4I34
ANYTHING I/O
MANUAL

V1.4

Table of Contents

GENERAL 1

DESCRIPTION 1

HARDWARE CONFIGURATION 2

GENERAL 2
CONFIGURATION I/O DECODE 2
PULLUP ENABLE 2
CONNECTOR POWER 2

CONNECTORS 3

CONNECTOR LOCATIONS AND DEFAULT JUMPER POSITIONS 3
4I34 I/O CONNECTOR PIN-OUT 4
4I34M I/O CONNECTOR PIN-OUT 6
PC/104 BUS PIN-OUT 7

OPERATION 8

FPGA 8
CONFIGURATION 8
CONFIGURATION REGISTERS 8
SC4I34 8
CONFIGURATION DISABLE 9
LOW BYTE BUFFER 9
CLOCK SIGNALS 10
LEDS 10
I/O LEVELS 10

SUPPLIED CONFIGURATIONS 11

64/48 BIT, WORD ACCESSED I/O PORT 11
HOST BASED MOTOR CONTROLLER 11
MICRO CONTROLLER BASED MOTOR CONTROLLER 12
EVENT CAPTURE CARD 13
FREQUENCY/PERIOD COUNTER 14
PROGRAMMABLE INTELLIGENT I/O CARD 15
MICRO-STEPPING STEPPER CONTROLLER 16

GENERAL

DESCRIPTION

The MESA 4I34 is a general purpose programmable I/O card for the PC/104 bus. The 4I34 uses a 100K or 200K gate Xilinx FPGA for all logic, so it is truly an "Anything I/O" card. The FPGA is downloadable from the PC/104 bus side, allowing creation of almost any kind of specialized I/O function, even including micro-controllers in the FPGA.

Several pre-made functions are provided, including a 64 channel event capture card, an 8 channel host based servo motor controller, a 8 channel micro-controller based servo motor controller (micro-controller CPU built into FPGA), and a 8 channel, 32 bit timer counter card capable of running at 100 MHz. VHDL source is provided for all examples.

All I/O bits are 5V tolerant and can sink 24 mA. Pullup resistors are provided for all pins so that they may be connected directly to opto-isolators, contacts etc.

The 4I34 is available in 2 versions, the 4I34 with 64 I/O bits and the 4I34M with 48 I/O bits. The 4I34 uses two 40 pin connectors with proprietary pinouts, while the 4I34M uses two 50 pin connectors with I/O module rack compatible pinouts and interleaved grounds.

HARDWARE CONFIGURATION

GENERAL

Hardware setup jumper positions assume that the 4I34/4I34M card is oriented in an upright position, that is, with the PC/104 connectors towards the person doing the configuration.

CONFIGURATION I/O DECODE

The FPGA chip that is the heart of the 4I34 is based on static RAM technology. This means that at power up or after a reset the FPGA is empty and must be configured to do whatever specific task is required of it. This configuration information is sent to the over the PC/104 bus as a sequence of bytes. The I/O location where these bytes are sent is selected by 2 jumpers on the 4I34 card, W4 and W5. The following table shows the standard I/O addresses:

W4	**W5**	**BASE ADDRESS**
DOWN	DOWN	220
DOWN	UP	230
UP	DOWN	240
UP	UP	CARD DISABLED

The configuration addresses are determined by socketed PAL U5 and can be changed if needed. Configuration I/O uses 2 contiguous I/O locations starting at the base address. Configuration I/O can be disabled by the FPGA once loaded, to reclaim the I/O space.

PULLUP ENABLE

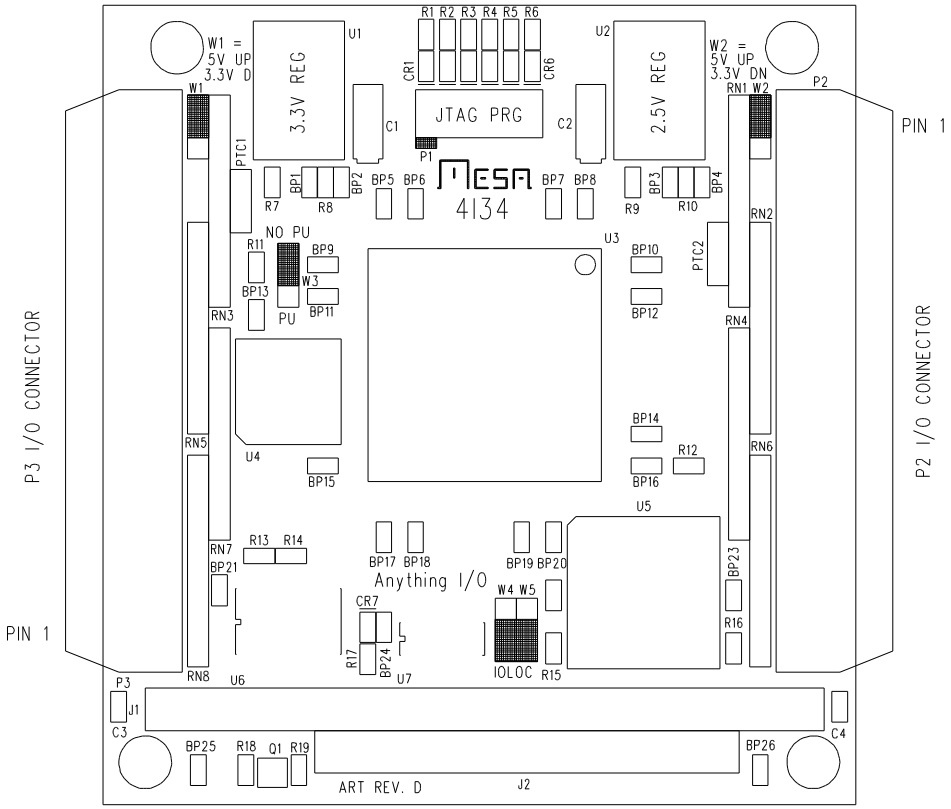
The Xilinx FPGA on the 4I34 has the option of having weak pull-ups on all I/O pins at power-up or reset. This may be desirable if the 4I34 is used without its 3.3k pull-ups on I/O pins. To enable the built-in pull-ups, jumper W3 should be placed in the down position. To disable the internal pull-ups (the default setting), W3 should be in the up position.

CONNECTOR POWER

The power connection on the I/O connectors can supply either 3.3V or 5V power. Supplied power should be limited to 400 mA total. W1 selects the power supplied to P3 and W2 selects the power supplied to P2. When W1 or W2 are in the up position, 5V power is supplied to the connector and associated pull-up resistors. When W1 or W2 are in the down position, 3.3V is supplied.

CONNECTORS

CONNECTOR LOCATIONS AND DEFAULT JUMPER POSITIONS



CONNECTORS

4I34 JTAG CONNECTOR

P1 is a JTAG programming connector. It is not normally used since the 4I34/4I34M can be programmed via the PC/104 interface, but can be useful when debugging.

P1 CONNECTOR PINOUT

PIN	FUNC	PIN	FUNC	PIN	FUNC
1	**TCK**	**2**	**TDI**	**3**	**TDO**
4	**TMS**	**5**	**GND**	**6**	**+5V**

I/O CONNECTORS

P2 and P3 are the 4I34s I/O connectors. These are latching 40 pin connector that mate with standard 40 conductor female IDC connectors. For information on which I/O pin connects to which FPGA pin, please see the 4I34IO.PIN file on the 4I34 distribution disk. 4I34 IO connector pinouts are as follows:

P2 CONNECTOR PINOUT

PIN	FUNC	PIN	FUNC	PIN	FUNC	PIN	FUNC
1	GND	2	IO0	3	IO1	4	IO2
5	IO3	6	IO4	7	IO5	8	IO6
9	IO7	10	POWER	11	GND	12	IO8
13	IO9	14	IO10	15	IO11	16	IO12
17	IO13	18	IO14	19	IO15	20	POWER
21	GND	22	IO16	23	IO17	24	IO18
25	IO19	26	IO20	27	IO21	28	IO22
29	IO23	30	POWER	31	GND	32	IO24
33	IO25	34	IO26	35	IO27	36	IO28
37	IO29	38	IO30	39	IO31	40	POWER

CONNECTORS

P3 CONNECTOR PINOUT

PIN	FUNC	PIN	FUNC	PIN	FUNC	PIN	FUNC
1	GND	2	IO32	3	IO33	4	IO34
5	IO35	6	IO36	7	IO37	8	IO38
9	IO39	10	POWER	11	GND	12	IO40
13	IO41	14	IO42	15	IO43	16	IO44
17	IO45	18	IO46	19	IO47	20	POWER
21	GND	22	IO48	23	IO49	24	IO50
25	IO51	26	IO52	27	IO53	28	IO54
29	IO55	30	POWER	31	GND	32	IO56
33	IO57	34	IO58	35	IO59	36	IO60
37	IO61	38	IO62	39	IO63	40	POWER

CONNECTORS

4I34M I/O CONNECTORS

P2 and P3 are the 4I34Ms I/O connectors. These are 50 pin box headers that mate with standard 50 conductor female IDC connectors. For information on which I/O pin connects to which FPGA pin, please see the 4I34MIO.PIN file on the 4I34 distribution disk. 4I34M IO connector pinouts are as follows:

P2 CONNECTOR PINOUT

PIN	FUNC	PIN	FUNC	PIN	FUNC	PIN	FUNC
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

4I34M I/O CONNECTORS

P3 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				

PC104 CONNECTIONS

The 4I34/4I34M implements connections for a fairly complete PC104 interface including full 16 bit I/O and memory interface (lower 1M only), All available interrupts, and 2 DMA channels. For information on which FPGA pin connects to which PC/104 pin, please see the 4I34104.PIN file on the distribution disk. Note that the .UCF files included with the example configurations can be used as starting points for customer implemented designs.

4I34 OPERATION

FPGA

The 4I34/4I34M use a Xilinx Spartan-II FPGA in a 208 pin QFP package The standard 4I34/4I34M use the 100K gate part. The 4I34-2/4I34M-2 use the 200K gate part.

CONFIGURATION

Before the 4I34/4I34M can do anything useful it must have its FPGA configuration data downloaded from the host CPU to the FPGA on the 4I34. This is done by writing a series of bytes from the configuration file to the 4I34 card's configuration data register. Before that data can be sent some control register bits must be set up. This is done in the control and status register.

CONFIGURATION REGISTERS

BASE ADDRESS CONFIGURATION DATA REGISTER

BIT	D7	D6	D5	D4	D3	D2	D1	D0
READ	CFGD7	CFGD6	CFGD5	CFGD4	CFGD3	CFGD2	CFGD1	CFGD0
WRITE	CFGD7	CFGD6	CFGD5	CFGD4	CFGD3	CFGD2	CFGD1	CFGD0

BASE ADDRESS+1 CONFIGURATION STATUS & CONTROL REGISTER

BIT	D7	D6	D5	D4	D3	D2	D1	D0
READ	XX	XX	XX	XX	XX	XX	XX	DONE
WRITE	XX	XX	XX	XX	LED	/WRITE	/PROG	/CS

SC4I34

A DOS executable SC4I34.EXE is provided to send configuration files to the 4I34. The Pascal and C source for this program is available on the distribution disk, and can be used as an example for writing a custom version of download software. SC4I34 is invoked with the FPGA configuration file and the 4I34 configuration base address on the command line:

```
SC4I34 FPGAFILE.BIN 220
```

would send the configuration file FPGAFILE.BIN to the 4I34 with a base address of 220 hex.

OPERATION

SC4I34

SC4I34 uses binary FPGA configuration files. These files can be standard Xilinx BIT files or PROM format files. The SC4I34 utility sends PROM files directly to the 4I34. BIT files have their headers stripped and are bit reversed before being sent to the 4I34.

Binary PROM format files can be created by translating an Exormacs format hex file to binary with the objcopy utility. The objcopy utility available on most Unix systems will do this conversion:

```
objcopy -S -O binary FPGAFILE.EXO FPGAFILE.BIN
```

A Windows version of objcopy is supplied on the 4I34 distribution disk.

CONFIGURATION DISABLE

The 4I34/4I34M cards can disable the configuration hardware after the FPGA chip has been configured. This is useful to reclaim I/O space used by the configuration hardware. It allows, for example, the final I/O address of a FPGA peripheral to be located at the same place as the configuration address. Several of the example configurations do this. Configuration disable is accomplished by driving the DIS pin on the FPGA high. See the 4I34104.PIN file for FPGA pin location of the DIS signal. To disable the configuration hardware once the FPGA is configured, simply assign the DIS pin a high level in the FPGA source file. The DIS pin will then be driven high once the FPGA is configured.

The disadvantage of using configuration disable is that you cannot re-configure the FPGA without reseting the host CPU. (Unless you implement the DIS pin as a host controlled bit)

LOW BYTE BUFFER

In order to allow configuration readback and meet PC/104 bus specifications, the low byte of the PC/104 data bus is buffered on the 4I34/4I34M card. This needs to be taken into account when doing FPGA designs. The buffer is controlled by 2 signals, /LBE and LBDIR. /LBE stands for Low Byte Enable and is the active low transceiver enable signal. /LBE would normally be driven by a chip select decode signal from the FPGA. LBDIR is the low byte direction control signal and is low for data out of the 4I34 card and high for data in. LBDIR would normally be driven by the FPGA with a signal like /IORD or /SMEMR. The VHDL sources on the distribution disk can be used as examples for driving these signals. See the 4I34104.PIN file for FPGA pin locations of the low byte buffer control signals.

OPERATION

CLOCK SIGNALS

The 4 FPGA clock signals on the 4I34 are routed to 4 separate clock sources. GCLK0 connects to a 50 MHz crystal oscillator on the 4I34 card, GCLK1 connects to the PC/104 /IOW signal, GCLK2 connects to the IO0 connector pin, and GCLK3 connects to the PC/104 SYSCLK signal

LEDS

The 4I34 has 6 status LEDS, and the 4I34M has 8 status LEDS. These 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 4I34104.PIN file for FPGA pin locations of the LED signals. *CR1 on the 4I34 uses a shared pin (init) so will light up when the FPGA is first powered-up or reset.*

IO LEVELS

The Xilinx FPGAs used on the 4I34 have programmable I/O levels for interfacing with different logic families. The 4I34 does not support use of the I/O standards that require input reference voltages, so only 5 I/O options can be used. The available I/O options are LVTTL (5V tolerant), PCI33_5 (5V tolerant), PCI33_3, PCI66_3, and LVCMOS2. Two of the I/O options allow 5V inputs. You must make sure to use 5V compatible IO options on the PC/104 part of your designs, because the PC/104 bus uses 5V levels. I/O levels of the users I/O pins do not have to be +5V compatible if not needed, but it is suggested to use one of the 5V tolerant I/O standards to avoid possible damage if larger than 3.3V signals are applied to I/O pins.

Note that even though the 4I34s FPGA can tolerate 5V signal inputs, its outputs will not swing to 5V. The outputs are push pull CMOS outputs 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, 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 tristate the output signals when no drive is desired (open drain).

SUPPLIED CONFIGURATIONS

IOPORT

The IOPORT configuration creates a simple 64 bit (4I34) or 48 bit (4I34M) parallel I/O port. The port base address is 220 hex and 16 (4I34) or 12 (4I34M) IO locations are used. The IOPORT is a word device, all accesses read or write 16 bit words. Each I/O bit can be individually programmed to be input or output. All I/O bits will be input on startup. For information on the register map of the IOPORT configuration, see the regmap file in the /configs/ioport directory of the 4I34 distribution disk.

I34LOOP

The IOPORT configuration provides a simple way to check that all the I/O pins are OK and that most of the host interface is working. A loopback program is provided for the 4I34 (I34LOOP) and 4I34M (I34MLOOP). Both programs depend on an external loopback cable between P2 and P3. The programs perform a rotating bit test with the P2 bits being inputs, the P3 being outputs and vice versa.The programs are invoked as follows:

I34LOOP 220 224 (for 4I34)

I34MLOOP 220 (for 4I34M)

HOSTMOT

The HOSTMOT configuration is a 8 channel (4I34) or 4 channel (4I34M) host based servo motor controller. Host based controllers depend on the host CPU to "close" the servo loop. This has advantages and disadvantages. One advantage is that less hardware is needed, since host based software does all the math and handles all the control bits. Another advantage is that since the host based software is easily examined and modified, it is more amenable to customization and is more useful as a teaching tool. One disadvantage is that host based motor controllers depend on fast interrupt response time, since the control loop is an interrupt driven background task. This means that host based motor controllers don't tend to work well with multitasking operating systems such as Windows or Unix. They will work with real-time operating systems or simple operating systems like DOS.

The HOSTMOT configuration consists of 8 (4 for 4I34M) 32 bit quadrature counters to measure motor position, 8 (4 for 4I34M) 12 bit PWM generators for driving the motor, plus miscellaneous control and interrupt generation logic. For information on the register map of the HOSTMOT configuration, see the regmap file in the /configs/hostmot directory of the 4I34 distribution disk.

Demonstration software provided with the 4I34 implements a PID + feedforward control loop, plus a ramp-up, slew, ramp-down, profile generator for position control applications. Demo program (newmove.exe) and sources are located in the /configs/hostmot/support directory of the 4I34 distribution disk.

SUPPLIED CONFIGURATIONS

SOFTDMC

The SOFTDMC configuration creates a 4 or 8 axis processor based servo motor controller, with the processor embedded in the FPGA. The SOFTDMC configuration has the advantage that the embedded processor takes care of all time critical functions, so it can control motor position and motions without host intervention.

The SOFTDMC configuration has programmable sample and PWM rates, and can operate 4 axis at up to 24 KHz sample rate and 8 axis at up to 12 KHz sample rate.

The control loop is a PID+F loop (F=feedforward) with 16 bit tuning parameters. Position and Velocity use 32 bit parameters for wide range.

The profile generator supports position,velocity, and homing modes. Position mode includes ramp-up, slew, ramp-down motions. Velocity mode supports breakpoints and a linked list parameter loading system for accurate profiling. Profile generator uses 48 bit accumulator to allow velocities down to ~2 turns per day (500 line - 2000 count encoder, 4 KHz sample rate)

There is a separate manual available for the SOFTDMC motion controller.