FPGA cards with Ethernet interface designed for interfacing up to 5 Axis of step&dir step motor or servo motor drives. The 7I76E/7I76ED also provide a spindle encoder interface, isolated analog spindle speed control and 48 isolated I/O points for general purpose field I/O use.

All step and direction outputs are buffered 5V signals that can drive 24 mA. All outputs support differential mode to reduce susceptibility to noise. An isolated analog spindle voltage with direction and enable outputs is provided for spindle control as is a single spindle encoder channel with TTL or differential inputs.

48 points of isolated field I/O are provided for general control use including limit switch and control panel inputs, coolant enable and tool changer control outputs. Isolated I/O includes 32 sinking inputs and 16 sourcing (7I76E) or sinking (7I76ED) outputs. Inputs can sense 5V to 32V signals and the outputs can switch 5V through 32V signals. Maximum output load is 350 mA. Outputs are short circuit protected.

In addition to the being able to read digital on/off status of each input, four input pin voltages are readable with 8 bit resolution, and two MPG encoder inputs are provided as an option on four field inputs. Field I/O is powered by an isolated 10-32V field power source.

One RS-422 interface is provided for I/O expansion via a serial I/O daughtercard. In addition to the on card I/O, two FPGA expansion connectors compatible with Mesa's 25 pin daughtercards allow almost unlimited I/O options including additional quadrature or absolute encoder inputs, step/dir or PWM/dir outputs, and field I/O expansion to hundreds of I/O of points. All field wiring is terminated in pluggable 3.5 mm screw terminal blocks.

Unless the 7I76ED is mentioned specifically, all information in this manual applies to both the 7I76E and 7I76ED

HARDWARE CONFIGURATION

GENERAL

Hardware setup jumper positions assume that the 7I76E card is oriented in an upright position, that is, with the host interface RJ-45 connector pointing towards the left.

VIN POWER SOURCE

The isolated field I/O on the 7I76E runs from a switching power supply that can be powered by field power or a separate supply (VIN) with ground common with field power. Normally the 7I76E will be powered with field power and an on card jumper, W1 allows VIN to be connected to field power. If you wish to use a single power supply for the 7I76Es field outputs and field logic power, W1 should be placed in the left hand position. This connects field power to VIN. If you wish to use a separate supply for VIN, W1 Should be placed in the right hand position.

LOGIC SECTION POWER

The 7I76E can get its 5V Ethernet, encoder, step/dir and serial interface power from connector P3 or the on card logic power regulator can be used with unregulated DC power supplied to TB3. When the on card regulator is used, the maximum power available from the encoder, serial, and 26 pin expansion connectors is 2A total. If isolation of field power and logic power is not required, a common 8 to 32VDC power supply can be used for all 7I76E power.

SETUP/OPERATE MODE

The 7I76E isolated I/O section can run in setup mode or operate mode. In setup mode, the serial interface baud rate is fixed at 115.2K baud. In the operate mode, the baud rate is set to 2.5M baud (default). Setup mode enables a normal PC to communicate with the 7I76E for setup purposes. W8 controls the setup/operate mode selection. W8 **must** be in the "operate" mode for normal operation.

W8	MODE	BAUD RATE
LEFT	Operate mode	2.5M baud (default, can be changed)
RIGHT	Setup Mode	115.2K baud (fixed)

ENCODER INPUT MODE

The 7I76Es high speed encoder input can be programmed for differential or single ended mode operation. W10, W11 and W13 set the encoder input mode. When W10,W11,and W13 are in the right hand position, the encoder input is mode is differential. When W10,W11, and W13 are in the left hand position, the encoder input mode is single ended or "TTL". Note that W10 controls the input mode for the ‘A’ signal, W11 controls the input mode for the ‘B’ signal and W13 controls the input mode for the index signal.

HARDWARE CONFIGURATION

CONNECTOR 5V POWER

The 7I76E has the option to supply 5V power from the to the breakout board connected to its expansion connectors.

The 5V power option is individually selectable for each of the two I/O connectors. The breakout 5V power is protected by per connector PTC devices so will not cause damage to the 7I76E or system if accidentally shorted. This option should only be enabled for Mesa breakout boards or boards specifically wired to accept 5V power on DB25 pins 22 through 25. When the option is disabled DB25 pins 22 through 25 are grounded. Jumper W7 controls the power option on P1, W12 controls the power option on P2.

JUMPER	POS	FUNCTION
W7,W12	UP	BREAKOUT POWER ENABLED
W7,W12	DOWN	BREAKOUT POWER DISABLED (*DEFAULT*)

5V I/O TOLERANCE

The FPGA used on the 7I76E has a 4V absolute maximum input voltage specification. To allow interfacing with 5V inputs on its expansion connectors, the 7I76E has bus switches on all expansion I/O pins. The bus switches work by turning off when the input voltage exceeds a preset threshold. *The 5V I/O tolerance option is the default and should normally be left enabled.*

For high speed applications where only 3.3V maximum signals are present and overshoot clamping is desired, the 5V I/O tolerance option can be disabled. W5 controls the 5V I/O tolerance option. When W5 is on the default UP position, 5V tolerance mode is enabled. When W5 is in the DOWN position, 5V tolerance mode is disabled. Note that W5 controls 5V tolerance on both I/O connectors.

PULLUP VOLTAGE

Jumper W6 selects the pull-up resistor voltage, When W6 is in the UP position the 4.7K I/O pullup resistor common is connected to 5V, When W6 is in the down position, The 4.7K I/O pullup resistor common is connected to 3.3V.

HARDWARE CONFIGURATION

PRECONFIG PULLUP ENABLE

The Xilinx FPGA on the 7I76E has the option of having weak pull-ups on all I/O pins at power-up or reset. The default is to enable the pull-ups To enable the built-in pull-ups, (the default condition) jumper W4 should be placed in the UP position. To disable the internal pull-ups, W4 should be in the DOWN position. It is suggested the W4 be left in the UP position.

IP ADDRESS SELECTION

The 7I76E has three options for selecting its IP address. These options are selected by Jumpers W2 and W3.

W2	**W3**	IP ADDRESS	
DOWN	DOWN	FIXED 192.168.1.121	(DEFAULT)
DOWN	UP	FIXED FROM EEPROM	
UP	DOWN	BOOTP	
UP	UP	INVALID	

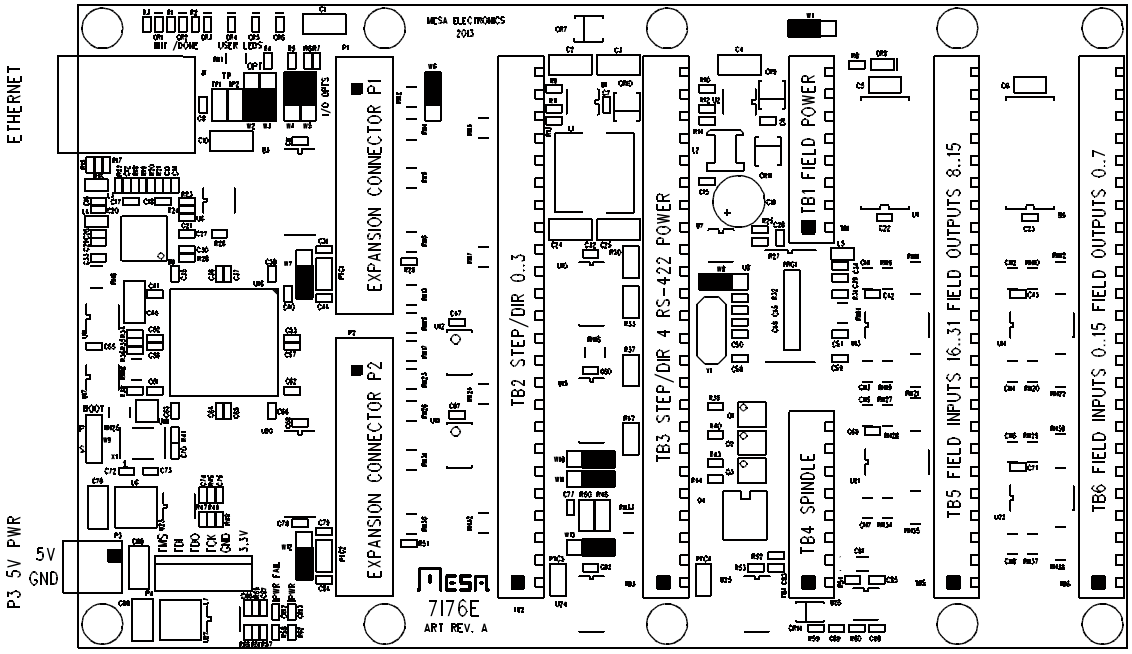
FPGA FLASH SELECT

To make recovery from FPGA configuration errors easier, there are two FPGA configuration flash memories on the 7I76E card. Jumper W9 selects between the two flash memories. That is, if one flash memory is inadvertently corrupted, the other one can be used to boot the 7I76E, allowing the corrupted flash memory to be re-written. It is suggested that W9 be left in the UP position (primary flash memory) for normal operation, and only changed to the DOWN position (secondary flash memory) if configuration fails. Once rebooted via a power cycle, jumper W9 should be promptly restored to the UP position to allow the primary flash memory to be re-written.

W9	**MEMORY**
UP	PRIMARY (NORMAL OPERATION)
DOWN	SECONDARY (BACKUP)

CONNECTORS

7I76E CONNECTOR LOCATIONS AND DEFAULT JUMPER POSITIONS



NOTE: BLACK SQUARES INDICATE PIN 1

CONNECTORS

POWER CONNECTOR PINOUT

P3 is the 7I76Es 5V power connector. 5V power for the 7I76Es logic section can be supplied here if the unregulated power input option is not used. ***Do not supply any voltage other than 5V to P3!*** P3 is a 3.5MM plug-in screw terminal block. P3 pinout is as follows:

PIN	FUNCTION	
1	+5V	TOP, SQUARE PAD
2	GND	BOTTOM, ROUND PAD

JTAG CONNECTOR PINOUT

P4 is a JTAG programming connector. This is normally used only for debugging or if both EEPROM configurations have been corrupted. In case of corrupted EEPROM contents the EEPROM can be re-programmed using Xilinx's Impact tool.

P4 JTAG CONNECTOR PINOUT

PIN	FUNCTION
1	TMS
2	TDI
3	TDO
4	TCK
5	GND
6	+3.3V

CONNECTORS

EXPANSION CONNECTORS

The 7I76E has two 26 pin headers to allow I/O expansion beyond the built in I/O on the 7I76E card. These headers have a pin-out that matches standard parallel port breakout cards and Mesa's 25 pin FPGA daughtercards, when terminated with a DB25 connector.

HDR PIN	P1 FUNC	P2 FUNC	HDR PIN	P1 FUNC	P2 FUNC
1	IO17	IO34	2	IO18	IO35
3	IO19	IO36	4	IO20	IO37
5	IO21	IO38	6	IO22	IO39
7	IO23	IO40	8	IO24	IO41
9	IO25	IO42	10	GND	GND
11	IO26	IO43	12	GND	GND
13	IO27	IO44	14	GND	GND
15	IO28	IO45	16	GND	GND
17	IO29	IO46	18	GND / 5V	GND / 5V
19	IO20	IO47	20	GND / 5V	GND / 5V
21	IO31	IO48	22	GND / 5V	GND or 5V
23	IO32	IO49	24	GND / 5V	GND or 5V
25	IO33	IO50	26	GND / 5V	GND or 5V

CONNECTORS

TB2 STEP AND DIR CONNECTOR

TB2 is the 7I76Es main step and direction output connector. Both polarities of step and direction signals are provided. Each channel on the interface uses 6 pins. TB2 is a 3.5 MM pluggable terminal block with supplied removable screw terminal plugs.

TB2 CONNECTOR PINOUT

TB2 PIN	SIGNAL	TB2 PIN	SIGNAL
1	GND	13	GND
2	STEP0-	14	STEP2-
3	STEP0+	15	STEP2+
4	DIR0-	16	DIR2-
5	DIR0+	17	DIR2+
6	+5VP	18	+5VP
7	GND	19	GND
8	STEP1-	20	STEP3-
9	STEP1+	21	STEP3+
10	DIR1-	22	DIR3-
11	DIR1+	23	DIR3+
12	+5VP	24	+5VP

Note: 5VP pins are PTC short circuit protected 5V output pins for field wiring.

CONNECTORS

TB3 STEP/DIR, ENCODER RS-422 CONNECTOR

TB3 has a mix of signals including step/dir channel 4, an encoder interface, a RS-422 interface, and the unregulated logic supply power input. TB3 is a 24 terminal 3.5 MM pluggable terminal block with supplied removable screw terminal plugs.

TB3 CONNECTOR PINOUT

TB3 PIN	SIGNAL	TB3 PIN	SIGNAL
1	GND	13	IDX+
2	STEP4-	14	IDX-
3	STEP4+	15	GND
4	DIR4-	16	RS-422 RX+
5	DIR4+	17	RS-422 RX-
6	+5VP	18	RS-422 TX+
7	ENCA+	19	RS-422 TX-
8	ENCA-	20	+5VP
9	GND	21	UNREG LOGIC PWR+ IN
10	ENCB+	22	UNREG LOGIC PWR+ IN
11	ENCB-	23	GND
12	+5VP	24	GND

Note: 5VP pins are PTC short circuit protected 5V output pins for field wiring.

CONNECTORS

TB4 SPINDLE CONNECTOR

TB4 is the spindle drive interface with isolated analog output and control signals for a spindle interface.TB4 is a 8 terminal 3.5 MM pluggable terminal block with supplied removable screw terminal plugs.

TB4 PINOUT

TB4 PIN	SIGNAL
1	SPINDLE-
2	SPINDLE OUT
3	SPINDLE+
4	NC
5	SPINDLE ENA-
6	SPINDLE ENA+
7	SPINDLE DIR-
8	SPINDLE DIR+

CONNECTORS

FIELD INPUT/OUTPUT CONNECTORS

Terminal blocks TB6 and TB5 are the 7I76Es field input and output terminals. Inputs 0 through 15 and outputs 0 through 7 are terminated at TB6. Inputs 16 through 31 and outputs 8 through 15 are terminated at TB5. TB6 and TB5 are 3.5 MM pluggable terminal block with supplied removable screw terminal plugs. Pin one is at the bottom edge of the 7I76E card.

TB6 CONNECTOR PINOUT

TB6 PIN	I/O	TB6 PIN	I/O
1	INPUT0	13	INPUT12
2	INPUT1	14	INPUT13
3	INPUT2	15	INPUT14
4	INPUT3	16	INPUT15
5	INPUT4	17	OUTPUT0
6	INPUT5	18	OUTPUT1
7	INPUT6	19	OUTPUT2
8	INPUT7	20	OUTPUT3
9	INPUT8	21	OUTPUT4
10	INPUT9	22	OUTPUT5
11	INPUT10	23	OUTPUT6
12	INPUT11	24	OUTPUT7

CONNECTORS

FIELD INPUT/OUTPUT CONNECTORS

TB5 CONNECTOR PINOUT

TB5 PIN	OUTPUT	TB5 PIN	OUTPUT
1	INPUT16	13	INPUT28
2	INPUT17	14	INPUT29
3	INPUT18	15	INPUT30
4	INPUT19	16	INPUT31
5	INPUT20	17	OUTPUT8
6	INPUT21	18	OUTPUT9
7	INPUT22	19	OUTPUT10
8	INPUT23	20	OUTPUT11
9	INPUT24	21	OUTPUT12
10	INPUT25	22	OUTPUT13
11	INPUT26	23	OUTPUT14
12	INPUT27	24	OUTPUT15

CONNECTORS

FIELD POWER CONNECTOR

TB1 is the 7I76Es field power connector. It supplies power to the isolated field I/O section of the 7I76E. TB1 pinout is as follows:

TB1 PIN	SIGNAL	FUNCTION	
1	VFIELD	FIELD POWER 8-32V	(Bottom pin)
2	VFIELD	FIELD POWER 8-32V	
3	VFIELD	FIELD POWER 8-32V	
4	VFIELD	FIELD POWER 8-32V	
5	VIN	ISOLATED I/O POWER 8-32V	
6	NC		
7	NC		
8	GROUND	VIN, VFIELD, COMMON	(Top pin)

Note: *When W1 is in the default left hand position, VIN is connected to VFIELD, so only VFIELD need be supplied to the 7I76E to power its field IO.*

MOTION INTERFACE

RS-422 INTERFACE

The 7I76E has one RS-422 interface available on TB3. This interface is intended for I/O expansion with Mesa SSERIAL devices. The easiest way to make a cable for interfacing the 7I76E to these devices is to take a standard CAT5 or CAT6 cable, cut it in half, and wire the individual wires to the 7I76E screw terminals. The following chart gives the CAT5 to 7I76E screw terminal connections (EIA/TIA 568B colors shown):

TB3 PIN	SIGNAL	DIRECTION	CAT5 PIN	CAT5 568B COLOR
15	GND	FROM 7I76E	4,5	BLUE, BLUE / WHITE
16	RX+	TO 7I76E	6	GREEN
17	RX-	TO 7I76E	3	GREEN / WHITE
18	TX+	FROM 7I76E	2	ORANGE
19	TX-	FROM 7I76E	1	ORANGE / WHITE
20	+5V	FROM 7I76E	7,8	BROWN / WHITE,BROWN

Note: The 6 pin terminal block requires the +5V (brown and brown/white) and ground (blue and blue/white) pairs to be terminated in single screw terminal positions.

MOTION INTERFACE

STEP/DIR INTERFACE

The 7I76E provides five channels of step/dir interface with buffered 5V differential signal pairs. Each differential pair consists of two complementary 5V outputs. The differential signals allows reliable signal transmission in noisy environments and can directly interface with RS-422 line receivers. Step motor drives with single ended inputs connect to just one of the STEP and DIR signal outputs, that is either the STEP+/DIR+ or STEP-/DIR- signals, with the unused signals left unconnected at the 7I76E. The input common signal on drives with single ended inputs connects to the 7I76Es GND or 5VP pins depending on the drive type.

ENCODER INTERFACE

The 7I76E provide a one channel encoder interface with index. This is intended as a spindle encoder but can be used for other purposes. The encoder input can be programmed for differential or single ended encoders. The encoder interface also provides short circuit protected 5V power to the encoder. When used with single ended encoders, the ENCA+, ENCB+ and IDX+ signals are wired to the encoder and the ENCA-,ENCB-, and IDX- terminal left unconnected.

SPINDLE INTERFACE

The 7I76E provides one analog output for spindle control. The analog output is a isolated potentiometer replacement type device. It functions like a potentiometer with SPINDLE + being one end of the potentiometer, SPINDLEOUT being the wiper and SPINDLE- being the other end. The voltage on SPINDLEOUT can be set to any voltage between SPINDLE- and SPINDLE+. Polarity and voltage range must always be observed for proper operation. The voltage supplied between SPINDLE+ and SPINDLE- must be between 5VDC an 15VDC with SPINDLE + always being more positive than SPINDLE-.

Because the analog output is isolated, bipolar output is possible, for example with SPINDLE+ connected to 5V and SPINDLE- connected to -5V, a +-5V analog output range is created. In this case the spindle output must be offset so that 50% of full scale is output when a 0V output is required. Note that if bipolar output is used, the output will be forced to SPINDLE- at startup or when SPINENA is false.

SPINDLE ISOLATED OUTPUTS

The 7I76E provides 2 isolated outputs for use for spindle direction control, and spindle enable. These outputs are OPTO coupler Darlington transistors. They are all isolated from one another so can be used for pull up or pull-down individually. They will switch a maximum of 50 mA at 0 to 100 VDC. The SPINENA output is special as it uses the same signal that enables the analog output. When the analog output is enabled, the SPINENA OPTO output is on.

MOTION INTERFACE

STATUS LEDS

The 7I76E has nine LEDS for card status monitoring. The color, function and locations are as follows:

LED	COLOR	FUNCTION	OK	LOCATION
CR1	YELLOW	FPGA /INIT	OFF	TOP LEFT
CR2	RED	FPGA /DONE	OFF	TOP LEFT
CR3	GREEN	USER LED3	ANY	TOP LEFT
CR3	GREEN	USER LED2	ANY	TOP LEFT
CR5	GREEN	USER LED1	ANY	TOP LEFT
CR6	GREEN	USER LED0	ANY	TOP LEFT
CR8	YELLOW	FIELD POWER	ON	TOP RIGHT
CR12	RED	POWER FAIL	OFF	BOTTOM LEFT
CR13	YELLOW	LOGIC POWER	ON	BOTTOM LEFT

In normal operation CR1 and CR2 will be off. If either is on after power-up there is a problem with configuring the FPGA. In normal operation CR8 and CR13 (power LEDS) will also be on.

FIELD I/O

FIELD I/O

The 7I76E has a 32 input, 16 output isolated field I/O system to support a wide range of input and output devices. The isolated I/O is intended for low voltage DC control systems (commonly 24VDC). Inputs are sinking type. That is they sense positive input voltages relative to field ground. Outputs are sourcing (7I76E) or sinking (7I76ED) type. Sourcing outputs supply field power to field ground referred loads. Sinking outputs ground field voltage referred loads when on.

VIN AND FIELD POWER SUPPLY

The 7I76E field I/O runs from field power supplies of 5 to 32 VDC. Field power supplies the power to the 7I76E outputs and determines the 7I76E input thresholds. VIN power runs the field I/O processor and normally is connected to field power. VIN must be greater than 8V for proper operation. This means VIN must come from a separate source if 5V field voltage is used. Power consumption is approximately 600 mW or 25 mA at 24V. *VIN power must be present for the 7I76E field I/O to be detected and operate.* Field voltages that are too high or too low will cause faults.

FIELD OUTPUT CHARACTERISTICS

The 7I76E (no D) field outputs are high side or sourcing type drivers, that is they source positive voltage to a ground referred load. For example with a standard 24V field power, +24V connects to the 7I76Es field power input (on TB1) and the outputs on TB5 and TB6 now source +24V power to loads. All 7I76 loads will have one side returned to ground or the negative lead of the 24V supply. The 7I76s outputs can drive loads of up to 350 mA. The 7I76ED field outputs are low side or sinking type drivers, that is they ground the load side of a field voltage referred load.

SOURCING VS SINKING OUTPUTS

The advantage of sourcing type field wiring is that it is less likely to cause inadvertent device actuation from the most likely type of field wiring problem which is a short to ground. The advantage of sinking drivers is compatibility with existing hardware on retrofits and the capability of using mixed output voltages.

SHORT CIRCUIT PROTECTION

The 7I76Es outputs have short circuit protection and will turn off if short circuit current exceeds approximately 800 mA. The 7I76E firmware will detect this condition, disable the affected output and indicate a fault.

OVERTEMPERATURE PROTECTION

The output driver chips detect over temperature conditions. If the 7I76E detects a driver chip with a over temperature warning flag asserted, it will disable the affected chip and indicate a fault.

FIELD I/O

FIELD OUTPUT CHARACTERISTICS

MAXIMUM PER CHIP CURRENT

Because of thermal limitations there is a maximum per driver chip total current of 1.4 amps continuous. Each driver chip connects to 8 sequential outputs. If this limit is exceeded, the driver chip may go into thermal shutdown.

VOLTAGE CLAMPS

The output driver chips used on the 7I76E have built in Zener diode clamps to clamp inductive turn-off (fly-back) spikes. This means that flyback diodes are not normally required on small (less than 60 mA) inductive loads. ***If high current inductive loads are switched or inductive loads are switched at high frequencies, they must have flyback diodes to limit power dissipation in the 7I76E's driver chips.***

FIELD INPUT CHARACTERISTICS

The 7I76E field inputs have a nominal input resistance of 20K Ohms to field power ground. 7I76E inputs sense positive input voltages above a preset threshold. For best general purpose use, default input threshold is 50% of the field power supply voltage with 10% hysteresis. That is with a 24V field voltage an input must be brought to 60% of 24V = 14.4V to be sensed as high and then brought to 40% of 24V = 9.6V to be sensed as low. These accurate thresholds and hysteresis allow high speed field signal detection while maintaining excellent noise immunity.

WHY SINKING INPUTS

7I76E field inputs are of the sinking type. That is, external power must be applied to the input to register as activated. This mode was chosen so that accidental grounding of an input will not register as an activated input.

It is suggested that inputs like limit switches use normally closed switches with one switch leg connected to field power and the other to the 7I76E input pin, so the normal machine state (not at limits) is to have the inputs activated. This way, a open switch wire or wire shorted to ground will cause a detectable machine fault.

ANALOG INPUTS

All field input pins are capable of reading the input voltage. These are not highly accurate or high resolution but can be useful for things like potentiometer inputs. Input resolution is 8 bits and input full scale value is 36.3V. Accuracy is +-5%. Software process data modes 1 and 2 allow reading the analog voltage on inputs 0 through 3, in addition to the 32 digital bit inputs.

FIELD VOLTAGE MONITORING

The 7I76E monitors the field voltage and can send this information to the host in some modes. If separate VIN is supplied to the 7I76E, the 7I76E can report loss of field voltage to the host.

FIELD I/O

FIELD I/O WATCHDOG AND FAULTS

The 7I76E has a watchdog timer that will set all set a fault flag if host communication does not occur at a minimum rate. Default watchdog time is 50 mS which means if not accessed at a greater than 20 Hz rate, the watchdog will bite and disable the outputs.

When a fault flag is set, outputs can not longer be set and the host must first clear the fault before normal operation can continue. This is also the 7I76Es startup condition, meaning the host must first clear the fault before starting normal operation. This is normally handled by SSLBP.

FIELD I/O PARAMETERS

The 7I76E has several user settable parameters, but normally only a very few need be changed in normal operation.

PARAMETER	TYPE	FUNCTION
NVBAUDRATE	UINT	Sets operate mode baudrate
NVUNITNUMBER	ULONG	Non-volatile unit number
UNITNUMBER	ULONG	Working unit number
NVWATCHDOGTIME	UINT	Non-volatile watchdog time in mS
WATCHDOGTIME	UINT	Working watchdog time in ms
OUTPUT	UINT	16 bits of output data
INPUT	ULONG	32 bits of input data
FAULT	UINT	7I76E fault register
STATUS	UINT	7I76E status register

FIELD I/O

NON-VOLATILE FIELD I/O PARAMETERS

All non volatile parameters start with the letters NV. Non-volatile parameters are stored permanently in the processors EEPROM and are copied to the volatile working parameters at power-up. Because of this, non-volatile parameters only take affect after a 7I76E power cycle.

OPERATE MODE BAUD RATE

The operate mode baud rate default is 2.5 MBaud. This should not be changed unless needed for non-standard applications. Baud rates are selected by writing an index value to the NVBAUDRATE parameter. The index numbers for available baud rates are as follows:

INDEX	BAUD	INDEX	BAUD	INDEX	BAUD
0	9600B	1	19200B	2	38400B
3	57600B	4	115200B	5	230400B
6	460800B	7	921600B	8	1.25MB
9	2.5MB*	10	5MB	11	10MB

WATCHDOG TIMEOUT

The default watchdog period is 50 mS but can be set to different periods to suit the application. Watchdog timeout units are mS. A watchdog timeout value of 0 will disable the watchdog. The watch dog is a safety feature and should normally not be disabled nor set to long timeout periods unless the consequences of loss of control of outputs is not important. The non-volatile watchdog timeout is set via the NVWATCHDOGTIMEOUT parameter. The working watchdog timeout is set with the WATCHDOGTIME parameter.

.

FIELD I/O

RPD, WPD, AND UFLBP

The RPD, WPD, and UFLBP are command line utilities allow reading and writing volatile and non-volatile 7I76E parameters, and updating the firmware on the 7I76E To use these utilities on most operating systems, the 7I76E must be in the setup mode or the operate mode baud rate must be 115200 KBaud or less

RPD, WPD, and UFLBP need environment variables preset before they will work. For Windows and 115200 baud, the following environment variables should be set:

SET BAUDRATE=115200

SET BAUDRATEMUL=1

SET PROTOCOL=LBP

SET INTERFACE=OSDEVICE

Example setting NVWATCHDOGTIMEOUT to 100 ms:

WPD NVWATCHDOGTIME 100

Note this is permanent change in the 7I76Es watchdog timeout and like all non-volatile parameters, will only be applied after the 7I76E has been power cycled

Example reading 7I76E faults in Hexadecimal:

RPD FAULT H

Example of temporarily disabling watchdog and the setting every other output on:

WPD WATCHDOGTIME 0

WPD OUTPUT AAAAAAAAAAAA H

Example of updating 7I76E firmware with UFLBP

UFLBP 7I76E.BIN

Note the 7I76E MUST be in setup mode for UFLBP to work properly.

FIELD I/O

SOFTWARE PROCESS DATA MODES

The 7I76E has three software selectable process data modes. These different modes select different sets of 7I76E data to be transferred between the host and the 7I76E during real time process data exchanges. For high speed applications, choosing the correct mode can reduce the data transfer sizes, resulting in higher maximum update rates.

MODE 0 I/O only mode (32 bits of input data, 16 bit of output data)

MODE 1 I/O plus analog input mode (32 bits of input data, 16 bits of output data, 4 analog input channels)

MODE 2 I/O plus analog input and field voltage and MPG mode (32 bits of input data, 16 bits of output data, 4 analog input channels, field voltage analog in, and 2 MPG encoders on inputs 16..19). Default encoder count mode is 1X to match normal 100 PPR MPGs. Encoder input threshold is fixed at 2.5V for compatibility with 5V encoder outputs.

HOST INTERFACE

FPGA

The 7I76E use a Xilinx Spartan6 FPGA in a 256 ball BGA package: XC6SLX16-FBGA256 or XC6SLX25-FBGA256 depending on 7I76E model.

IP ADDRESS SELECTION

Initial communication with the 7I76E requires knowing its IP address. The 7I76E has 3 IP address options: Default, EEPROM, and Bootp, selected by jumpers W1 and W2. Default IP address is always 192.168.1.121. The EEPROM IP address is set by writing Ethernet EEPROM locations 0x20 and 0X22. BootP allows the 7I76E address to be set by a DHCP/ BootP server. If BootP is chosen, the 7I76E will retry BootP requests at a ~1 Hz rate if the BootP server does not respond.

HOST COMMUNICATION

The 7I76E standard firmware is designed for low overhead real time communication with a host controller so implements a very simple set of IPV4 operations. These operations include ARP reply, ICMP echo reply, and UDP packet receive/send for host data communications. UDP is used so that the 7I76E can be used on a standard network with standard tools for non-real time applications. No fragmentation is allowed so maximum packet size is 1500 bytes.

UDP

All 7I76E Ethernet communication is done via UDP packets. The 7I76E socket number for UDP data communication is 27181. Read data is routed to the requesters port number. Under UDP, a simple register access protocol is used. This protocol is called LBP16.

LBP16

LBP16 allows read and write access to up to eight separate address spaces with different sizes and characteristics. Current firmware uses seven of these spaces. For efficiency, LBP16 allows access to blocks of registers at sequential increasing addresses. (Block transfers)

WINDOWS ARP ISSUES

Windows TCP stack has a characteristic that causes it to drop outgoing UDP packets when refreshing its ARP cache. Because of this you must either verify packet transmission via echoing data from the 7I76E for every transaction (reading RXUDPCount is suggested) and retrying failed transactions, or alternatively, setting up a static entry for the 7I76E in the ARP table. This is done with windows ARP command.

HOST INTERFACE

CONFIGURATION

The 7I76E is configured at power up by a SPI FLASH memory. This flash memory is an 16M bit chip that has space for two configuration files. Since all Ethernet logic on the 7I76E is in the FPGA, a problem with configuration means that Ethernet access will not be possible. For this reason there are two backup methods to recover from FPGA boot failures.

FALLBACK

The first backup system is called Fallback. The 7I76E flash memory normally contains two configuration file images, A user image and a fallback image. If the primary user configuration is corrupted, the FPGA will load the fallback configuration so the flash memory image can be repaired remotely without having to resort to switching memories or JTAG programming.

DUAL EEPROMS

The second backup method relies on the fact that there are two flash memories on the 7I76E card, selectable via jumper W9. If a configuration fails in such a way that it loads correctly (has a valid CRC) but does not work, the fallback configuration will not be invoked. To recover from this problem, the secondary flash can be selected by moving W8 to the DOWN position and using it to boot the FPGA (by cycling the power), restoring remote access and allowing the primary configuration to be repaired via Ethernet. The backup EEPROM is not write protected so if the primary EEPROM has been corrupted, you should always restore W9 to the UP position to avoid writing a bad configuration to both EEPROMS, necessitating a slow and awkward JTAG bootstrap.

HOST INTERFACE

EEPROM LAYOUT

The EEPROM used on the 7I76E for configuration storage is the M25P16. The M25P16 is a 16 M bit (2 M byte) EEPROM with 32 64K byte sectors. Configuration files are stored on sector boundaries to allow individual configuration file erasing and updating. Standard EEPROM sector layout is as follows:

0x000000	BOOT BLOCK
0x010000	FALLBACK CONFIGURATION BLOCK 0
0x020000	FALLBACK CONFIGURATION BLOCK 1
0x030000	FALLBACK CONFIGURATION BLOCK 2
0x040000	FALLBACK CONFIGURATION BLOCK 3
0x050000	FALLBACK CONFIGURATION BLOCK 4
0x060000	FALLBACK CONFIGURATION BLOCK 5
0x070000	FALLBACK CONFIGURATION BLOCK 6
0x080000	FALLBACK CONFIGURATION BLOCK 7 / UNUSED 7I76E-16
0x090000	FALLBACK CONFIGURATION BLOCK 8 / UNUSED 7I76E-16
0x0A0000	FALLBACK CONFIGURATION BLOCK 9 / UNUSED 7I76E-16
0x0B0000	FALLBACK CONFIGURATION BLOCK 10 / UNUSED 7I76E-16
0x0C0000	FALLBACK CONFIGURATION BLOCK 11 / UNUSED 7I76E-16
0x0D0000	FALLBACK CONFIGURATION BLOCK 12 / UNUSED 7I76E-16
0x0E0000	UNUSED/FREE
0x0F0000	UNUSED/FREE

HOST INTERFACE

EEPROM LAYOUT

0x100000	USER CONFIGURATION BLOCK 0
0x110000	USER CONFIGURATION BLOCK 1
0x120000	USER CONFIGURATION BLOCK 2
0x130000	USER CONFIGURATION BLOCK 3
0x140000	USER CONFIGURATION BLOCK 4
0x150000	USER CONFIGURATION BLOCK 5
0x160000	USER CONFIGURATION BLOCK 6
0x170000	USER CONFIGURATION BLOCK 7 / UNUSED 7I76E-16
0x180000	USER CONFIGURATION BLOCK 8 / UNUSED 7I76E-16
0x190000	USER CONFIGURATION BLOCK 0 9 / UNUSED 7I76E-16
0x1A0000	USER CONFIGURATION BLOCK 0 10 / UNUSED 7I76E-16
0x1B0000	USER CONFIGURATION BLOCK 0 11 / UNUSED 7I76E-16
0x1C0000	USER CONFIGURATION BLOCK 0 12 / UNUSED 7I76E-16
0x1D0000	UNUSED/FREE
0x1E0000	UNUSED/FREE
0x1F0000	UNUSED/FREE

HOST INTERFACE

BITFILE FORMAT

The configuration utilities expect standard FPGA bitfiles without any multiboot features enabled. If multiboot FPGA files are loaded they will likely cause a configuration failure. In addition for fallback to work, the -g next_config_register_write:disable, -g reset_on_error:enable and -g CRC:enable bitgen options must be set.

MESAFLASH

Linux and Windows utility programs MESAFLASH are provided to write configuration files to the 7I76E EEPROM. These files depend on a simple SPI interface built into both the standard user FPGA bitfiles and the fallback bitfile. *The MESAFLASH utilities expect standard FPGA bitfiles without any multiboot features enabled. If multiboot FPGA files are loaded they will likely cause a configuration failure.*

If mesaflash is run with a –help command line argument it will print usage information.

The following examples assume the target 7I76E is using the ROM IP address of 192.168.1.121.

`mesaflash --device 7i76e --write FPGAFILE.BIT`

Writes a standard bitfile FPGAFILE.BIT to the user area of the EEPROM.

`mesaflash --device 7i76e --verify FPGAFILE.BIT`

Verifies the user EEPROM configuration against the bit file FPGAFILE.BIT.

`mesaflash --device 7i76e --fallback --write FALLBACK.BIT`

Writes the fallback EEPROM configuration to the fallback area of the EEPROM. In addition if the bootblock is not present in block 0 of the EEPROM, it re-writes the bootblock.

SETTING EEPROM IP ADDRESS

MESAFLASH can also write the EEPROM IP address of the 7I76E:

`MESAFLASH --device 7i76e --set ip=192.168.0.100`

The above examples assume the 7I76E has its default ROM IP address (192.168.1.121). If the 7I76E is using another IP address, this must be specified on the command line with a –addr XX.XX.XX.XX command line argument.

HOST INTERFACE

FREE FLASH MEMORY SPACE

Five 64K byte blocks of flash memory space are free when both user and fallback configurations are installed on the 7I76E-25. Seventeen 64K byte blocks are free one the 7I76E-16 It is suggested that only the last three blocks, 0x1D0000 through a 0x1F0000 in the user area, be used for FPGA application flash storage.

FALLBACK INDICATION

Mesa's supplied fallback configurations blink the red INIT LED on the top right hand side of the card if the primary configuration fails and the fallback configuration loaded successfully. If this happens it means the user configuration is corrupted or not a proper configuration for the 7I76Es FPGA. This can be fixed by running the configuration utility and re-writing the user configuration**.**

FAILURE TO CONFIGURE

The 7I76E should configure its FPGA within a fraction of a second of power application. If the FPGA card fails to configure, the red /DONE LED CR2 will remain illuminated. If this happens the secondary EEPROM boot should be used. If booting from the secondary EEPROM fails, the 7I76Es EEPROMs must be re-programmed via the JTAG connector or (faster) JTAG FPGA load followed by Ethernet EEPROM update.

CLOCK SIGNALS

The 7I76E has a single 50 MHz clock signal from an on card crystal oscillator. The clock a can be multiplied and divided by the FPGAs clock generator block to generate a wide range of internal clock signals. The 50 MHz clock is also used to generate the 25MHz clock for the Ethernet interface chip.

HOST INTERFACE

LOGIC POWER

5V logic power for the host interface FPGA, expansion connectors, RS-422 and encoder connections and step/dir connections can be provided at connector P3, or alternatively an unregulated (but filtered) DC power source of +8 to +36 VDC can be provided at TB3 pins 21 and 22 (input common on TB3 pins 23 and 24). This connection uses the 7I76Es built in 5V switching regulator to supply logic power.

PULLUP RESISTORS

All expansion I/O pins are provided with pull-up resistors to allow connection to open drain, open collector, or OPTO devices. These resistors have a value of 4.7K so have a maximum pull-up current of ~1.07 mA (5V pull-up) or ~.7 mA (3.3V pull-up).

EXPANSION CONNECTOR IO LEVELS

The Xilinx FPGAs used on the 7I76E have programmable I/O levels for interfacing with different logic families. The 7I76E does not support use of the I/O standards that require input reference voltages. All standard Mesa configurations use LVTTL levels.

Note that even though the 7I76E expansion I/O can tolerate 5V signal inputs, its outputs will not swing to 5V. The outputs are push pull CMOS that will drive to the output supply rail of 3.3V. This is sufficient for TTL compatibility but may cause problems with some types of loads. For example when driving an LED that has its anode connected to 5V, in such devices as OPTO isolators and I/O module rack SSRs, the 3.3V high level may not completely turn the LED off. To avoid this problem, either drive loads that are ground referred, Use 3.3V as the VCC for VCC referred loads, or use open drain mode.

EXPANSION CONNECTOR STARTUP I/O VOLTAGE

After power-up or system reset and before the the FPGA is configured, the pull-up resistors will pull all I/O signals to a high level. If the FPGA is used for motion control or controlling devices that could present a hazard when enabled, external circuitry should be designed so that this initial state (high) results in a safe condition.

REFERENCE INFORMATION

Note that the following interface details presented here are not normally needed for users, as all register level interface details are handed by the driver code. This information is presented here for use by interface and driver developers.

SSLBP

GENERAL

SSLBP is a firmware option to HostMot2s SSERIAL serial interface that allows simple communication to LBP based peripherals like the isolated field I/O section of the 7I76E and expansion card connected to the 7I76Es RS-422 I/O port . SSERIAL is a part of the HostMot2 motion interface firmware for MESA's Anything-I/O FPGA cards.

REGISTER MAP

SSLBP has two global processor interface registers and four per channel remote device interface registers. For more details on mapping of these registers in HostMot2 memory space, see the REGMAP file that is included with the HostMot2 source distribution.

PROCESSOR INTERFACE REGISTERS

There are two processor interface registers, the COMMAND register and the DATA register. These registers allow low level communication to SSLBP's interface processor for issuing global commands, discovery, and debug operations.

REFERENCE INFORMATION

SSLBP

COMMAND REGISTER

The commands register is a 16 bit register (right justified in the 32 bit interface) with the following format:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W	M	R	D	S	T	T	T	N	N	N	N	N	N	N	N

W = BIT 15 Write bit, set high for control data write commands

M = BIT 14 ROM enable/ reset bit, set high to reset processor / download ROM

R = BIT 13 Request bit, set high for read or write command

D = BIT 12 DoIt bit, set high for DoIt commands

S = BIT 11 Start/Stop bit, actual operation depends on T:

ST = 1,0,0,0 Stop LBP interface = 0x08NN

ST = 1,0,0,1 Start LBP interface in normal mode = 0x09NN S T = 1,1,1,1 Start LBP interface in setup mode = 0x0FNN

N bits determine which channels start or do data transfer with remote device. A set bit indicates that the corresponding channel will start or do a data transfer.

A command is started when written to the command register. Command completion is signaled by the command register being cleared (to 0x0000) by the internal SSLBP firmware. If the command register is read before the command is complete, it will reflect the previously written command. *The command register should not be written when non-zero or unpredictable behavior may result.* There are two exceptions to this rule:

1. A STOP ALL command can always be written to reset the SSLBP interface.

2. Command writes with the ignore bit set can always be written (see below)

COMMAND REGISTER WRITE IGNORE

The command register has a feature that any command written with the MSB (bit 31) set will be ignored. This is for compatibility with DMA driven interfaces or any interfaces that use a fixed address list for low level hardware access so cannot skip writes.

REFERENCE INFORMATION

SSLBP

DATA REGISTER

SSLBP has a global 8 bit data register for debug and custom setup purposes. This register allows access to internal SSLBP parameters. The data register is right justified in the 32 bit Hostmot2 register.

LOCAL READ OPERATIONS

The sequence used for reading a local SSLBP variable is as follows:

1. The parameter address ORed with the Request bit (bit 13) is written to the command register.

2. The host polls the command register until it reads as zero.

3. The host reads the parameter byte from the data register

LOCAL WRITE OPERATIONS

The sequence used for writing a local SSLBP variable is as follows:

1. The host polls the command register until it reads as zero.

2. The host writes the data byte to the data register

3. The host writes the command register with the the parameter address Ored with both the Request bit (bit 13) and the Write bit (bit 15)

REFERENCE INFORMATION

SSLBP

LOCAL PARAMETERS

There are a number of local SSLBP read only parameters that are useful for interface software and drivers to access using the local read operations:

LOCAL PARAMETER	ADDRESS	DESCRIPTION
INTERFACE_TYPE	0x0000	0x12 for SSLBP
INTERFACE_WIDTH	0x0001	Data port width (8)
MAJORREV	0x0002	Major SSLBP firmware revision
MINORREV	0x0003	Minor SSLBP firmware revision
GP_INPUTS	0x0004	Number of GP input bits (0 for SSLBP)
GP_OUTPUTS	0x0005	Number of GP output bits (0 for SSLBP)
PROCESSOR_TYPE	0x0006	0xD8 for Dumb8
CHANNELS	0x0007	1 to 8 depending on configuration

REFERENCE INFORMATION

SSLBP

NORMAL START

When the FPGA is first configured or after a STOP command, all local communication, error and status parameters are initialized and all LBP communication channels are idle. A normal START command begins to establish communications with all remote LBP devices. A normal start command is issued by writing a Start bit with type bits of 0,0,1 with a bit mask of the desired channels to start in the low byte, This is 0x9NN hex where NN is the bitmask of channels to start. This command is written to the command register to start the selected channels.

Once a start command has been issued, all channels that are selected in the bit mask will be probed to determine if a LBP device exists. If a device exists on a channel, the SSLBP firmware will acquire the device name, and device unit number, and pointers to process data information from the remote device..

A normal start command also does a standard set of remote device setup operations when it detects a remote device. This setup includes clearing any faults, setting remote operational mode, and setting the outputs off. If no errors have occurred and all faults are clearable, the SSLBP firmware enters a "chatter" loop where it repeatedly sends output data of all 0's. This keeps the remote devices watchdog fed while waiting for the first DOIT command.

When the command completes (the command register is clear), the data register can be read to determine if all selected channels have started. A 1 bit in any position in the data register indicates that the corresponding channel has failed to start. If a channel has failed to start, more information about the failure can be determined by reading the CS register of the failed channel.

Once a DOIT command has been executed, the firmware no longer "chatters" and it becomes the responsibility of the host interface to continue sending DOIT commands at a rate sufficient to feed the remote devices watchdog (faster than 20 Hz with the default 50 mS watchdog timeout period). If this is not done, the remote device’s watchdog will bite, disabling its outputs and setting the fault flag. This will require a channel stop followed by a channel start to resume normal operations.

REFERENCE INFORMATION

SSLBP

STOP ALL

A STOPALL command is issued to stop all channel communication. *STOPALL resets all channel variables and should always be issued by a driver when initializing the SSLBP interface.* A STOPALL followed by a START command can be used after a fault condition to re-establish communication with the remote LBP devices. Device discovery is only done once when START command is issued to a STOPed SSLBP. This means that if cabling, devices, or device hardware modes are are changed, a STOPALL command followed by a START command must be issued by the host to detect the changes. A STOPALL command is 0x0800.

STOP INDIVIDUAL CHANNELS

In addition to stopping all channels, a individual stop command can be issued. A individual stop command include a bitmask of the channels to stop in the least significant 8 bits of the command (the N bits), that is a stop channel 1 command would be 0x802. The intended use of individual stop is per channel error recovery. It should not be used for normal interface startup as it does not reset channel variables, that is a 0x8FF command (stop all individual channels) is not equivalent to a 0X800 (STOPALL) command.

DOIT

In normal operation SSLBP is designed to send host data from the interface registers to the remote device and request data from the remote device for presentation in the interface registers to the host. This SSLBP function is designed for high speed real time operation. Synchronization with the host is accomplished with the DOIT command.

When the host writes a DOIT command,, all outgoing process data from the host is sent to the remote devices and incoming process data is requested. Completion of the DOIT command is signaled by SSLBP clearing the COMMAND register. A DOIT command is completed when al requested channel transfers have completed or timed out. After the completion of a successful DOIT command, the incoming process data from the remote can be read.

A DOIT command contains the DOIT bit and an 8 bit mask in the 8 LSBs that selects the channels that will be requested to transfer data. *A DOIT should not be requested on an inactive channel, that is a channel that did not start*. After DOIT command completion the data register will contain a bit mask of channel status data. If any bit is set in the data register, it indicates a problem with the transfer (all zeros indicates no faults or errors).

The data register contents returned after a DOIT command can be used to minimize host access cycles by avoiding the need to read the per channel status registers. If detailed fault information is desired, the CS register can be read on any channel that shows a failed transfer.

REFERENCE INFORMATION

SSLBP

PER CHANNEL INTERFACE DATA REGISTERS

SSLBP supports three 32 bit interface data registers per channel. These are called interface register 0, interface register 1, and interface register 2. These are read/write registers with independent incoming and outgoing data. These registers are used for both setup/discovery data when starting a data link and process data once the link is running. When a start command is issued and has successfully completed, per channel setup data will be available in the interface registers.

PER CHANNEL CONTROL AND STATUS REGISTERS

SSLBP has a 32 bit control and status register for each channel. Like the interface data registers, these registers are used both for data link startup information and for status when the link is in operation.

REMOTE MODES

Some remote devices have software selectable modes that determine the specific data transferred for each DOIT command. These modes are selected by writing the mode number to the most significant byte of the remote channels CSR before a START or SETUP START command is issued. A default value of 0x00000000 should be written to all CSRs if MODE is not used.

REMOTE MODE IS WRITTEN TO CSR MS BYTE BEFORE START

CS REG	MODE	0	0	0.

INTERFACE AND CS REGISTER DATA AT START

After a successful start command (either setup start or normal start), Interface register 0 reports the remote device's unit number. This is the number printed on the card label. Interface register 1 reports the remote device's 4 letter name (LSB first). Interface register 2 reports the remote devices global table of contents pointer (GTOCP) and process table of contents pointer (PTOCP) for the currently selected remote device mode. The GTOCP and PTOCP will be 0x0000 for devices that do not support process data discovery. *Note that the setup data will be overwritten with process data once the first DOIT command is issued.*

READ DATA FROM PER CHANNEL INTERFACE REGISTERS AFTER START

CS REG	X	COM_STATE	STATUS	LOCAL FLT.
INTERFACE 0	UNIT# BYTE 3	UNIT# BYTE 2	UNIT# BYTE 1	UNIT# BYTE 0
INTERFACE 1	NAME BYTE 3	NAME BYTE 2	NAME BYTE 1	NAME BYTE 0
INTERFACE 2	GTOCP BYTE1	GTOCP BYTE 0	PTOCP BYTE1	PTOCP BYTE 0

REFERENCE INFORMATION

SSLBP

CS REGISTER AFTER START

The CS register is used for local SSLBP, and remote LBP device status and control information. Read access returns status information in both normal and setup mode. In normal mode, writes to the CS register are not used. After a normal start or setup start the CS register has the following format:

Byte3 = X undefined for SSLBP versions < 29, remote fault for versions >28 (See CS REGISTER AFTER DOIT section)

Byte2 = COM_STATE Communication state code (debug only)

Byte1 = Communication status code (0x00 for OK)

Bit 7 = CommunicationNotReady

Bit 6 = NoRemoteID

Bit 5 = CommunicationError

Bit 0 = RemoteFault

Byte0 = Local Communication faults (sticky, cleared only by STOP)

Bit 7 = TooManyerrors

Bit 6 = RemoteFault

Bit 5 = SerialBreakError

Bit 4 = ExtraCharacterError

Bit 3 = TimeoutError

Bit 2 = OverrunError

Bit 1 = InvalidCookieError

Bit 0 = CRCError

REFERENCE INFORMATION

SSLBP

CS REGISTER AFTER DOIT

After a successful DOIT command, or normal start with SSLBP versions >28 bytes 0 through 2 of CS register are the same as after a start command but in addition, the previously invalid byte 3 of the CS register contains remote fault information:

Byte3 = REMOTE_FAULTS

Bit 7 = LBPCOMFault

Bit 6 = IllegalMode Fault

Bit 5 = LowVoltageFault

Bit 4 = HighVoltageFault

Bit 3 = OverCurrentFault

Bit 2 = OverTempFault

Bit 1 = NoEnableFault

Bit 0 = WatchdogFault

REFERENCE INFORMATION

SSLBP

PROCESS DATA DISCOVERY

The SSLBP interface provides information to allow the host to determine the name, number, units, sizes, types, directions, and scaling of process data elements. This information is read from the remote device via a setup mode start followed by a series of remote read operations.

Note to the bewildered: process data discovery and its complications are not needed to access the 7I76E via SSLBP. In fact the 7I76E's data can be accessed via SSLBP with no more than a few register reads and writes The sole purpose of process data discovery is to allow the driver to present nicely named and formatted data to the host without the driver having any built in knowledge of the remote device.

PROCESS TABLE OF CONTENTS

After a normal start or setup start command, the PTOCP word in the low word of interface register 2 is a pointer to the current process table of contents (PTOC) in the remote device.

If remote devices that do not support process device discovery are present, their PTOCP will be 0, and process data organization must be inferred from the remote device name.

Remote reads from this location will return the first entry in the PTOC. All PTOC entries are pointers with a size of 2 bytes. The end of the PTOC is marked with a 0 sentinel. Each PTOC entry points to a process data descriptor. Here is an example of a 5 entry PTOC (PDD is Process Data Descriptor)

ENTRY	ADDRESS	CONTENTS
0	PTOCP	POINTER TO PDD 0
1	PTOCP+2	POINTER TO PDD 1
2	PTOCP+4	POINTER TO PDD 2
3	PTOCP+6	POINTER TO PDD 3
4	PTOCP+8	POINTER TO PDD 4
5	PTOCP+10	0x0000 (END OF TABLE)

REFERENCE INFORMATION

SSLBP

PROCESS DATA DESCRIPTOR

Each PTOC entry points to a process data descriptor or a mode descriptor. Each process data descriptor is a record with fields for data size, data type, data direction, minimum and maximum values, the address of the process data and the unit name and process data name. Each process data element has a corresponding process data descriptor record. In addition there are mode descriptor records that indicate the current hardware and software modes of the remote device. The process data descriptor record structure is as follows:

FIELD NAME	FIELD LENGTH	DESCRIPTION
RECORD_TYPE	8 BITS	RECORD TYPE = 0xA0
DATA_SIZE	8 BITS	DATA SIZE IN BITS
DATA_TYPE	8 BITS	DATA ELEMENT TYPE
DATA_DIRECTION	8 BITS	DATA DIRECTION
PARAM_MIN	32 BITS	IEEE-754 FP PARM MIN
PARAM_MAX	32 BITS	IEEE-754 FP PARM MAX
PARAM_ADD	16 BITS	ADDRESS OF PARM
UNIT_STRING	VARIABLE	NULL TERM. STRING
NAME_STRING	VARIABLE	NULL TERM. STRING

PROCESS DATA DESCRIPTOR FIELDS

RECORD_TYPE

The RECORD_TYPE field is a single byte at the beginning of the process data descriptor for record typing and sanity checking. It is 0xA0 for process data records.

DATA_LENGTH

The DATA_LENGTH field is a single byte field that specifies the length of the process data element in bits. Minimum is 1 bit, maximum is 255 bits, however current SSLBP implementations are limited by the number of interface registers to a maximum of 96 bits.

REFERENCE INFORMATION

SSLBP

DATA_TYPE

The DATA_TYPE field is a single byte field that specifies the data type of the process data element. Data types are as follows:

NUMBER	DATA_TYPE	NOTE
0x00	PAD	To pad for byte alignment
0x01	BITS	Packed bits, LSB is BIT 0
0x02	UNSIGNED	Numeric unsigned
0x03	SIGNED	Numeric twos complement LSB first
0x04	NONVOL_UNSIGNED	Numeric unsigned
0x05	NONVOL_SIGNED	Numeric twos complement LSB first
0x06	STREAM	Continuous data stream
0x07	BOOLEAN	Any length non-zero = true

DATA_DIRECTION

The DATA_DIRECTION field is a single byte field that specifies the data direction. Valid Data direction bytes are as follows:

0x00	INPUT	(Read from remote)
0x40	BI_DIRECTIONAL	(Read from and written to remote)
0X80	OUTPUT	(Written to remote)

PARAMETER_MIN

The PARAMETER_MIN field is a 32 bit IEEE-754 floating point number that specifies the minimum value of the process data element. This is to allow the driver to present data in engineering units. Not valid for non-numeric data types

PARAMETER_MAX

The PARAMETER_MAX field is a 32 bit IEEE-754 floating point number that specifies the maximum value of the process data element. This is to allow the driver to present data in engineering units. Not valid for non-numeric data types.

REFERENCE INFORMATION

SSLBP

UNIT_STRING

The UNIT_STRING is a variable length null terminated string that specifies the units of the process data element

NAME_STRING

The NAME_STRING is a variable length null terminated string that begins immediately after the UNIT_STRING. It specifies the name of the process data element.

NUMERIC PROCESS DATA SCALING

Currently all numeric process data is simple unsigned or signed (twos complement) binary data. The process data element PARAM_MIN and PARAM_MAX values in conjunction with the DATA_SIZE can be used to scale this numeric data.

For unsigned data, PARAM_MIN corresponds to a value of 0 and PARAM_MAX corresponds to a value of (2 ^ DATA_SIZE) -1. Meaning scaled unsigned data is RAW_DATA*(PARAM_MAX-PARAM_MIN) / ((2 ^ DATA_SIZE) -1) +PARAM_MIN.

For signed data. PARAM_MIN corresponds the value -(2 ^ DATA_SIZE-1)-1 and PARAM_MAX corresponds the value (2 ^ DATA_SIZE-1)-1, meaning scaled signed data is RAW_DATA (PARAM_MAX-PARAM_MIN) / ((2 ^ DATA_SIZE-1) -1) +PARAM_MIN.

MODE DESCRIPTOR

In addition to the process data descriptors, the PTOC will have pointers to two mode descriptors. These are the currently selected hardware and software modes of the remote device.

FIELD NAME	FIELD LENGTH	DESCRIPTION
RECORD_TYPE	8 BITS	RECORD TYPE = 0xB0
MODE INDEX	8 BITS	WHICH MODE
MODE TYPE	8 BITS	MODE TYPE
UNUSED	8 BITS	UNUSED
MODE_NAME_STRING	VARIABLE	NULL TERM. STRING

MODE TYPES

Currently there are only two mode types, HWMODE = 0x00 and SWMODE = 0x01 these correspond to hardware (EEPROM or Jumper setting)and software (dynamically changeable operational modes)

REFERENCE INFORMATION

SSLBP

PROCESS DATA ELEMENT PACKING AND UNPACKING

Ultimately all process data is transferred to and from the host via the interface 0,1,2 registers.

The packing of outgoing process data elements into these interface registers and unpacking of incoming process data elements from these interface registers is done in the order of process data descriptors listed in the PTOC. Process data elements in PTOC order and process descriptor DATA_SIZE are packed into or unpacked from the interface registers from LSB to MSB and from interface register 0 through interface register 2.

Read data and bidirectional data is unpacked from the interface registers read by the host. Write data and bidirectional data is packed into the interface registers written by the host.

Before a DOIT command is written to start a data transfer cycle with the remote device, the host must write its packed outgoing process data (OPD in table below) to the interface registers. (The CS register not currently used for outgoing data/control so is not written)

HOST WRITES OUTGOING INTERFACE REGISTERS BEFORE DOIT

CS REG	MODE	X	X	X
INTERFACE 0	OPD BYTE 3	OPD BYTE 2	OPD BYTE 1	OPD BYTE 0
INTERFACE 1	OPD BYTE 7	OPD BYTE 6	OPD BYTE 5	OPD BYTE 4
INTERFACE 2	OPD BYTE 11	OPD BYTE 10	OPD BYTE 9	OPD BYTE 8

REFERENCE INFORMATION

SSLBP

PROCESS DATA ELEMENT PACKING AND UNPACKING

After the DOIT command has completed, the incoming process data (IPD in table below) can be read along with the local and remote faults.

HOST READS INCOMING INTERFACE REGISTERS AFTER DOIT

CS REG	REMOTE. FLT	COM_STATE	STATUS	LOCAL FLT.
INTERFACE 0	IPD BYTE 3	IPD BYTE 2	IPD BYTE 1	IPD BYTE 0
INTERFACE 1	IPD BYTE 7	IPD BYTE 6	IPD BYTE 5	IPD BYTE 4
INTERFACE 2	IPD BYTE 11	IPD BYTE 10	IPD BYTE 9	IPD BYTE 8

7I76E SPECIFIC PROCESS DATA EXAMPLE

Process data is remote device dependent and also dependent on remote device mode. The 7I76E isolated I/O supports 3 software modes.

REFERENCE INFORMATION

SSLBP

7I76E SPECIFIC PROCESS DATA EXAMPLE

In the default input/output mode the process data appears in the interface registers in the order shown:

7I76E OUTGOING PROCESS DATA FOR MODE (1)

CS REG	X	X	X	X
INTERFACE 0	SPINOUT 15..8	SPINOUT 7..0	TB5 OUTS 15..8	TB6 OUTS 7..0
INTERFACE 1	X	X	SPINDIR	SPINENA
INTERFACE 2	X	X	X	X

7I76E INCOMING PROCESS DATA FOR MODE (1)

CS REG	REMOTE. FLT	COM_STATE	STATUS	LOCAL FLT.
INTERFACE 0	TB5 INS 31..24	TB5 INS 23..16	TB6 INS 15..8	TB6 INS 7..0
INTERFACE 1	ANALOG3	ANALOG2	ANALOG1	ANALOG0
INTERFACE 2	X	X	X	X

Note that this information is just for user convenience as the process data organization in the interface registers can be determined by process data discovery.

REFERENCE INFORMATION

SSLBP

NORMAL MODE OPERATION

In normal mode the sequence of operations for a cyclic access with write before read is as follows:

Note steps 1 through 5 are setup operations and are only done once per session

1. Issue STOP ALL command (0x800), wait for COMMAND register clear to verify stop command completion.

2. Issue normal START command (0x9NN) with bitmask (NN) of channels to start.

3. Wait for COMMAND register clear to verify start command completion. (may be many mS)

4. Read data register to verify that all selected channels started (a 1 in any channel position bit means a fault in the channel that the bit represents)

5. Read device unit number (This can only be read before DOIT has been asserted)

6. Check command register, if not clear, cycle time is too short.

(Note the command register should never be written to when not clear except to issue a stop command or when written with the command ignore bit set)

7. Check data register, any 1 bits indicate previous DOIT command failed for in the corresponding channels

8. Read per channel Interface register 0 and interface register 1 for input process data

9. Write per channel output process data (for 7I76E) to interface 0 register and interface 1 register

10. Write DOIT command = 0x10NN where NN is the bit mask of channels to initiate transfers.

11. Wait for next cycle, at next cycle time, loop back to state 6

This sequence can be modified if a read-modify-write sequence is required, this requires polling the command register for send/receive completion. This will take a maximum of 100 uSec from the DOIT command to command register clear and valid input data.

REFERENCE INFORMATION

SETUP START

When the FPGA is first configured or after a stop all command, all LBP communication channels are idle. A SETUP START command first initializes and all local communication, error and status parameters and begins to establish communications with all remote LBP devices. Unlike the NORMAL START command, SETUP START does no device specific setup but instead creates a pass-through access mode that allows the host to read or write any remote LBP device parameter. This allows utilities to setup 7I76E volatile and non-volatile parameters, and allows the host to do process data discovery to determine the input and output process data information from the remote device.

SETUP MODE OPERATION

In setup mode the SSLBP interface is used as a passthrough device to allow reading and writing parameters to the remote LBP device.

REMOTE READ EXAMPLE:

For a remote word read, the sequence of operations is as follows:

1. Issue a STOPALL command (0x800), wait for COMMAND register clear to verify stop command completion.

2. Issue a setup START command (0xFNN) with bitmask (NN) of channels to start

3. Wait for COMMAND register clear to verify start command completion. (may be many mS)

4. Read data register to verify that all selected channels started (a 1 bit means a fault in the channel that the bit represents)

5. Write LBP word read command (0x45) in the MSByte ORed with the parameter address to the selected channels CS register. (0x4500PPPP)

6. Issue a DOIT Command

7. Wait for the command register to be clear

8. Check that the data register is clear, any set bits indicate an error

9. Read the returned data in the LS word of the selected channels Interface0 register

10. Repeat from step 5 for any additional remote data reads

Remote read byte, word, long and double are basically equivalent, the only difference being the LBP command (0x44,0x45,0x46,0x47 respectively) and the size of the data read from the interface register(s)

REFERENCE INFORMATION

SSLBP

REMOTE WRITE EXAMPLE:

For a remote word write, the sequence of operations is as follows:

1. Issue a STOPALL (0x800) command, wait for COMMAND register clear to verify stop command completion.

2. Issue a setup START command (0xFNN) with bitmask (NN) of channels to start

3. Wait for COMMAND register clear to verify start command completion. (may be many mS)

4. Read data register to verify that all selected channels started (a 1 bit means a fault in the channel that the bit represents)

5. Write the new parameter data to the selected channels Interface0 register (right justified)

6. Write LBP word write command (0x65) in the MSByte ORed with the parameter address to the selected channels CS register. (0x6500PPPP)

7. Issue a DOIT Command

8. Wait for the command register to be clear

9. Check that the data register is clear, any set bits indicate an error

Repeat from step 5 for any additional remote parameter writes

Remote write byte, word, long and double are basically equivalent, the only difference being the LBP command (0x64,0x65,0x66,0x67 respectively) and the size of the data written to the interface register(s)

REFERENCE INFORMATION

SSLBP

DISCOVERY SEQUENCE:

for process data discovery (of one channel) the sequence of operations is as follows:

Note that the first section acquires the PTOC and the second section reads the records pointed to by the PTOC. For brevity, the remote read sequence (steps 5 through 9 of the remote read procedure) will be listed here as "remote read"

FIRST PART, ACQUIRE PTOC:

1. Issue a STOPALL (0x800) command, wait for COMMAND register clear to verify stop command completion.

2. Issue a setup START command (0xFNN) with bitmask (NN) of channels to start

3. Wait for COMMAND register clear to verify start command completion. (may be many mS)

4. Read data register to verify that the selected channels started (a 1 bit means a fault in the channel that the bit represents)

5. Read PTOCP from interface register 2, of selected channel, if zero, remote device does not support discovery

6. Remote read word at PTOCP

7. If word data is 0, PTOC collection is complete goto step 11

8. Save value in local PTOC table, and increment local PTOC table index

9. Increment PTOCP value by 2 (as it is a word pointer)

10. Repeat from step 6

REFERENCE INFORMATION

SSLBP

DISCOVERY SEQUENCE

SECOND PART, READ PROCESS DESCRIPTOR AND MODE DESCRIPTOR RECORDS:

11. For each PTOC entry acquired in the previous step:

12. Remote read byte at PTOC+0

12. If byte is 0xA0, proceed to step 16, reading process data descriptor

14 If byte is 0xB0, proceed to step 25 reading mode descriptor

15. If byte is neither, there is a error

16. Remote read byte at PTOC+1 This is DATA_SIZE

17. Remote read byte at PTOC+2 This is DATA_TYPE

18. Remote read byte at PTOC+3 This is DATA_DIRECTION

19. Remote read long at PTOC+4 This is PARAM_MIN.

20. Remote read long at PTOC+8 This is PARAM_MAX

21. Remote read word at PTOC+10 This is PARAM_ADD (not used normally)

22. Read UNIT_STRING starting at PTOC+12

    Initialize CharPointer to PTOC+12

    repeat (remote read byte at CharPointer, increment CharPointer, if byte is 0: done)

23 Read NAME_STRING starting at CharPointer

    repeat (remote read byte at CharPointer, increment CharPointer, if byte is 0: done)

24. Repeat with next PTOC = step 11

REFERENCE INFORMATION

SSLBP

DISCOVERY SEQUENCE

SECOND PART, READ PROCESS DESCRIPTOR AND MODE DESCRIPTOR RECORDS:

25. Remote read byte at PTOC+1 This is MODE_INDEX

26. Remote read byte at PTOC+2 This is MODE TYPE

27. Read MODE_NAME_STRING starting at PTOC+4

    Initialize CharPointer to PTOC+4

    repeat (remote read byte at CharPointer, increment CharPointer, if byte is 0: done)

28. Repeat with next PTOC = step 1

29. Select next channel # and repeat from step 5

REFERENCE INFORMATION

LBP

LBP is a simple binary master slave protocol where the host sends read, write, or RPC commands to the 7I76E, and the 7I76E responds. All controller communication to the 7I76E is done via LBP. LBP commands always start with a command header byte. This header specifies whether the command is a read or write or RPC, the number of address bytes(0, or 2), and the number of data bytes(1 through 8).The 0 address size option indicates that the current address pointer should be used. This address pointer will be post incremented by the data size if the auto increment bit is set. RPC commands allow any of up to 64 stored commands to be executed in response to the single byte command.

Note that the low level serial interface details presented here are not normally needed for 7I76E card access, as all the low level details are handed by the SSLBP code in the SSerial interface built into the FPGA, but is presented here for completeness.

LBP DATA READ/WRITE COMMAND

0	1	WR	RID	AI	AS	DS1	DS0

Bit 7.. 6 **CommandType:** Must be 01b to specify data read/write command

Bit 5 **Write:** 1 to specify write, 0 to specify read

Bit 4 **RPCIncludesData:** 0 specifies that data is from stream, 1, that data is from RPC (RPC only, ignored for non RPC commands)

Bit 3 **AutoInc:** 0 leaves address unchanged, 1 specifies that address is post incremented by data size in bytes.

BIT 2 **AddressSize:** 0 to specify current address, 1 to specify 2 byte address.

Bit 1..0 **DataSize:** Specifies data size, 00b = 1 bytes, 01b = 2 bytes, 10 b= 4 bytes, 011b = 8 bytes.

When multiple bytes are specified in a read or write command, the bytes are always written to or read from successive addresses. That is, a 4 byte read at location 0x21 will read locations 0x21, 0x22, 0x23, 0x24. The address pointer is not modified after the command unless the AutoInc bit is set.

REFERENCE INFORMATION

LBP

EXAMPLE LBP COMMANDS

Write 4 bytes (0xAA, 0xBB,0xCC,0xDD) to addresses 0x010,0x011,0x012,0x013 with AutoInc so that the address pointer will be left at 0x014 when the command is completed:

COMMAND BITS	CT1	CT0	WR	RID	AI	AS	DS1	DS0
LBPWrite: 2 add 4 data	0	1	1	0	1	1	1	0
Write Address LSB	0	0	0	1	0	0	0	0
Write Address MSB	0	0	0	0	0	0	0	0
Write data 0	1	0	1	0	1	0	1	0
Write Data 1	1	0	1	1	1	0	1	1
Write Data 2	1	1	0	0	1	1	0	0
Write Data 3	1	1	0	1	1	1	0	1

Write 2 more bytes (0xEE,0xFF) at 0x014 and 0x015:

COMMAND BITS	CT1	CT0	WR	RID	AI	AS	DS1	DS0
LBPWrite: 0 add 2 data	0	1	1	0	0	0	0	1
Write data 0	1	1	1	0	1	1	1	0
Write data 1	1	1	1	1	1	1	1	1

Read 8 bytes at 0x010,0x011,0x012,0x013,0x014,0x015,0x016,0x017:

COMMAND BITS	CT1	CT0	WR	RID	AI	AS	DS1	DS0
LBPRead: 2 add 8 data	0	1	0	0	0	1	1	1
Read Address LSB	0	0	0	1	0	0	0	0
Read Address MSB	0	0	0	0	0	0	0	0

REFERENCE INFORMATION

LBP

LOCAL LBP COMMANDS

In addition to the basic data access commands, there are a set of commands that access LBP status and control the operation of LBP itself. These are organized as READ and WRITE commands

LOCAL LBP READ COMMANDS

(HEX), all of these commands return a single byte of data.

0xC0 Get unit address

0xC1 Get LBP status

LBP Status bit definitions:

BIT 7 Reserved

BIT 6 Command Timeout Error

BIT 5 Invalid write Error (attempted write to protected area)

BIT 4 Buffer overflow error

BIT 3 Watchdog timeout error

BIT 2 Reserved

BIT 1 Reserved

BIT 0 CRC error

0xC2 Get CRC enable status (note CRCs are always enabled on the 7I76E)

0xC3 Get CRC error count

0xC4 .. 0xC9 Reserved

0xCA Get Enable_RPCMEM access flag

0xCB Get Command timeout (character times/10 for serial)

0xCC .. 0xCF Reserved

0xD0 .. 0xD3 4 character card name

REFERENCE INFORMATION

LBP

LOCAL LBP READ COMMANDS

0xD5 .. 0xD7 4 character configuration name (only on some configurations)

0xD8 Get low address

0xD9 Get high address

0xDA Get LBP version

0xDB Get LBP Unit ID (Serial only, not used with USB)

0xDC Get RPC Pitch

0xDD Get RPC SizeL (Low byte of RPCSize)

0xDE Get RPC SizeH (High byte of RPCSize)

0xDF Get LBP cookie (returns 0x5A)

REFERENCE INFORMATION

LBP

LOCAL LBP WRITE COMMANDS

(HEX), all of these commands except 0xFF expect a single byte of data.

0xE0 Reserved

0xE1 Set LBP status (0 to clear errors)

0xE2 Set CRC check enable (Flag non-zero to enable CRC checking)

0xE3 Set CRC error count

0xE4 .. 0xE9 Reserved

0xEA Set Enable_RPCMEM access flag (non zero to enable access to RPC memory)

0xEB Set Command timeout (in mS for USB and character times for serial)

0xEC .. 0xEF Reserved

0xF0 .. 0xF6 Reserved

0xF7 Write LEDs

0xF8 Set low address

0xF9 Set high address

0xFA Add byte to current address

0xFB .. 0xFC Reserved

0xFD Set unit ID (serial only)

0xFE Reset LBP processor if followed by 0x5A

0xFF Reset LBP parser (no data follows this command)

REFERENCE INFORMATION

LBP

RPC COMMANDS

RPC commands allow previously stored sequences of read/write commands to be executed with a single byte command. Up to 64 RPC's may be stored. RPC write commands may include data if desired, or the data may come from the serial data stream. RPCs allow significant command compression which improves communication bandwidth. When used with SSLBP, the 7I76Es process data transfer uses an RPC for efficiency

LBP RPC COMMAND

1	0	RPC5	RPC4	RPC3	RPC2	RPC1	RPC0

Bit 7..6 **CommandType:** must be 10b to specify RPC

Bit 5..0 **RPCNumber:** Specifies RPC 0 through 63

In the 7I76E LBP implementation, RPCPitch is 0x8 bytes so each RPC command has native size of 0x08 bytes and start 0x8 byte boundaries in the RPC table area. RPCs can cross RPCPitch boundaries if larger than RPCPitch RPCs are needed. The stored RPC commands consist of LBP headers and addresses, and possibly data if the command header has the RID bit set. RPC command lists are terminated by a 0 byte.

The RPC table is accessed at addresses 0 through RPCSize-1 This means with a RPCPitch of 0x8 bytes, RPC0 starts at 0x0000, RPC1 starts at 0x008, RPC2 starts at 0x0010 and so on.

Before RPC commands can be written to the RPC table,the RPCMEM access flag must be set. The RPCMEM access flag must be clear for normal operation.

REFERENCE INFORMATION

LBP

EXAMPLE RPC COMMAND LIST

This is an example stored RPC command list. Note RPC command lists must start at a RPCPitch boundary in the RPC table but an individual RPC list can extend until the end of the table. This particular RPC example contains 2 LBP commands and uses 7 bytes starting at 0x0028 (RPC5 for 0x08 pitch RPC table)

Command1. Writes two data bytes to address 0x10, 0x11 with 2 data bytes supplied by host

Command2. Reads two data bytes from address 0x12,0x13

COMMAND BITS	CT1	CT0	WR	RID	I	AS	DS1	DS0
LBPWrite: 2 add 2 data	0	1	1	0	0	1	0	1
Write Address LSB	0	0	0	1	0	0	0	0
Write Address MSB	0	0	0	0	0	0	0	0
LBPRead: 2 add 2 data	0	1	0	0	0	1	0	1
Read Address LSB	0	0	0	1	0	0	1	0
Read Address MSB	0	0	0	0	0	0	0	0
Terminator	0	0	0	0	0	0	0	0

The data stream for this RPC would consist of these 3 bytes:

COMMAND BITS	CT1	CT0	R5	R4	R3	R2	R1	R0
RPC 5	1	0	0	0	0	1	0	1
Data 0 for Command 1	0	1	0	1	0	1	0	1
Data 1 for Command 1	1	1	0	0	1	1	0	0

REFERENCE INFORMATION

SPECIAL RPCS

All remotes that work with SSLBP must implement three special RPCs, the ProcessDataRPC, The UnitNumberRPC, and the DiscoveryRPC.

DiscoveryRPC = 0xBB – Returns one byte that specifies process input data size in bytes, and one byte that specifies the process output data size in bytes. Following the size bytes are two 16 bit pointers, the first is the PTOC and the second is the GTOC. Note that the remote software mode must be set before issuing the discovery RPC.

UnitNumberRPC = 0xBC – Returns 32 bit unit number

ProcessDataRPC = 0xBD -- Normal process data transfer RPC followed by output data bytes. Returns one byte of remote fault information followed by input data. Number of input and output bytes are as specified in the DiscoveryRPC.

CRC

LBP on the 7I76E uses CRC checking of all commands and data to insure validity. The CRC used is a 8 bit CRC using the same polynomial as the Dallas/Maxim one wire devices (X^8+X^5++X^4+X^0). The CRC must be appended to all LBP commands and all returned data will have a CRC byte appended. Commands with no returned data (writes or RPCs with no reads) will still cause a CRC byte to be returned, this CRC byte will always be 00H.

FRAMING

Since LBP is a binary protocol with no special sync characters, the packet framing must be determined by other methods.

Framing is done by a combination of timing and pre-parsing the serial data. Timing based framing is used to reset the parser at gaps in the serial data stream. This provides fast re-synchronization to allow robust operation in noisy environments. The actual timeout used needs to be optimized for the operating mode. In setup mode where a non real-time OS may be communicating with the remote device, the frame timing is set to its maximum value (25.5 character times). This is equivalent to 2.1 mS at 115200 baud. This means that host communications cannot have more than 2.1 mS delays between characters in a command sequence when in setup mode.

In operate mode, command timeout is set by SSLBP to be 4 character times (16 uSec at 2.5M baud). The SSLBP firmware always sends commands in bursts without inter-character gaps so will always meet this timing. The timing is set short so that the parser on the remote device will always be reset and ready for the next command at the highest repetition rates even if data has been corrupted by noise so that incomplete commands have been received.

REFERENCE INFORMATION

LBP16

GENERAL

LBP16 is the simple register access protocol used by the 7I76E for all Ethernet communications.

LBP16 COMMANDS

LBP16 is a simple remote register access protocol to allow efficient register access over the Ethernet link. All LBP16 commands are 16 bits in length and have the following structure:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W	A	C	M	M	M	S	S	I	N	N	N	N	N	N	N

W Is the write bit (1 means write, 0 means read)

A Is the includes Address bit. If this is '1' the command is followed by a 16 bit address and the address pointer is loaded with this address. if this is 0 the current address pointer for the memory space is used. Each memory space has its own address pointer.

C Indicates if memory space itself (C=’0') or associated info area for the memory will be accessed (C= ‘1')

M Is the 3 bit memory space specifier 000b through 111b

S Is the transfer element size specifier (00b = 8 bits, 01b = 16 bits 10b = 32 bits and 11b = 64 bits)

I Is the Increment address bit. if this is '1' the address pointer is incremented by the element transfer size (in bytes) after every transfer ('0' is useful for FIFO transfers)

N Is the transfer count in units of the selected size. 1 through 127. A transfer count of 0 is an error.

LBP16 read commands are followed by the 16 bit address (if the A bit is set). LBP16 Write commands are followed by the address (if bit A is set) and the data to be written. LBP16 Addresses are always byte addresses. LBP data and addresses are little endian so must be sent LSB first.

REFERENCE INFORMATION

LBP16

INFO AREA

There are eight possible memory spaces in LBP16. Each memory space has an associated read only info area. The first entry has a cookie to verify correct access. The next two entries in the info area are the MemSizes word and the MemRanges word. Only 16 bit read access is allowed to the info area.

0000	COOKIE = 0X5A0N WHERE N = ADDRESS SPACE 0..7
0002	MEMSIZES
0004	MEMRANGES
0006	ADDRESS POINTER
0008	SPACENAME 0,1
000A	SPACENAME 2,3
000C	SPACENAME 4,5
000E	SPACENAME 6,7

INFO AREA MEMSIZES FORMAT

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W	T	T	T	T	T	T	T	X	X	X	X	A	A	A	A

W Memory space is Writeable

T Is type: 01 = Register, 02 = Memory, 0E = EEPROM, 0F = Flash

A Is access types (bit 0 = 8 bit, bit 1 = 16 bit etc)so for example 0x06 means 16 bit and 32 bit operations allowed

REFERENCE INFORMATION

LBP16

INFO AREA MEMRANGES FORMAT

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E	E	E	E	E	P	P	P	P	P	S	S	S	S	S	S

E Is erase block size

P Is Page size

S Ps address range

Ranges are 2^E, 2^P, 2^S. All sizes and ranges are in bytes. E and P are 0 for non-flash memory

REFERENCE INFORMATION

LBP16

INFO_AREA ACCESS

As discussed above, all memory spaces have an associated information area that describes the memory space. Information area data is all 16 bits and read-only.The hex command examples below are written in LSB first order for convenience. In the hex command examples, the NN is the count/increment field of the LBP16 command and the LLHH is the low and high bytes of the address.

Ispace 0 read with address	NN61LLHH	HostMot2 space
Ispace 0 read	NN21	
Ispace 1 read with address	NN65LLHH	Ethernet chip space
Ispace 1 read	NN25	
Ispace 2 read with address	NN69LLHH	Ethernet EEPROM space
Ispace 2 read	NN29	
Ispace 3 read with address	NN6DLLHH	FPGA flash space
Ispace 3 read	NN2D	
Ispace 6 read with address	NN79LLHH	LBP16 R/W space
Ispace 6 read	NN39	
Ispace 7 read with address	NN7DLLHH	LBP16 R/O space
Ispace 7 read	NN3D	

REFERENCE INFORMATION

LBP16

7I76E SUPPORTED MEMORY SPACES

The 7I76E firmware supports 6 address spaces. These will be described individually with example hexadecimal commands. The hex command examples below are written in LSB first order for convenience. In the hex command examples, the NN is the count/increment field of the LBP16 command and the LLHH is the low and high bytes of the address.

SPACE 0: HOSTMOT2 REGISTERS

This address space is the most important as it gives access to the FPGA I/O. This is a 64K byte address range space with 32 bit R/W access.

Space 0 read with address	NN42LLHH
Space 0 write with address	NNC2LLHH
Space 0 read	NN02
Space 0 write	NN82

REFERENCE INFORMATION

LBP16

SPACE 0: HOSTMOT2 REGISTERS

Example: read first 5 entries in hostmot2 IDROM:

85420004

85	; 85 == NN = 5 | Inc bit (0x80) so address is incremented after each access
42	; Read from space 0 with address included after command
00	; LSB of address (IDROM starts at 0x0400)
04	; MSB of address (IDROM starts at 0x0400)

Example: write 4 GPIO ports starting at 0x1000:

84C20010AAAAAAAABBBBBBBBCCCCCCCCDDDDDDDD

84	; 84 == NN = 4 | Inc bit so address is incremented after each access
C2	; Write to space 0 with address included after command
00	; LSB of address (GPIO starts at 0x1000)
10	; MSB of address (GPIO starts at 0x1000)
AAAAAAAA	; 32 bit data for GPIO port 0 at 0x1000
BBBBBBBB	; 32 bit data for GPIO port 0 at 0x1004
CCCCCCCC	; 32 bit data for GPIO port 0 at 0x1008
DDDDDDDD	; 32 bit data for GPIO port 0 at 0x100C

Note like all LBP16 data, write data is LS byte first

REFERENCE INFORMATION

LBP16

SPACE 1: ETHERNET CHIP ACCESS

Space 1 allows access to the KSZ8851-16 registers for debug purposes. All accesses are 16 bit.

Space 1 read with address	NN45LLHH
Space 1 write with address	NNC5LLHH
Space 1 read	NN05
Space 1 write	NN85

Example: read Ethernet chip CIDER register: 0145C000

01	; = NN = read 1 16 bit value
45	; read space 1 with address included
C0	; LSB of CIDER address
00	; MSB of CIDER address

SPACE 2: ETHERNET EEPROM CHIP ACCESS

This space is used to store the Ethernet MAC address, card name, and EEPROM settable IP address. The Ethernet EEPROM space is accessed as 16 bit data. The first 0x20 bytes are read only and the remaining 0x60 bytes are read/write.

Space 2 read with address	NN49LLHH
Space 2 write with address	NNC9LLHH
Space 2 read	NN09
Space 2 write	NN89

REFERENCE INFORMATION

LBP16

SPACE2: ETHERNET EEPROM CHIP ACCESS

Writes and erases require that the EEPROMWEna be set to 5A02. *Note that EEPROMWEna is cleared at the end of every LPB packet so the write EEPROMWEna command needs to prepended to all EEPROM write and erase packets.* For EEPROM write operations a LBP16 read operation should follow the write(s) for host synchronization.

Example: write EEPROM IP address with 192:168.0.32 (C0:A8:0:20 in hex)

01D91A00025A	Enable EEPROM area writes
82C920002000A8C0	Write 2 words to 0020 : C0A80020 (with inc). Note this must be in the same packet and the EEPROMWEna write

ETHERNET EEPROM LAYOUT

ADDRESS	DATA
0000	Reserved RO
0002	MAC address LS Word RO
0004	MAC address Mid Word RO
0006	MAC address MS Word RO
0008	Reserved RO
000A	Reserved RO
000C	Reserved RO
000E	Unused RO

REFERENCE INFORMATION

LBP16

ETHERNET EEPROM LAYOUT

ADDRESS	DATA
0010	CardNameChar-0,1 RO
0012	CardNameChar-2,3 RO
0014	CardNameChar-4,5 RO
0016	CardNameChar-6,7 RO
0018	CardNameChar-8,9 RO
001A	CardNameChar-10,11 RO
001C	CardNameChar-12,13 RO
001E	CardNameChar-14,15 RO
0020	EEPROM IP address LS word RW
0022	EEPROM IP address MS word RW
0024	EEPROM Netmask LS word RW (V16 and > firmware)
0026	EEPROM Netmask MS word RW (V16 and > firmware)
0028	DEBUG LED Mode (LS bit determines HostMot2 (0) or debug(1)) RW
002A	Reserved RW
002C	Reserved RW
002E	Reserved RW
0030..007E	Unused RW

REFERENCE INFORMATION

LBP16

SPACE 3: FPGA FLASH EEPROM CHIP ACCESS

Space 3 allows access to the FPGAs configuration flash memory. All flash memory access is 32 bit. Flash memory access is different from other memory spaces in that it is done indirectly via a 32 bit address pointer and 32 bit data port.

Space 3 read with address NN4ELLHH

Space 3 write with address NNCELLHHDDDDDDDD

Space 3 read NN0E

Space 3 write NN8E

FLASH MEMORY REGISTERS

Flash memory spaces have only 4 accessible registers:

ADDRESS	DATA	
0000	FL_ADDR	32 bit flash address register
0004	FL_DATA	32 bit flash data register
0008	FL_ID	32 bit read only flash ID register
000C	SEC_ERASE	32 bit write only sector erase register

Unlike other memory spaces, flash memory space is accessed indirectly by writing the address register (FL_ADDR) and then reading or writing the data (FL_DATA). The flash byte address is automatically incremented by 4 each data access.

Note that reads can read all of flash memory with consecutive read operations but write operations can only write a flash page worth of data before the page write must be started. Also unless you are doing partial page writes, page write should always start on a page boundary.

The page write is started by writing the flash address, reading the flash address, reading flash data, reading flash ID or issuing a erase sector command. For host synchronization, a read operation should follow every sector erase or page write.

REFERENCE INFORMATION

LBP16

SPACE 3: FPGA FLASH EEPROM CHIP ACCESS

Example: read 1024 bytes (0100h doublewords) of flash space at address 00123456:

01CE000056341200	Write FL_ADDR (0000) with pointer (0x00123456)
404E0400	Issue read command (FL_DATA = 0004) With count of 0x40 double words (256 bytes). Note do not use LBP16 increment bit! Flash address always autoincremented
400E	Next 0x40 doublewords = 256 bytes
400E	Next 0x40 doublewords = 256 bytes
400E	Next 0x40 doublewords = 256 bytes

Note that this is close to the maximum reads allowed in a single LBP packet (~1450 bytes)

Writes and erases require that the EEPROMWEna be set to 5A03. *Note that EEPROMWEna is cleared at the end of every LPB packet so the write EEPROMWEna command needs to prepended to all flash write and erase packets. The following is written on separate lines for clarity but must all be in one packet for correct operation.*

Example: Write a 256 byte page of flash memory starting at 0xC000:

01D91A00035A	Write EEPROMWEna with 0x5A03
01CE000000C00000	Write flash address
40CE0400	Issue write flash data command with count
12345678	Doubleword 0
ABCD8888	Doubleword 1
...	
FFFFFFFF	Doubleword 63 (= 256 bytes)
014E0000	Read new address to commit write and so some data is returned for host synchronization (so host waits for write to complete)

REFERENCE INFORMATION

LBP16

SPACE 3: FPGA FLASH EEPROM CHIP ACCESS

Example: Erase flash sector 0x00010000:

01D91A00035A	Write EEPROMWEna with 0x5A03
01CE000000000100	Write flash address with 0x 00010000
01CE0C0000000000	Write sector erase command (with dummy 32 bit data = 0)
014E0000	Read flash address for host synchronization (this will echo the address _after_ the sector is erased)

REFERENCE INFORMATION

LBP16

SPACE 4 LBP TIMER/UTILITY AREA

Address space 4 is for read/write access to LBP specific timing registers. All memory space 4 access is 16 bit.

Space 4 read with address	NN51LLHH
Space 4 write with address	NND1LLHHDDDD
Space 4 read	NN11
Space 4 write	NN91DDDD

MEMORY SPACE 4 LAYOUT:

ADDRESS	DATA
0000	uSTimeStampReg
0002	WaituSReg
0004	HM2Timeout
0006	WaitForHM2RefTime
0008	WaitForHM2Timer1
000A	WaitForHM2Timer2
000C	WaitForHM2Timer3
000E	WaitForHM2Timer4
0010..001E	Scratch registers for any use

The uSTimeStamp register reads the free running hardware microsecond timer. It is useful for timing internal 7I76E operations. Writes to the uSTimeStamp register are a no-op. The WaituS register delays processing for the specified number of microseconds when written, (0 to 65535 uS) reads return the last wait time written. The HM2TimeOut register sets the timeout value for all WaitForHM2 times (0 to 65536 uS).

All the WaitForHM2Timer registers wait for the rising edge of the specified timer or reference output when read or written, write data is don't care, and reads return the wait time in uS. The HM2TimeOut register places an upper bound on how long the WaitForHM2 operations will wait. HM2Timeouts set the HM2TImeout error bit in the error register.

REFERENCE INFORMATION

LBP16

SPACE 6 LBP STATUS/CONTROL AREA

Address space 6 is for read/write access to LBP specific control, status, and error registers. All memory space 6 access is 16 bit. The RXUDPCount and TXUDPCount can be used as sequence numbers to verify packet reception and transmission.

Space 6 read with address	NN59LLHH
Space 6 write with address	NND9LLHHDDDD
Space 6 read	NN19
Space 6 write	NN99DDDD

MEMORY SPACE 6 LAYOUT:

ADDRESS	DATA
0000	ErrorReg
0002	LBPParseErrors
0004	LBPMemErrors
0006	LBPWriteErrors
0008	RXPktCount
000A	RXUDPCount
000C	RXBadCount
000E	TXPktCount
00010	TXUDPCount
00012	TXBadCount

REFERENCE INFORMATION

LBP16

MEMORY SPACE 6 LAYOUT:

ADDRESS	DATA	
0014	LEDMode	If LSb is 0, LEDs are "owned" by HostMot2, otherwise LEDs are local debug LEDs
0016	DebugLEDPtr	What variable in space 6 local debug LEDs show (default is RXPktCount).
0018	Scratch	Can be used for sequence numbers
001A	EEPROMWEna	Must be set to 5A0N to enable EEPROM or flash writes or erases (N is memory space of EEPROM or flash) Note that this is cleared at the end of every packet.
001C	LBPReset	Setting this to a non-zero value will do a full reset of the LBP16 firmware. The 7I76E will read ita IP address jumpers and re-assign its IP address. The 7I76E will be unresponsive for as much as ½ of a second after this command.
001E	FPGAICAP	FPGA ICAP-16 register to allow remote FPGA reload and other low level FPGA access.

ERROR REGISTER FORMAT

BIT	ERROR
0	LBPParseError
1	LBPMemError
2	LBPWriteError
3	RXPacketErr
4	TXPacketErr
5	HM2TimeOutError
6..15	Reserved

REFERENCE INFORMATION

LBP16

SPACE 7: LBP READ ONLY AREA

Memory space 7 is used for read only card information. Memory space 7 is accessed as 16 bit data.

Space 7 read with address	NN5DLLHH
Space 7 read	NN1D

MEMORY SPACE 7 LAYOUT:

ADDRESS	DATA	
0000	CardNameChar-0,1	
0002	CardNameChar-2,3	
0004	CardNameChar-4,5	
0006	CardNameChar-6,7	
0008	CardNameChar-8,9	
000A	CardNameChar-10,11	
000C	CardNameChar-12.13	
000E	CardNameChar-14,15	
0010	LBPVersion	
0012	FirmwareVersion	
0014	Option Jumpers	
0016	Reserved	
0018	RecvStartTS	1 uSec timestamps
001A	RecvDoneTS	For performance monitoring
001C	SendStartTS	Send timestamps are
001E	SendDoneTS	from ***previous*** packet

REFERENCE INFORMATION

LBP16

ELBPCOM

ELBPCOM is a very simple demo program in Python (2.x) to allow simple checking of LBP16 host communication to the 7I76E. ELBPCOM accepts hexadecimal LBP16 commands and data and returns hexadecimal results. Note that the timeout value will need to be increased to about 2 seconds to try flash sector erase commands.

```
import socket
s = socket.socket(socket.AF_INET,socket.SOCK_DGRAM,0)
sip  = "192.168.1.121"
sport = 27181
s.settimeout(.2)
while(2 >0):
 sdata = raw_input ('>')
 sdata = sdata.decode('hex')
 s.sendto(sdata,(sip,sport))
 try:
  data,addr = s.recvfrom(1280)
  print ('>'),data.encode('hex')
 except socket.timeout:
  print ('No answer')
```

Sample run:

```
>01420001                       ; read hostmot2 cookie at 0x100
> fecaaa55                      ; 7I76E returns 0x55AACAFE

>82492000                       ; read EEPROM IP address at 0x0020
> 450a5863                      ; 63:58:0A:45 = 99.88.10.69
                                ;(for example)

>01D91A00025A82C920000100a8C0 ; write EEPROM IP address
                                ;(at 0x0020) with
                                ; C0:A8:0:1 = 192.168.0.1
```

REFERENCE INFORMATION

SPECIFICATIONS

	MIN	MAX	NOTES
GENERAL			
HOST SUPPLY VOLTAGE 5V	4.5 VDC	5.5 VDC	
5V CURRENT	----	500 mA	No ext load.
STEP/DIR OUTPUTS			
STEP/DIR OUTPUT HIGH V	4V	----	10 mA source
STEP/DIR OUTPUT LOW V	----	1V	10mA sink
FIELD I/O			
VIN (FIELD I/O LOGIC POWER)	8VDC	32 VDC	
VIN POWER CONSUMPTION	----	1 W	Typ. 600 mW
FIELD POWER	5VDC	32VDC	
FIELD OUTPUT CURRENT	----	350 mA	Per output
(RESISTIVE LOADS AND INDUCTIVE LOADS WITH FLYBACK DIODE)			
FIELD OUTPUT CURRENT	----	60 mA	Per output
(INDUCTIVE LOADS WITH NO FLYBACK DIODE)			
PER DRIVER CHIP CURRENT	----	1.4A	Per chip
HIGH SPEED ENCODER INPUT			
INPUT COMMON MODE RANGE	-7	+12	Volts
INPUT TTL MODE THRESHOLD	1.4	1.8	Volts
DIFFERENTIAL MODE IMPEDANCE	131	135	Ohms
COUNT RATE	----	10 MHz	

REFERENCE INFORMATION

SPECIFICATIONS

		MIN	MAX	NOTES
HIGH SPEED ENCODER INPUT				
INPUT COMMON MODE RANGE		-7	+12	Volts
INPUT TTL MODE THRESHOLD		1.4	1.8	Volts
DIFFERENTIAL MODE IMPEDANCE		131	135	Ohms
COUNT RATE		----	10 MHz	
RS-422 INTERFACE				
MAXIMUM DATA RATE		----	10	MBIT/S
INPUT COMMON MODE RANGE		-7	+12	Volts
INPUT TERMINATION RESISTOR		131	135	Ohm
OUTPUT LOW	(24 mA sink)	----	.8	Volts
OUTPUT HIGH	(24 mA source)	VCC-.8	----	Volts

REFERENCE INFORMATION

SPECIFICATIONS

SPINDLE INTERFACE

REFERENCE VOLTAGE	5	15	Volts
(SPINDLE+ –> SPINDLE-)			
SUPPLY CURRENT	----	20	mA
ISOLATION VOLTAGE	----	500	Volts DC
NON-LINEARITY	----	1	% at 5KHz
DIR/ENA OUTPUT CURRENT	----	50	mA
DIR/ENA OUTPUT VOLTAGE	----	100	Volts DC
DIR/ENA ISOLATION VOLTAGE	----	500	Volts DC

EXPANSION I/O

OUTPUT VOLTAGE LOW	----	.4V	16 mA sink
OUTPUT VOLTAGE HIGH	2.9V	----	16 mA source

ENVIRONMENTAL

TEMPERATURE -C VERSION	0°C	70°C
TEMPERATURE -I VERSION	-40°C	85°C

REFERENCE INFORMATION

DRAWINGS