# 4I74 QUADRATURE COUNTER MANUAL

1.2

This page intentionally not blank

# Table of Contents

# Table of Contents

# GENERAL

## DESCRIPTION

The 4I74 is a stackable PCI/104 card with eight quadrature up/down counters with quadrature count inputs, quadrature edge timestamping for velocity estimation and per channel index inputs. The 4I74 is intended for robotic, motor control, measurement, and instrumentation applications.

The 4I74 has selectable TTL or RS-422 levels on its quadrature and index inputs. TTL or RS-422 operation is jumper selectable on a per input basis. The differential RS-422 inputs are suited for longer cable lengths and are terminated.

In addition to the quadtrature counters, eight RS-422 serial ports are provided. Two ports can be run in RS-485 mode. These ports can be used for asynchronous serial communication, SSI or BISS encoder interfaces, Mesa Smart serial remote I/O communication or other uses depending on FPGA firmware.

The encoder connectors are compatible with the Mesa 4I30 and 4I36. The 4I74 counters can count in normal quadrature mode (4X) or up/down mode. Programmable digital filtering is used on encoder inputs to reject input noise. The 4I74 counters may cleared individually. Each counter has a option to be cleared or latched by either the rising or falling edge of the index signal.

Maximum count rate of the 4I74 with TTL inputs is 4 million counts per second. Maximum count rate with RS-422 inputs is 10 million counts per second. The 4I74 uses a FPGA chip for all counting and I/O so can be easily upgraded or modified in the field for specific requirements. The FPGA configuration flash memory can be updated from the host, no special cable or adapters are required.

# HARDWARE CONFIGURATION

## GENERAL

Hardware setup jumper positions assume that the 4I74 card is oriented in an upright position, that is, with the PC/104 connector on the bottom and the white PCB markings right side up.

## PC104-PCI SLOT NUMBER

The 4I74 card must be assigned a PC104/PCI slot number before use. This is done with the slot number jumpers on the 4I74 card. 2 jumpers. W27 and W28 determine the 4I74 slot assignment. The following table shown the jumper settings:

| W27 | W28 | SLOT | NOTES |
|---|---|---|---|
| DOWN | DOWN | 0 | DEFAULT |
| DOWN | UP | 1 | |
| UP | DOWN | 2 | |
| UP | UP | 3 | |

## PRECONFIG PULLUP ENABLE

The Xilinx FPGA on the 4I74 has the option of having weak pull-ups on all I/O pins at power-up or reset. The default is to disable the pull-ups To enable the built-in pull-ups, jumper W1 should be placed in the DOWN position. To disable the internal pull-ups, W1 should be in the UP position. It is suggested the W1 be left in the UP position.

## FPGA FLASH SELECT

To make recovery from FPGA configuration errors easier, there are two FPGA configuration flash memories on the 4I74 card. Jumper W20 selects between the two flash memories. That is, if one flash memory is inadvertently corrupted, the other one can be used to boot the 4I74, allowing the corrupted flash memory to be re-written. It is suggested that W20 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 W20 should be promptly restored to the UP position to allow the primary flash memory to be re-written.

| W20 | MEMORY |
|---|---|
| UP | PRIMARY (NORMAL OPERATION) |
| DOWN | SECONDARY (BACKUP) |

# HARDWARE CONFIGURATION

## RS-422 INPUT ENABLE

The 4I74 can accept either TTL or RS-422 (differential) encoder inputs. The choice of inputs is made with jumpers. Each jumper control the input mode for one input channel (A, B or I). When a jumper is in the ‘OUT’’ position (toward the encoder connector), RS-422 mode is selected. When a jumper is in the "IN" position (toward the center of the board), TTL mode is selected. The following table shows the correspondence between mode jumpers and input channels:

**Encoders on P1**

| | |
|---|---|
| W26,W24,W22 | Controls encoder input channel 0 DEFAULT=IN=TTL |
| W19,W17,W15 | Controls encoder input channel 1 DEFAULT=IN=TTL |
| W13,W11,W9 | Controls encoder input channel 2 DEFAULT=IN=TTL |
| W7,W5,W3 | Controls encoder input channel 3 DEFAULT=IN=TTL |

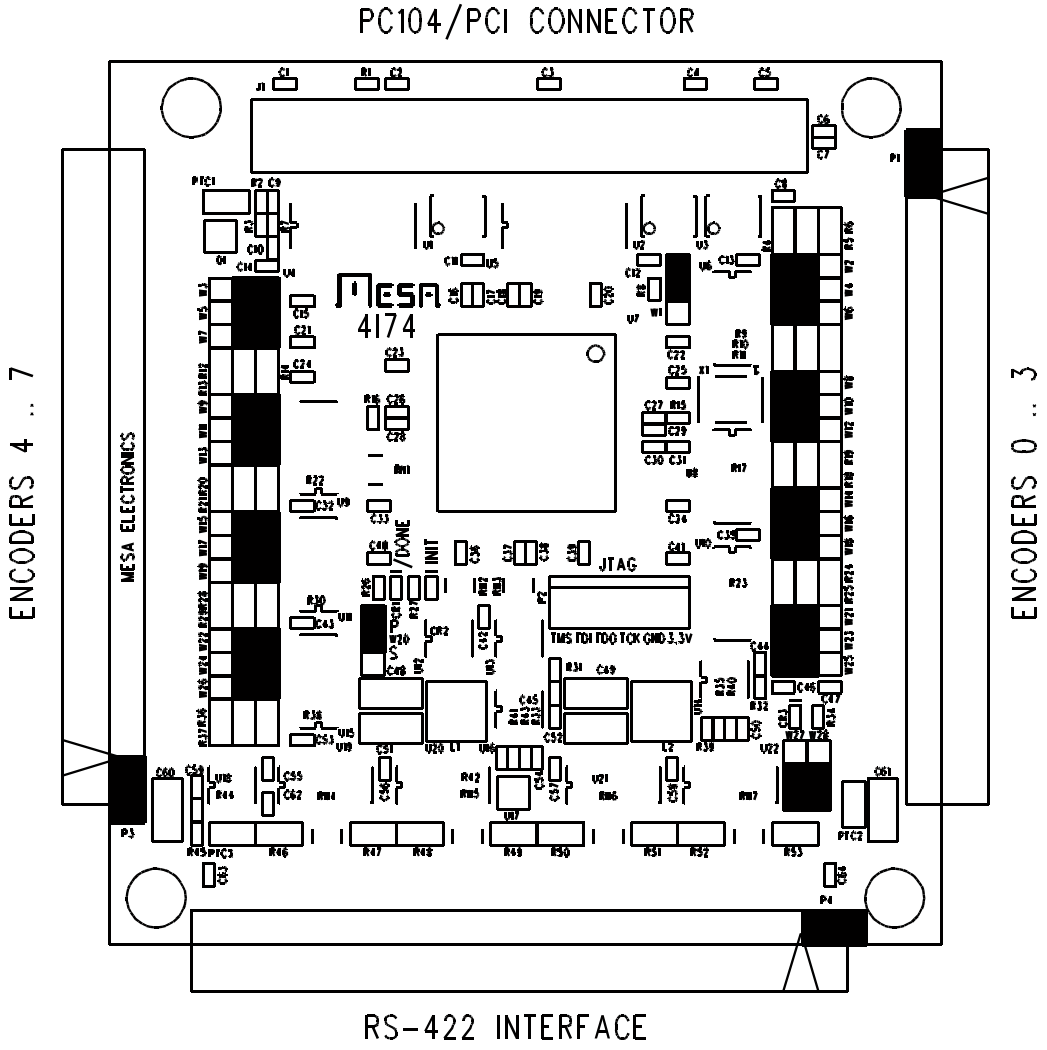
**Encoders on P3**

| | |
|---|---|
| W2,W4,W6 | Controls encoder input channel 4 DEFAULT=IN=TTL |
| W8,W10,W12 | Controls encoder input channel 5 DEFAULT=IN=TTL |
| W14,W16,W18 | Controls encoder input channel 6 DEFAULT=IN=TTL |
| W21,W23,W25 | Controls encoder input channel 7 DEFAULT=IN=TTL |

Note: each jumper controls a single encoder signal, jumpers above are listed in the order A,B,IDX.

# CONNECTORS

## CONNECTOR LOCATIONS AND DEFAULT JUMPER POSITIONS



# CONNECTORS

## ENCODER CONNECTORS

P1, and P3, are the 4I74's encoder connectors. These are 50 pin box headers that mate with standard 50 conductor female IDC connectors. Suggested compatible IDC receptacle is AMP PN 1-1658621-0. The 4I74 encoder pinouts match the Mesa 4I30 and 4I36, so the 4I74 can be used a a hardware compatible replacement for the 4I30 or 4I36. Note that TTL inputs connect to the /xxx inputs. The RS-422 inputs have 120 Ohm termination.

P1 encoder connector pinout is as follows:

**P1 CONNECTOR PINOUT — ENCODERS 0 .. 3**

| PIN | FUNC | PIN | FUNC | PIN | FUNC | PIN | FUNC |
|---|---|---|---|---|---|---|---|
| 1 | QA0 | 2 | /QA0 | 3 | GND | 4 | QB0 |
| 5 | /QB0 | 6 | GND | 7 | IDX0 | 8 | /IDX0 |
| 9 | GND | 10 | +5V | 11 | QA1 | 12 | /QA1 |
| 13 | GND | 14 | QB1 | 15 | /QB1 1 | 6 | GND |
| 17 | IDX1 | 18 | /IDX1 | 19 | GND | 20 | +5V |
| 21 | QA2 | 22 | /QA2 | 23 | GND | 24 | QB2 |
| 25 | /QB2 | 26 | GND | 27 | IDX2 | 28 | /IDX2 |
| 29 | GND | 30 | +5V | 31 | QA3 | 32 | /QA3 |
| 33 | GND | 34 | QB3 | 35 | /QB3 | 36 | GND |
| 37 | IDX3 | 38 | /IDX3 | 39 | GND | 40 | +5V |
| 41 | GND | 42 | GND | 43 | GND | 44 | GND |
| 45 | GND | 46 | +5V | 47 | +5V | 48 | +5V |
| 49 | +5V | 50 | +5V | | | | |

Note that the 5V power is fed through a 2A PTC device. The pinout is arranged so that the 50 conductor flat cable can be spilt into five 10 pin sections, four sections for encoders 0 through 3 and one section for 5V power/ground.

# CONNECTORS

## ENCODER CONNECTORS

### P3 CONNECTOR PINOUT – ENCODERS 4 .. 7

| PIN | FUNC | PIN | FUNC | PIN | FUNC | PIN | FUNC |
|---|---|---|---|---|---|---|---|
| 1 | QA4 | 2 | /QA4 | 3 | GND | 4 | QB4 |
| 5 | /QB4 | 6 | GND | 7 | IDX4 | 8 | /IDX4 |
| 9 | GND | 10 | +5V | 11 | QA5 | 12 | /QA5 |
| 13 | GND | 14 | QB5 | 15 | /QB5 | 16 | GND |
| 17 | IDX5 | 18 | /IDX5 | 19 | GND | 20 | +5V |
| 21 | QA6 | 22 | /QA6 | 23 | GND | 24 | QB6 |
| 25 | /QB6 | 26 | GND | 27 | IDX6 | 28 | /IDX6 |
| 29 | GND | 30 | +5V | 31 | QA7 | 32 | /QA7 |
| 33 | GND | 34 | QB7 | 35 | /QB7 | 36 | GND |
| 37 | IDX7 | 38 | /IDX7 | 39 | GND | 40 | +5V |
| 41 | GND | 42 | GND | 43 | GND | 44 | GND |
| 45 | GND | 46 | +5V | 47 | +5V | 48 | +5V |
| 49 | +5V | 50 | +5V | | | | |

Note that the 5V power is fed through a 2A PTC device. The pinout is arranged so that the 50 conductor flat cable can be spilt into five 10 pin sections, four sections for encoders 4 through 7 and one section for 5V power/ground.

# CONNECTORS

## RS-422 PORT CONNECTOR

Connector P4 is the RS-422 connector. It provides 8 full duplex RS-422 channels that can be used for communications (UART), SSI and BISS encoders or other applications depending on FPGA programming.

### P4 CONNECTOR PINOUT

| PIN | FUNC | PIN | FUNC | PIN | FUNC | PIN | FUNC |
|---|---|---|---|---|---|---|---|
| 1 | GND | 2 | RX0+ | 3 | RX0- | 4 | GND |
| 5 | TX0+ | 6 | TX0- | 7 | GND | 8 | RX1+ |
| 9 | RX1- | 10 | GND | 11 | TX1+ | 12 | TX1- |
| 13 | GND | 14 | RX2+ | 15 | RX2- | 16 | GND |
| 17 | TX2+ | 18 | TX2- | 19 | GND | 20 | RX3+ |
| 21 | RX3- | 22 | GND | 23 | TX3+ | 24 | TX3- |
| 25 | GND | 26 | RX4+ | 27 | RX4- | 28 | GND |
| 29 | TX4+ | 30 | TX4- | 31 | GND | 32 | RX5+ |
| 33 | RX5- | 34 | GND | 35 | TX5+ | 36 | TX5- |
| 37 | GND | 38 | RX6+ | 39 | RX6- | 40 | GND |
| 41 | TX6+ | 42 | TX6- | 43 | GND | 44 | RX7+ |
| 45 | RX7- | 46 | GND | 47 | TX7+ | 48 | TX7- |
| 49 | +5V | 50 | GND | | | | |

# OPERATION

## PCI ACCESS

The 4I74 normally uses 4I74 specific HostMot2 firmware which currently has a simple target only PCI core with a single Base Address Register (BAR 0). Card specific PCI identifiers are as follows:

| | |
|---|---|
| VENDOR ID | 0X2718 |
| DEVICE ID | 0x4174 |
| SUBSYSTEM VENDOR ID | 0x2718 |
| SUBSYSTEM DEVICE ID | 0x4174 |

The single base address register (BAR0) maps a 64K Byte region of non-cacheble 32 bit wide memory.

## QUADRATURE COUNTER

The standard 4I74 configuration contains 8 encoder counters. These counters are part of Mesa's HostMot2 firmware suite. Each counter has two 32 bit registers for reading counts, reading timestamps, reading latched counts and setting the counter mode. The two regisater are referred to as the QCR (Quadrature Count Register) and CCR (Counter Control Register)

### REGISTER ACCESS

All Hostmot2 registers are read at offsets from the 4I74 PCI base addresses. All register access is via 32 bit memory read or 32 bit memory writes. *Only 32 bit access is legal, byte or word or unaligned access will result in undefined operation*.

### QCR REGISTER ADDRESSING

| OFFSET | REGISTER |
|---|---|
| 0x3000 | Channel 0 QCR |
| 0x3004 | Channel 1 QCR |
| 0x3008 | Channel 2 QCR |
| 0x300C | Channel 3 QCR |
| 0x3010 | Channel 4 QCR |
| 0x3014 | Channel 5 QCR |
| 0x3018 | Channel 6 QCR |
| 0x301C | Channel 7 QCR |

# OPERATION

## QUADRATURE COUNTER

### QCR REGISTER DESCRIPTION

The QCR is the main count and timestamp register. Bits 0 through 15 are the current count value and bits 16 through 31 are the timestamp value. The timestamp is a latched copy of the free running timestamp register. The timestamp is latched anytime the count value changes. This allows for accurate velocity estimation at low count rates by measuring the time between encoder edges. Writing the QCR will clear the current count value.

### CCR REGISTER ADDRESSING

| OFFSET | REGISTER |
|---|---|
| 0x3100 | Channel 0 CCR |
| 0x3104 | Channel 1 CCR |
| 0x3108 | Channel 2 CCR |
| 0x310C | Channel 3 CCR |
| 0x3110 | Channel 4 CCR |
| 0x3114 | Channel 5 CCR |
| 0x3118 | Channel 6 CCR |
| 0x311C | Channel 7 CCR |

### CCR REGISTER DESCRIPTION

The CCR is the counter control register and contains bits that determine counter operation. The top 16 bits (bits 16 through 31) of the counter control register are the latched count when the current count is latched by a latch on index or latch on probe operation.

### CCR REGISTER BITS

| BIT | FUNCTION |
|---|---|
| Bit 31..16 | Read only latched count (for LatchOnIndex,LatchOnProbe) |
| Bit 15 | Quad Error: set if quadrature sequence error |
| Bit 14 | AB mask polarity: A*B must be high for index gate |
| Bit 13 | LatchOnProbe 1 = Latch count on probe |
| Bit 12 | ProbePolarity 1 = active high |
| Bit 11 | Quad filter (0 = 3 clocks 1 = 15 clocks) |
| Bit 10 | CounterMode 0 = Quadrature, 1 = up/down |

# OPERATION

## QUADRATURE COUNTER

### CCR REGISTER BITS

| BIT | FUNCTION |
|---|---|
| Bit 9 | UseIndexMask 1 = use mask |
| Bit 8 | IndexMask Polarity 1=active high |
| Bit 7 | ABgateIndex 1 = gate index signal with A,B |
| Bit 6 | JustOnce 1 = ClearOnIndex, LatchOnIndex, LatchOnProbe happen only once |
| Bit 5 | ClearOnIndex 1 = Clear count on index |
| Bit 4 | LatchOnIndex 1 = Latch count on index |
| Bit 3 | IndexPol 1 = active high |
| Bit 2 | Read only realtime index signal |
| Bit 1 | Read only realtime B signal |
| Bit 0 | Read only realtime A signal |

Note: All of the CCR bits are read/write except bits the latched count bits and bits 0,1,and 2.

### INDEX

It is suggested that systems that home to the index signal use LatchOnIndex rather than ClearOnIndex to avoid disturbing velocity calculations that depend on count differences.

### DIGITAL FILTER

All counter input signals pass through a digital filter to reduce susceptibility to noise. The per-counter Quad filter bit in the CCR determines if an input must be stable for 3 or 15 clocks before being recognized. At the default 33.333 MHz clock rate, with the Quad filter bit clear, the filter is set for 3 counts (90 ns) which will allow the maximum 10 MHz count rate (limited by input buffers). When the Quad filter bit is set, the filter is set for 15 counts (450 ns) allowing count rates up to approximately 4.5 MHz. If high count rates are not needed, the global clock to all encoders can be lowered via the QfilterRate register, increasing these filter times.

# OPERATION

## QUADRATURE COUNTERS

### GLOBAL REGISTERS

In addition to the per channel registers, there are three global registers that control quadrature counter operation. These registers are the TSDiv register, the TS register and the QFilterRate register.

### GLOBAL REGISTER ADDRESSING

| OFFSET | REGISTER | |
|---|---|---|
| 0x3200 | TSDiv | Write only, default is 0xFFFF (divide by one) |
| 0x3300 | TSCount | Read only |
| 0x3400 | QfilterRate | Write only, default is 0x0FFF (divide by one) |

### TSDIV REGISTER

The quadrature timestamp registers (per counter CCR bits 16 through 31) latch a free running timestamp counter. The rate of this free running counter can be set with the TSDiv register. Higher timestamp counter rates will result in higher timestamp resolution but if the rate is too high , the 16 bit timestamp may overflow more than once between readings, leading to inaccurate velocity estimation. To prevent this, the count rate of the timestamp counter should be set to a rate that will count to 2^16 no faster than the host polls the timestamp in the QCR. The TDDIV register sets the divide ratio of a 16 bit programmable divider. The timestamp counter rate is 33.33MHz/(TsDiv+2). The divide value is in the low 16 bits of the TSDiv register. Any TSDiv register value with the MSb (bit 15) set, will divide by 1, resulting in a 33.33 MHz timestamp clock.

### TSCOUNT REGISTER

The TSCount register is a free running up counter that provide the timestamp values for all encoder counters. The TSCount register counts at a rate determined by the TSDiv register and can be used for general purpose timing functions. The TSCount register value is 16 bit right justified, with bits 16 through 31 being undefined.

### QFILTERRATE REGISTER

The QfilterRate register sets the master clock to all encoder counter digital filters. The QFilterRate is set with a programmable divider value that is 12 bit right justified. The encoder counter digital filter rate is 33.33MHz/(QFilterRate+2). If the MSb (bit 11) is set, the QFilterRate will be 33.33 MHz.

# OPERATION

## OTHER FUNCTIONS

In addition to the standard encoder counters, the 4I74 can use all of the HostMot2 firmware suites modules for additional I/O. These functions include SSI and BISS encoder interfaces, UARTs, Interfaces to Mesa's SmartSerial remote I/O modules, PWM generation, hardware step/dir and quadrature signal generation, SPI interfaces and more. Specific firmware options can be created on request and all firmware source code is provided with an open source license.

### REGMAP FILE

For detailed interface information to available firmware modules, please refer to the regmap file in the HostMot2 source directory.

### GPIO TO IO PORT MAPPING

For input testing and output signal control via GPIO or embedded functions, the GPIO to I/O connector mapping is required. Note that in the 4I74 configurations, GPIO is mapped into 2 ports of 21 pins each, GPIO 0 through 20 (Port 0) and GPIO 21 through 41(Port 1).

| GPIO | SIGNAL | I/O CONNECTOR | PINS | DIR | PORT,BIT |
|---|---|---|---|---|---|
| 0 | QA0 | P1 | 1,2 | IN | 0,0 |
| 1 | IDX0 | P1 | 7,8 | IN | 0,1 |
| 2 | QB0 | P1 | 4,5 | IN | 0,2 |
| 3 | QA1 | P1 | 11,12 | IN | 0,3 |
| 4 | QB1 | P1 | 14,15 | IN | 0,4 |
| 5 | QA2 | P1 | 21,22 | IN | 0,5 |
| 6 | IDX1 | P1 | 17,18 | IN | 0,6 |
| 7 | QB2 | P1 | 24,25 | IN | 0,7 |
| 8 | IDX2 | P1 | 27,28 | IN | 0,8 |
| 9 | QB3 | P1 | 34,35 | IN | 0,9 |
| 10 | QA3 | P1 | 31,32 | IN | 0,10 |
| 11 | IDX3 | P1 | 37,38 | IN | 0,11 |

# OPERATION

### GPIO TO IO PORT MAPPING

| GPIO | SIGNAL | I/O CONNECTOR | PINS | DIR | PORT,BIT |
|---|---|---|---|---|---|
| 12 | RX0 | P4 | 2,3 | IN | 0,12 |
| 13 | RX1 | P4 | 8,9 | IN | 0,13 |
| 14 | RX2 | P4 | 14,15 | IN | 0,14 |
| 15 | RX3 | P4 | 20,21 | IN | 0,15 |
| 16 | TX0 | P4 | 5,6 | OUT | 0,16 |
| 17 | TX1 | P4 | 11,12 | OUT | 0,17 |
| 18 | TX2 | P4 | 17,18 | OUT | 0,18 |
| 19 | TX3 | P4 | 23,24 | OUT | 0,19 |
| 20 | RX4 | P4 | 26,27 | IN | 0,20 |
| 21 | RX5 | P4 | 32,33 | IN | 1,0 |
| 22 | RX6 | P4 | 38,39 | IN | 1,1 |
| 23 | RX7 | P4 | 44,45 | IN | 1,2 |
| 24 | TX4 | P4 | 29,30 | OUT | 1,3 |
| 25 | TX5 | P4 | 35,36 | OUT | 1,4 |
| 26 | TXEN45 | | | OUT | 1,5 |
| 27 | TXEN67 | | | OUT | 1,6 |
| 28 | TX6 | P4 | 41,42 | OUT | 1,7 |
| 29 | TX7 | P4 | 47,48 | OUT | 1,8 |

Note: GPIO 26 and GPIO 27 enable the TX4,TX5 pairs and the TX6,TX7 pairs respectively. Enables are active high.

# OPERATION

### GPIO TO IO PORT MAPPING

| GPIO | SIGNAL | I/O CONNECTOR | PINS | DIR | PORT,BIT |
|---|---|---|---|---|---|
| 30 | QA4 | P3 | 1,2 | IN | 1,9 |
| 31 | IDX4 | P3 | 7,8 | IN | 1,10 |
| 32 | QB4 | P3 | 4,5 | IN | 1,11 |
| 33 | QA5 | P3 | 11,12 | IN | 1,12 |
| 34 | QB5 | P3 | 14,15 | IN | 1,13 |
| 35 | QA6 | P3 | 21,22 | IN | 1,14 |
| 36 | IDX5 | P3 | 17,18 | IN | 1,15 |
| 37 | QB6 | P3 | 24,25 | IN | 1,16 |
| 38 | IDX6 | P3 | 27,28 | IN | 1,17 |
| 39 | QB7 | P3 | 34,35 | IN | 1,18 |
| 40 | QA7 | P3 | 31,32 | IN | 1,19 |
| 41 | IDX7 | P3 | 37,38 | IN | 1,20 |

### OUTPUT ENABLES

To be able to use the TX0 through TX7 and TXEN45 and TXEN67outputs, the HostMot2 DDR registers must have bits set to ‘1' in the positions corresponding to the output GPIO pins. In addition to use output functions other than GPIO, the ALTSRC register bits must also be set to ‘1' in output bit positions.

Write to offset 0x1100,0x000F0000 ; enable outputs on pins TX0 ..TX3

Write to offset 0x1200,0x000F0000 ; select FPGA function instead of GPIO

Write to offset 0x1104,0x000001F8 ; enable outputs on TX4..TX7 & TXEN’s

Write to offset 0x1204,0x000001F8 ; select FPGA function instead of GPIO

# OPERATION

## CONFIGURATION

### GENERAL

The 4I74 is configured at power up by a SPI FLASH memory. This flash memory is an 8M bit chip that has space for two configuration files. Since all PCI logic on the 4I74 is in the FPGA, a problem with configuration means that PCI 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 4I74 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 4I74 card, selectable via jumper W20. 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 W20 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 the PCI bus. The backup EEPROM is not write protected so if the primary EEPROM has been corrupted, you should always restore W20 to the UP position to avoid writing a bad configuration to both EEPROMS, necessitating a slow and awkward JTAG bootstrap.

# OPERATION

## CONFIGURATION

### EEPROM LAYOUT

The EEPROMs used on the 4I74 for configuration storage are M25P80. The M25P80 is a 8 M bit (1 M byte) EEPROM with sixteen 64KByte sectors. Configuration files are stored on sector boundaries to allow individual configuration file erasing and updating. Standard EEPROM sector layout is as follows:

| | |
|---|---|
| 0x00000 | BOOT BLOCK |
| 0x10000 | FALLBACK CONFIGURATION BLOCK 0 |
| 0x20000 | FALLBACK CONFIGURATION BLOCK 1 |
| 0x30000 | FALLBACK CONFIGURATION BLOCK 2 |
| 0x40000 | FALLBACK CONFIGURATION BLOCK 3 |
| 0x50000 | FALLBACK CONFIGURATION BLOCK 4 |
| 0x60000 | FALLBACK CONFIGURATION BLOCK 5 |
| 0x70000 | RESERVED |
| 0x80000 | USER CONFIGURATION BLOCK 0 |
| 0x90000 | USER CONFIGURATION BLOCK 1 |
| 0xA0000 | USER CONFIGURATION BLOCK 2 |
| 0xB0000 | USER CONFIGURATION BLOCK 3 |
| 0xC0000 | USER CONFIGURATION BLOCK 4 |
| 0xD0000 | USER CONFIGURATION BLOCK 5 |
| 0xE0000 | UNUSED/FREE |
| 0xF0000 | UNUSED/FREE |

# OPERATION

## CONFIGURATION

### BITFILE FORMAT

*The configuration utilities expects 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.*

### MFLASH

The DOS utility program MFLASH is provided to write configuration files to the 4I74 EEPROM unde DOS. MFLASH depends on a simple SPI interface built into both the standard user FPGA bitfiles and the fallback bitfile.

**MFLASH FPGAFILE.BIT**

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

**MFLASH FPGAFILE.BIT V**

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

**MFLASH FALLBACK.BIT FallBack**

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.

### MESAFLASH

Linux and Windows utility programs mesaflash and mesaflash.exe are provided to write configuration files to the 4I74 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 no command line arguments it will print usage information.

**mesaflash --device 4I74 --write FPGAFILE.BIT**

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

**mesaflash --device 4I74 --verify FPGAFILE.BIT**

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

**mesaflash --device 4I74 --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.

# OPERATION

## CONFIGURATION

### SPI INTERFACE DESCRIPTION

This is the register level description of the simple SPI interface to the 4I74s's configuration EEPROM. This hardware is built into all Mesa 4I74 configurations. This information is only needed if you are writing your own programming utility.

**DATA REGISTER** at offset 0x74 from 4I74 base address

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

**CONTROL REGISTER** at offset 0x70 from 4I74 base address

| X | X | X | X | X | DAV | BUSY | CS |
|---|---|---|---|---|---|---|---|
| X | X | X | X | X | R/O | R/O | R/W |

The SPI interface is very minimal, just enough hardware to avoid slow bit banging of the SPI data when reading or writing the configuration EEPROM. Operation is as follows: To transfer SPI data, CS is asserted low and an outgoing command/data byte is written to the data register.

This write to the data register causes the SPI interface to clear its DAV bit, shift out its outgoing data byte, and shift in its incoming data. This shifting is done at a fixed PCI Clock/3 rate or about 11 MHz. When the byte data transfer is done, The DAV bit is set in the control register. Host software can poll this bit to determine when the transfer is done. When the transfer is done the incoming data from the EEPROM can be read in the data register, and the next byte sent out.

Note that CS operation is entirely controlled by the host, that is for example with a 5 byte command sequence, the host must assert CS low, transfer 5 bytes with 5 write/read commands to the data register with per byte DAV bit polling and finally assert CS high when done.

# OPERATION

## CONFIGURATION

### FREE EEPROM SPACE

Three 64K byte blocks of EEPROM space are free when both user and fallback configurations are installed. It is suggested that only the last two blocks, 0xE0000 and 0xF0000 in the user area, be used for FPGA application EEPROM 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 4I74s FPGA. This can be fixed by running the configuration utility and re-writing the user configuration**.**

### FAILURE TO CONFIGURE

The 4I74 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 4I74s EEPROMs must be re-programmed via the JTAG connector or (faster) JTAG FPGA load followed by PCI EEPROM update. .

## CLOCK SIGNALS

The 4I74 has two FPGA clock signals. One is the PCI clock and the other is a 50 MHz crystal oscillator on the 4I74 card. Both clocks a can be multiplied and divided by the FPGAs clock generator block to generate a wide range of internal clock signals. Note that the PCI bus clock is often not known to a high degree of accuracy so for accurate timing applications, the on card 50 MHz oscillator should be used.

## LEDS

The 4I74 has three status LEDs. The status LEDs reflect the state of the FPGA's DONE, and /INIT pins and 3.3V power. The red /DONE LED (CR1) lights until the FPGA is configured at power-up. The red /INIT LED (CR2) lights when the power on reset is asserted, or when there has been a CRC error during configuration. The yellow power LED lights when the Local 3.3V power is OK. When using Mesas configurations, the /INIT LED blinks when the fallback configuration has been loaded**.**

# REFERENCE

## SPECIFICATIONS

| | MIN | MAX | NOTES: |
|---|---|---|---|
| POWER SUPPLY | 4.5V | 5.5V | 5V only |
| POWER CONSUMPTION: | ---- | 300 mA | No external power |
| COUNT RATE (RS-422) | 10 MHz | ---- | Faster available |
| COUNT RATE (TTL) | 4 MHz | ---- | |
| TTL INPUT PULLUP | 3.0K | 3.6K | Pull-up to 5V |
| 5V POWER PER CONNECTOR | ---- | 1.2A | 2A PTC/Connector |
| RS-422 /ENCODER TERMINATION | 114 | 126 | Ohms |
| TEMPERATURE RANGE | 0 | +70 | $^{o}$C |
| TEMPERATURE RANGE (-I Version) | -40 | +85 | $^{o}$C |