

# **7193 ETHERNET ANYTHING I/O MANUAL**

Version 1.0

This page intentionally not blank

# Table of Contents

GENERAL .....	1
DESCRIPTION .....	1
HARDWARE CONFIGURATION .....	2
GENERAL .....	2
BREAKOUT POWER OPTION .....	2
5V I/O TOLERANCE .....	2
PRECONFIG PULL-UP ENABLE .....	3
IP ADDRESS SELECTION .....	3
CONNECTORS .....	4
CONNECTOR LOCATIONS AND DEFAULT JUMPER POSITIONS .....	4
7193 I/O CONNECTOR PIN-OUT .....	5
POWER CONNECTOR PIN-OUT .....	7
JTAG CONNECTOR PIN-OUT .....	7
OPERATION .....	8
FPGA .....	8
FPGA PINOUT .....	8
IP ADDRESS SELECTION .....	8
HOST COMMUNICATION .....	8
UDP .....	8
LBP16 .....	8
WINDOWS ARP ISSUES .....	8
CONFIGURATION .....	9
FALLBACK .....	9
DUAL EEPROMS .....	9
EEPROM LAYOUT .....	10
BITFILE FORMAT .....	12
MESAFLASH .....	12
SETTING EEPROM IP ADDRESS .....	12
FREE EEPROM SPACE .....	13
FALLBACK INDICATION .....	13
FAILURE TO CONFIGURE .....	13
CLOCK SIGNALS .....	13
LEDS .....	14
PULLUP RESISTORS .....	14
I/O LEVELS .....	14
STARTUP I/O VOLTAGE .....	14

# Table of Contents

SUPPLIED CONFIGURATIONS .....	15
HOSTMOT2 .....	15
SVST8_4IM2 .....	15
SVST4_8 .....	15
SVST8_8IM2 .....	15
SVST1_4_7I47S .....	15
2X7I65 .....	15
SV12IM_2X7I48 .....	15
SV6_7I49 .....	15
PIN FILES .....	16
REFERENCE INFORMATION .....	17
LBP16	
LBP16 COMMANDS .....	17
INFO AREA .....	18
INFO AREA MEMSIZES FORMAT .....	18
INFO AREA MEMRANGES FORMAT .....	19
INFO AREA ACCESS .....	20
7I93 SUPPORTED MEMORY SPACES .....	21
SPACE0: HOSTMOT2 REGISTERS .....	21
SPACE1: ETHERNET CHIP ACCESS .....	23
SPACE2: ETHERNET EEPROM CHIP ACCESS .....	23
ETHERNET EEPROM LAYOUT .....	24
SPACE3: FPGA FLASH EEPROM CHIP ACCESS .....	26
FLASH MEMORY REGISTERS .....	26
SPACE4: LBP TIMER/UTIL REGISTERS .....	29
SPACE6: LBP STATUS/CONTROL REGISTERS .....	30
MEMORY SPACE 6 LAYOUT .....	30
ERROR REGISTER FORMAT .....	31
SPACE7: LBP READ ONLY INFORMATION .....	32
MEMORY SPACE 7 LAYOUT .....	32
ELBPCOM .....	33
SPECIFICATIONS .....	34
CARD DRAWING .....	35

# GENERAL

## DESCRIPTION

The MESA 7193 is a low cost, general purpose, FPGA based programmable I/O card with 100 BaseT Ethernet host connection. The 7193 that uses 50 pin I/O connectors with interleaved grounds and IO module rack compatible pinouts. The 7193 is compatible with all of Mesa's 50 pin daughtercards.

The 7193 has a simplified UDP host data transfer systems that allows operation in real time and compatibility with standard networks. The 7193 provides 48 I/O bits (24 per connector) All I/O bits are 5V tolerant and have pullup resistors. A power source option allows the 7193 to supply 5V or 3.3V power to breakout boards as desired. The 5V power option is protected by a PTC.

Firmware modules are provided for hardware step generation, quadrature encoder counting, PWM generation, digital I/O, Smart Serial remote I/O, BISS, SSI, SPI, UART interfaces and more. All motion control firmware is open source and easily modified to support new functions or different mixes of functions.

# HARDWARE CONFIGURATION

## GENERAL

Hardware setup jumper positions assume that the 7193 card is oriented in an upright position, that is, with the Ethernet connector towards the left and the I/O connectors towards the right.

## CONNECTOR POWER AND 5V TOLERANCE OPTION

The 7193 has the option to supply 5V or 3.3V power from 7193s 50 pin I/O connectors to each daughtercard.

The power option applies to both of the 7193s I/O connectors. The 5V power is protected by per connector PTC devices so will not cause damage to the 7193 or system if accidentally shorted. The daughtercard voltage also selects the pullup resistor supply voltage for both I/O connectors. Note that all current Mesa daughtercards use 5V.

<b>POS</b>	<b>FUNCTION</b>
W4 UP	5V DAUGHTERCARD AND PULLUP POWER
W4 DOWN	3.3V DAUGHTERCARD AND PULLUP POWER

The FPGA used on the 7193 has a 4V absolute maximum input voltage specification. To allow interfacing with 5V inputs, the 7193 has bus switches on all 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. In addition to selecting the daughtercard and pull-up resistor voltage, W4 controls the 5V I/O tolerance option. When W4 is on the default UP position, 5V tolerance mode is enabled. When W4 is in the DOWN position, 5V tolerance mode is disabled. Note that W4 controls 5V tolerance on all I/O connectors.

# HARDWARE CONFIGURATION

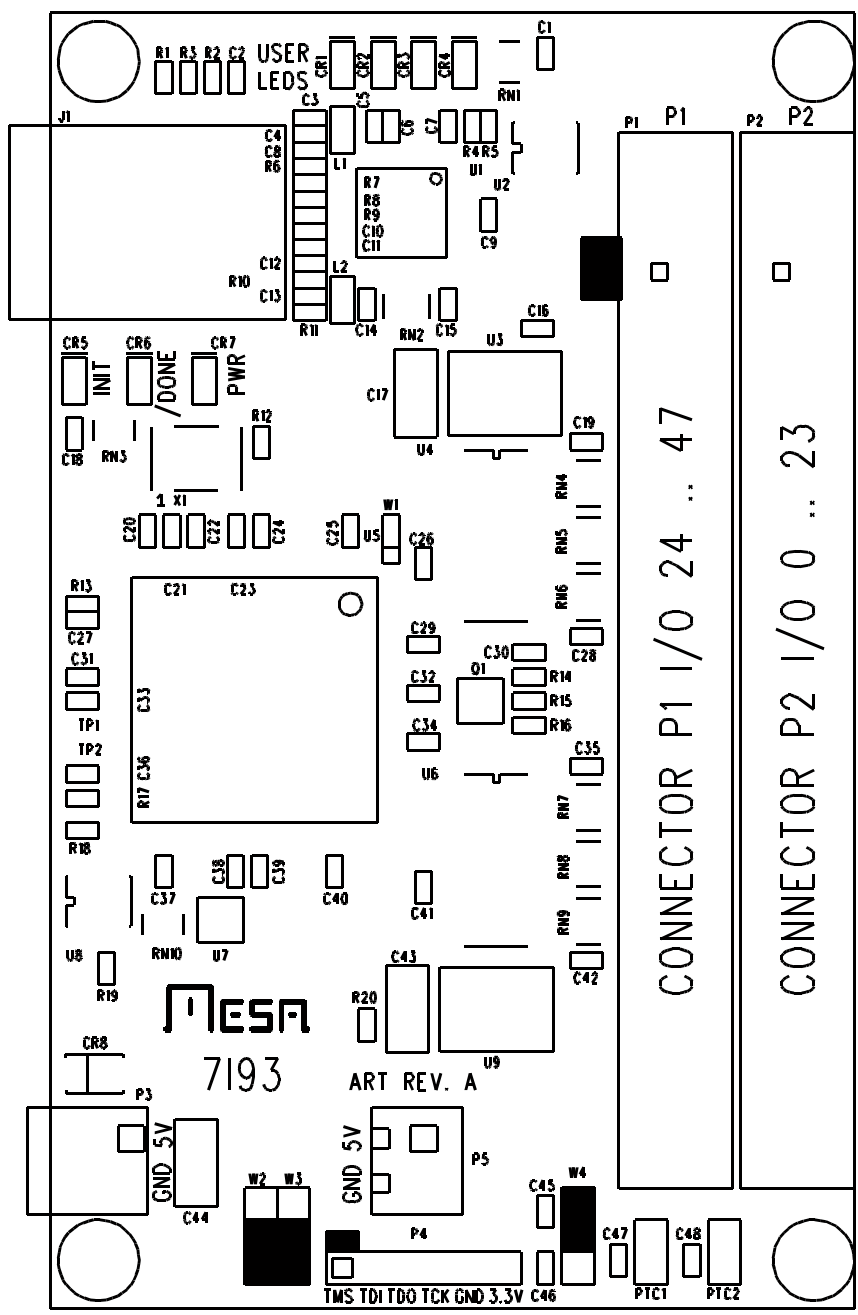
## IP ADDRESS SELECTION

The 7I93 has three options for selecting its IP address. These options are selected by Jumpers W2 and W3.

<b>W2</b>	<b>W3</b>	<b>IP ADDRESS</b>	
DOWN	DOWN	FIXED 192.168.1.121	(DEFAULT)
DOWN	UP	FIXED FROM EEPROM	
UP	DOWN	BOOTP	
UP	UP	INVALID	

# CONNECTORS

## CONNECTOR LOCATIONS AND DEFAULT JUMPER POSITIONS





# CONNECTORS

## I/O CONNECTORS

The 7193 has 2 I/O connectors, P1 and P2. 7193 IO connector pinouts are as follows:

### P2 CONNECTOR PINOUT

<b>PIN</b>	<b>FUNC</b>	<b>PIN</b>	<b>FUNC</b>	<b>PIN</b>	<b>FUNC</b>	<b>PIN</b>	<b>FUNC</b>
1	IO0	2	GND	3	IO1	4	GND
5	IO2	6	GND	7	IO3	8	GND
9	IO4	10	GND	11	IO5	12	GND
13	IO6	14	GND	15	IO7	16	GND
17	IO8	18	GND	19	IO9	20	GND
21	IO10	22	GND	23	IO11	24	GND
25	IO12	26	GND	27	IO13	28	GND
29	IO14	30	GND	31	IO15	32	GND
33	IO16	34	GND	35	IO17	36	GND
37	IO18	38	GND	39	IO19	40	GND
41	IO20	42	GND	43	IO21	44	GND
45	IO22	46	GND	47	IO23	48	GND
49	POWER	50	GND				

# CONNECTORS

## I/O CONNECTORS

### P1 CONNECTOR PINOUT

PIN	FUNC	PIN	FUNC	PIN	FUNC	PIN	FUNC
1	IO24	2	GND	3	IO25	4	GND
5	IO26	6	GND	7	IO27	8	GND
9	IO28	10	GND	11	IO29	12	GND
13	IO30	14	GND	15	IO31	16	GND
17	IO32	18	GND	19	IO33	20	GND
21	IO34	22	GND	23	IO35	24	GND
25	IO36	26	GND	27	IO37	28	GND
29	IO38	30	GND	31	IO39	32	GND
33	IO40	34	GND	35	IO41	36	GND
37	IO42	38	GND	39	IO43	40	GND
41	IO44	42	GND	43	IO45	44	GND
45	IO46	46	GND	47	IO47	48	GND
49	POWER	50	GND				

# CONNECTORS

## POWER CONNECTOR PINOUT

P5 is the 7193s power connector. P5 is a 3.5MM plug-in screw terminal block. P5 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

# OPERATION

## FPGA

The 7I93 use a Xilinx Spartan6 FPGA in a 144 ball TQFP package: XC6SLX9-2TQG144C.

## IP ADDRESS SELECTION

Initial communication with the 7I93 requires knowing its IP address. The 7I93 has 3 IP address options: Default, EEPROM, and Bootp, selected by jumpers W2 and W3. 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 7I93 address to be set by a DHCP/ BootP server. If BootP is chosen, the 7I93 will retry BootP requests at a ~1 Hz rate if the BootP server does not respond.

## HOST COMMUNICATION

The 7I93 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 7I93 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 7I93 data communication is done via UDP packets. The 7I93 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 7I80 for every transaction (reading RXUDPCount is suggested) and retrying failed transactions, or alternatively, setting up a static entry for the 7I93 in the ARP table. This is done with windows ARP command.

# **OPERATION**

## **CONFIGURATION**

The 7193 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 7193 is in the FPGA, a problem with configuration means that Ethernet access will not be possible. For this reason there is a backup system called fallback that will load a fallback configuration if the primary configuration is corrupted

### **FALLBACK**

The 7193 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.

# OPERATION

## EEPROM LAYOUT

The EEPROM used on the 7193 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	UNUSED/FREE
0x080000	UNUSED/FREE
0x090000	UNUSED/FREE
0x0A0000	UNUSED/FREE
0x0B0000	UNUSED/FREE
0x0C0000	UNUSED/FREE
0x0D0000	UNUSED/FREE
0x0E0000	UNUSED/FREE
0x0F0000	UNUSED/FREE

# OPERATION

## 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	UNUSED/FREE
0x170000	UNUSED/FREE
0x180000	UNUSED/FREE
0x190000	UNUSED/FREE
0x1A0000	UNUSED/FREE
0x1B0000	UNUSED/FREE
0x1C0000	UNUSED/FREE
0x1D0000	UNUSED/FREE
0x1E0000	UNUSED/FREE
0x1F0000	UNUSED/FREE

# OPERATION

## 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 and mesaflash.exe are provided to write configuration files to the 7193 EEPROM. These files depend on a simple SPI interface built into both the standard user FPGA bitfiles and the fallback bitfile.

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

The following examples assume the target 7193 is using the ROM IP address of 192.168.1.121.

```
mesaflash --device 7i93 --write FPGAFILE.BIT
```

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

```
mesaflash --device 7i93 --verify FPGAFILE.BIT
```

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

```
mesaflash --device 7i93 --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 7193:

```
mesaflash --device 7i93 --set ip=192.168.0.100
```

The above examples assume the 7193 has its default ROM IP address (192.168.1.121). If the 7193 is using another IP address, this must be specified on the command line with a -addr XX.XX.XX.XX command line argument.



## **OPERATION**

### **FREE FLASH MEMORY SPACE**

17 64K byte blocks are free on the 719. It is suggested that only the last three blocks, 0x1D0000 through 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 7193s FPGA. This can be fixed by running the configuration utility and re-writing the user configuration.

### **FAILURE TO CONFIGURE**

The 7193 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 7193s EEPROMs must be re-programmed via the JTAG connector or (faster) JTAG FPGA load followed by Ethernet EEPROM update.

### **CLOCK SIGNALS**

The 7193 has a single 50 MHz clock signal from an on card crystal oscillator. The clock can be multiplied and divided by the FPGA's 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.

# OPERATION

## LEDS

The 7I93 has 4 FPGA driven user LEDs (User 0 through User 3 = Green), and 2 FPGA driven status LEDs (red). The user LEDs can be used for any purpose, and can be helpful as a simple debugging feature. A low output signal from the FPGA lights the LED. See the 7I93IO.PIN file for FPGA pin locations of the LED signals. The status LEDs reflect the state of the FPGA's DONE, and /INIT pins. The /DONE LED lights until the FPGA is configured at power-up. The /INIT LED lights when the power on reset is asserted, when there has been a CRC error during configuration. When using Mesas configurations, the /INIT LED blinks when the fallback configuration has been loaded.

In addition to the FPGA driven LEDs, there are two power supply status LEDs on the bottom edge of the 7I93 card. There are the PWRFAIL and PWR LEDs. The PWRFAIL LED is illuminated if the locally regulated 1.2V or 3.3V power rails are out of specification. The PWR LED is illuminated when 3.3V is present.

## PULLUP RESISTORS

All 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).

## IO LEVELS

The Xilinx FPGAs used on the 7I93 have programmable I/O levels for interfacing with different logic families. The 7I93 does not support use of the I/O standards that require input reference voltages. All standard Mesa configurations use LVTTTL levels.

Note that even though the 7I93 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.

## 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.

# SUPPLIED CONFIGURATIONS

## HOSTMOT2

All supplied configurations are part of the HostMot2 motion control firmware set. All HostMot2 firmware is open source and easily extendible to support new interfaces or different sets of interfaces embedded in one configuration. For detailed register level information on Hostmot2 firmware modules, see the regmap file in the hostmot2 source code directory.

## SVST8\_4IM2

SVST8\_4IM2 is a 8 axis servo/ 4 axis stepmotor configuration with 8 PWM outputs, 8 encoder inputs with index mask, 4 hardware stepgenerators, a watchdog timer and GPIO.

## SVST4\_8

SVST4\_8 is a 4 axis servo/ 8 axis stepmotor configuration with 4 PWM outputs, 4 encoder inputs, 8 hardware stepgenerators, a watchdog timer and GPIO.

## SVST1\_4\_7I47S

SVST1\_4\_7I47S is a 4 axis stepmotor configuration with 1 PWM output for spindle, and 4 encoder inputs, a watchdog timer and GPIO. For the 7I47S card.

## 2X7I65

2X7I65 is a configuration for up to two 7I65 octal analog servo interface cards. It has 16 encoder inputs, 2 SPI ports, a watchdog timer and GPIO., 12 PWM outputs, a watchdog timer and GPIO.

## SV6\_7I49

SV6\_7I49 is a six axis servo configuration for use with the 7I49 resolver input daughter card. Its has a 6 channel resolver interface, 6 pwm channels, a watchdog timer and GPIO.

## **SUPPLIED CONFIGURATIONS**

### **PIN FILES**

Each of the configurations has an associated file with file name extension .pin that describes the FPGA functions included in the configuration and the I/O pinout. These are plain text files that can be printed or viewed with any text editor.

# REFERENCE INFORMATION

## LBP16

### LBP16 COMMANDS

LBP16 is a simple remote register access protocol to allow efficient register access over a serial 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 4 read with address	NN71LLHH	Timer/Utility space
Ispace 4 read	NN21	
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

### 7193 SUPPORTED MEMORY SPACES

The 7193 firmware supports 7 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 3 GPIO ports starting at 0x1000:

83C20010AAAAAAAABBBBBBBBCCCCCCCC

83 ; 83 == NN = 3 | 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

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 be 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 be 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
01CE00000000100	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 <u>after</u> 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 7180 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 7I93 will read its IP address jumpers and re-assign its IP address. The 7I93 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	RXPacketError
4	TXPacketError
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 <i>previous</i> 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 7I93. 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 ; 7I93 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

## SPECIFICATIONS

<b>POWER</b>	<b>MIN</b>	<b>MAX</b>	<b>NOTES:</b>
5V POWER SUPPLY	4.5V	5.5V	P4 supplied 5V
5V POWER CONSUMPTION:	----	5A	Depends on FPGA configuration and external load
MAX 5V CURRENT TO I/O CONNS	---	1000 mA	Each (PTC Limit)
TEMPERATURE RANGE -C version	0 °C	+70 °C	
TEMPERATURE RANGE -I version	-40 °C	+85 °C	

# REFERENCE INFORMATION

## CARD DRAWING

