7I61 MANUAL

V1.2

This page intentionally not blank

Table of Contents

Table of Contents

GENERAL

DESCRIPTION

The 7I61 is a USB/EPP version of the FPGA based Anything I/O card series. It provides 96 programmable I/O bits and a high speed USB interface or EPP interface for real time applications.

Initial FPGA configurations can be downloaded to the 7I61 via the USB or EPP port. The 7I61 also has a serial EEPROM for FPGA configuration storage when the 7I61 is used in stand-alone applications.

The 96 I/O bits are available on four 50 pin connectors, 24 bits per connector. The 50 pin connectors have I/O module rack compatible pin-outs. The connector pin-out uses interleaved grounds for lower crosstalk and controlled impedance. All I/O can be 5V tolerant and all I/O supports LVDS signaling.

/Done, /Init and power status LEDs are provided for debugging purposes as are 8 FPGA driven LEDs. Many I/O interface daughter cards are available for the 7I61. These cards include a 4 axis 3A Hbridge, a 2 Axis 3A stepper motor driver, an analog servo amp. interface, various RS-422/485 interfaces, resolver interfaces and a debug LED card. Two daughter cards can plug directly onto the 7I61.

Many IO configuration files are provided with the 7I61 including simple remote I/O, 4 and 8 axis servo motion control, 4 and 8 axis microstepping stepper motor control, multiple channel PWM generator, quadrature counters and more. VHDL source is provided for all configurations.

FPGA system clock is 50MHZ Oscillator. The Spartan6 used can multiply or divide this frequency to suitable values via on chip PLLs and DLLs for application use.

The 7I61 uses a XC6SLX16 or XC6SLX25 Xilinx Spartan6 FPGA Free development tools for the Spartan6 are available (Xilinx WebPack) from Xilinx's web site.

HARDWARE CONFIGURATION

GENERAL

Hardware setup jumper positions assume that the 7I61 card is oriented in an upright position, that is, with the USB connector towards the person doing the configuration, and the power connector on top right.

FPGA CONFIGURATION SOURCE

The 7I61's FPGA can be configured via the USB port, The EPP port, or the on card serial EEPROM. Jumpers W203 and W204 in the lower left part of the 7I61 card select the configuration source.

W203	W204	MODE
DOWN	DOWN	EPP (PARALLEL PORT) CONFIG
DOWN	UP	USB CONFIG
UP	DOWN	EEPROM CONFIG

USB POWER

The 7I61 can be powered by the USB host. The maximum power that can be supplied by a USB host is 450 mA. This will be sufficient for many but not all 7I61 applications. For applications that require more than the 450 mA supplied by the host, the 7I61 has provisions for external power. W201 (in the lower left part of the 7I61 card) connects host USB power to the 7I61's power supplies. To use host USB power, W201 must be set to the "UP" position. If external 5V power is used, W201 **must** be set to the "DOWN" position.

WARNING: ***Connecting an external 5V supply to the 7I61 while W201 is in the "UP" position and a USB cable connects the 7I61 to a host computer is likely to damage the computer by feeding external power 'backwards' into the USB port!***

POWER ENABLE

The 7I61 can be set to power-up only after the USB interface is activated. This is the suggested operational mode when the 7I61 is interfaced via USB. For applications where the 7I61 must operate without the USB interface, This function must be disabled. W205 controls the power up enable mode. When W205 is in the "UP" position, the 7I61 power supplies are always enabled. When W205 is in the "DOWN" position, the 7I61 power supplies will only be enabled when the USB interface is active.

HARDWARE CONFIGURATION

CONNECTOR AND PULLUP POWER

The power connection on each of the I/O connectors (Pin 49) can supply either 3.3V or 5V power. Supplied power should be limited to 400 mA per connector. When W1,W3,W8,W10 are in the "DOWN" position, 3.3V power is supplied to the associated I/O connector. When W1,W3,W8,W10 are in the "UP" position, 5V power is supplied the associated I/O connector. Note that most Mesa I/O adapter cards that connect to Anything I/O cards require 5V.

BUS SWITCH MODE

The 7I61 has bus switches in series with al FPGA I/O pins for 5V tolerance. Each I/O connector can have the 5V tolerance mode or 3.3V mode. Jumpers W2,W4,W9,W11 determine the bus switch mode for the associated connector. When jumper W2,W4,W9,W11 are in the "UP" position, 5V tolerant mode is selected, when 'down', 3.3V mode is selected. ***Note that 3.3V mode is not 5V tolerant. The FPGA can be damaged by input voltages greater than 4V in 3.3V mode.*** The following table shows which jumper selects 3.3V/5V power and bus switch mode for each I/O connector.

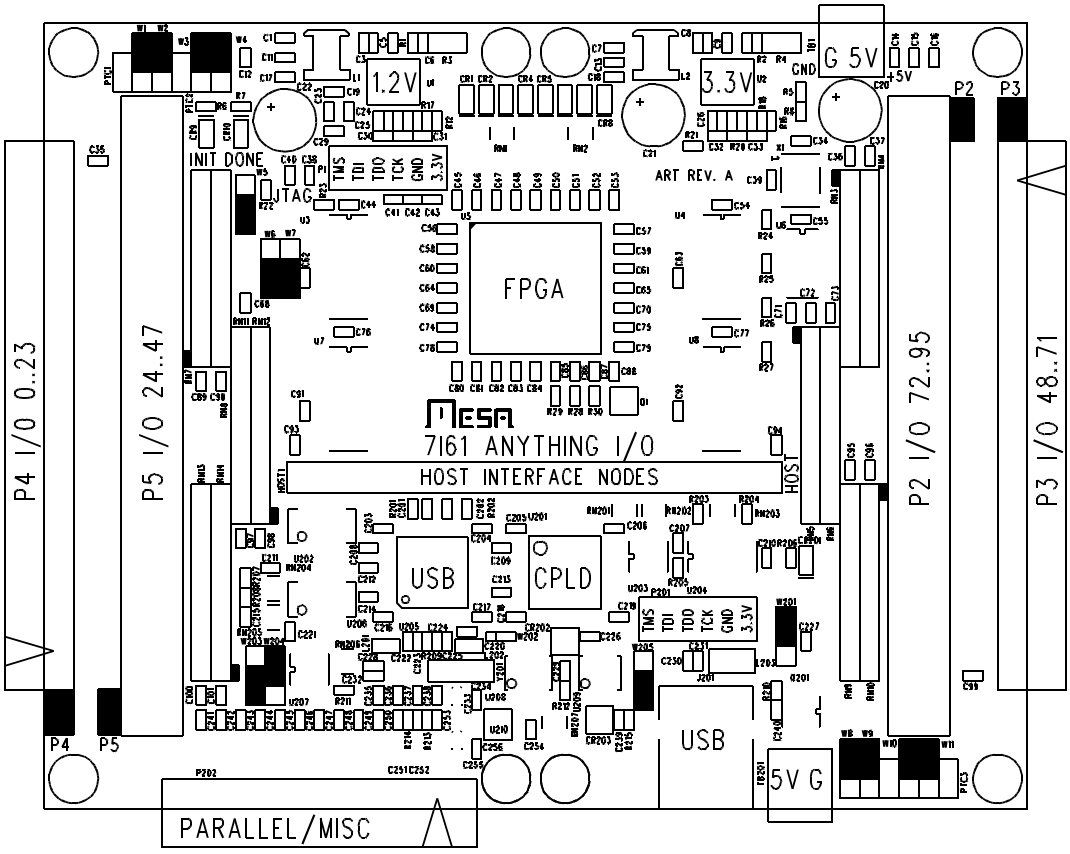
POWER	SWITCH MODE	CONNECTOR
W1	W2	P4
W3	W4	P5
W8	W9	P2
W10	W11	P3

PRE-CONFIGURATION PULL-UPS

The 7I61 has 3.3K pull-up resistors on its user I/O pins. These resistors can be removed in groups of 8 I/O pins if not desired. If the pullup resistor(s) are removed, pins with no pullups will not have a defined state before the FPGA is configured, If this is not desired, internal pull-up resistors on all FPGA pins can be enabled via Jumper W5. When W5 is in the "DOWN" position, user I/O will float until the FPGA is configured. When W5 is in the "UP" position, all FPGA pins including user I/O pins will have a pull-up resistor to 3.3V so the pins will be in a "HIGH" state. It is suggested that the internal pull-ups be enabled unless this causes a problem with connected I/O devices. Note that once the FPGA is configured, each FPGA input pin can have programmable pull-up or pull-down resistors.

CONNECTORS

CONNECTOR LOCATIONS AND DEFAULT JUMPER POSITIONS



CONNECTORS

I/O CONNECTORS

P2,P3,P4 and P5 are the 7I61s 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 7I61IO.PIN file on the 7I61 distribution disk. 7I61 IO connector pinouts are as follows:

P4 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

7I61 I/O CONNECTORS

P5 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

7I61 I/O CONNECTORS

P3 CONNECTOR PINOUT

PIN	FUNC	PIN	FUNC	PIN	FUNC	PIN	FUNC
1	IO48	2	GND	3	IO49	4	GND
5	IO50	6	GND	7	IO51	8	GND
9	IO52	10	GND	11	IO53	12	GND
13	IO54	14	GND	15	IO55	16	GND
17	IO56	18	GND	19	IO57	20	GND
21	IO58	22	GND	23	IO59	24	GND
25	IO60	26	GND	27	IO61	28	GND
29	IO62	30	GND	31	IO63	32	GND
33	IO64	34	GND	35	IO65	36	GND
37	IO66	38	GND	39	IO67	40	GND
41	IO68	42	GND	43	IO69	44	GND
45	IO70	46	GND	47	IO71	48	GND
49	POWER	50	GND				

CONNECTORS

7I61 I/O CONNECTORS

P2 CONNECTOR PINOUT

PIN	FUNC	PIN	FUNC	PIN	FUNC	PIN	FUNC
1	IO72	2	GND	3	IO73	4	GND
5	IO74	6	GND	7	IO75	8	GND
9	IO76	10	GND	11	IO77	12	GND
13	IO78	14	GND	15	IO79	16	GND
17	IO80	18	GND	19	IO81	20	GND
21	IO82	22	GND	23	IO83	24	GND
25	IO84	26	GND	27	IO85	28	GND
29	IO86	30	GND	31	IO87	32	GND
33	IO88	34	GND	35	IO89	36	GND
37	IO90	38	GND	39	IO91	40	GND
41	IO92	42	GND	43	IO93	44	GND
45	IO94	46	GND	47	IO95	48	GND
49	POWER	50	GND				

CONNECTORS

7I61 JTAG CONNECTORS

P1 and P201 are the 7I61s JTAG programming connectors. These are not normally used since the 7I61 can be programmed via the USB or EPP interface, but can be useful when debugging or reprogramming the CPLD or debugging FPGA code with Xilinx tools. 3.3V levels are used for JTAG signals. P1 is the FPGA JTAG connector and P201 is the CPLD JTAG connector

P1,P201 CONNECTOR PINOUT

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

POWER CONNECTORS

The 7I61 has two external 5V power connectors, TB1 and TB201. These connectors supply power to the 7I61 in EPP, standalone, and USB applications where USB host power is not sufficient to power the 7I61. TB1 supplies 5V un-switched 5V power to the 7I61 while TB201 supplies switched 5V power. Power switching is determined by USB enumeration or jumper W205. If jumper W205 is in the "DOWN" position TB201 will only supply power to the 7I61 after the USB has enumerated. If W205 is in the "UP" position, the power switch on TB201 is always on. TB1 may be used as a switched 5V power source for external devices. Maximum switched current is 5A, this includes any 7I61 current.

TB1,TB201 CONNECTOR PINOUT

PIN	FUNCTION	NOTE
1	+5V	Square pad on bottom
2	G	

CONNECTORS

EPP INTERFACE CONNECTOR

P2 is the EPP printer port interface connector. P2 is a 26 pin header. P2 pin-out matches stands DB25 printer port pin-out, allowing a simple flat cable with a DB25M IDC connector on one end and a 26 pin female header on the other end to interface the hosts printer port to the 7I61.

P2 PIN	DB25 PIN	SIGNAL	P2 PIN	DB25 PIN	SIGNAL
1	1	/STROBE	2	14	/AUTOFD
3	2	PD0	4	15	/FAULT
5	3	PD1	6	16	/INIT
7	4	PD2	8	17	/SELECTIN
9	5	PD3	10	18	GND
11	6	PD4	12	19	GND
13	7	PD5	14	20	GND
15	8	PD6	16	21	GND
17	9	PD7	18	22	GND
19	10	/ACK	20	23	GND
21	11	BUSY	22	24	GND
23	12	PERROR	24	25	GND
25	13	SELECT	26	VCC	

OPERATION

FPGA

The 7I61 uses a Xilinx Spartan6 FPGA in a 256 ball BGA package, Either PN XC6SLX16-2 or XC6SLX25-2 depending on 7I61 model.

HOST INTERFACE

The 7I61 uses either a USB or EPP printer port interface to the host. These interfaces can be used for programming the FPGA and accessing the FPGA once programmed.

EPP CONFIGURATION

When the7I61 is jumpered so the configuration source is EPP, *and* the FPGA is not configured (DONE is low), the on card CPLD implements two EPP registers to allow configuring the FPGA via the EPP port.

The two EPP registers are the control register and the data register. The control register is at EPP address 1 and has a single output bit (at D0) that controls FPGA /PROGRAM, and a single input bit (at D0) that reads the FPGA's done status. The data register at EPP address 0, is used for the byte wide configuration data. Reads from the data register will return the FPGA size in D0, 1 = 25 and 0 = 16.

EPP CONFIGURATION PROCEDURE

```
EPPWriteAddress(1)          ; Select EPP address 0x01 = control register

EPPWriteData(0)             ; Set /PROGRAM low

EPPWriteData(1)             ; Set /PROGRAM High

(Wait 10 mS)                ; Wait 10mS for FPGA to initialize

EPPWriteAddress(0)          ; Select EPP address 0x00 = data register

EPPReadData                 ; Verify FPGA size

EPPWriteData(FPGAByte0)     ; Write first byte of FPGA config data

EPPWriteData(FPGAByte1)     ; Write second byte of FPGA config data

(write remaining FPGA config bytes)
```

OPERATION

EPP CONFIGURATION

Once the FPGA is configured, the CPLD EPP registers and EPP handshake logic are disabled and it is the FPGA's responsibility to handle the EPP host interface. The CPLD is still used at this point to forward some of the EPP port handshaking lines through to the FPGA for 5V tolerance.

USB CONFIGURATION

When the7I61 is jumpered so the configuration source is USB, and the FPGA is not configured (DONE is low), the on card CPLD implements a simple data handshake so all data sent to the USB port is written to the FPGAs configuration port.

The CPLD also will echo characters to indicate the FPGA size and DONE status. If a character is sent to the 7I61 and the characters LSb is '1' the DONE status will be returned in the echoed characters LSb. If a character with a '0' LSb is sent, a character will be echoed indicating the FPGA size. This echoed character will have a '0' LSb for SLX16 7I61s and a '1' LSb for SLX25 7I61s. Since it in not desirable to deal with echoed characters for every configuration byte sent to the 7I61, status character echoing is disabled after receiving 4 consecutive characters with a '0' LSB.. Once the FPGA is configured the CPLD data handshake is disabled and it is the FPGA's responsibility to handle the interface to the USB chip.

USB CONFIGURATION PROCEDURE

Flush receive buffer	; Optional
Send "1" character	; Optional
Check echoed character for LSb = 0 (done should be low	; Optional
Send "0" character	; Optional
Check echoed character LSb to determine FPGA size	; Optional
Send 4 more "0" characters	; Disable echo
Delay 10 mS	; wait for FPGA reset
Send configuration byte 0	
Send configuration byte 1	
(Send remaining configuration bytes)	

OPERATION

EEPROM CONFIGURATION

For stand-alone applications and when it is not desired to have to preconfigure the FPGA via the host interface at power up, the 7I61 can be configured via its serial EEPROM. Of course the Serial EEPROM must first be programmed with the desired configuration file. The serial EEPROM used is a ST M25P80 SPI flash serial EEPROM.

All access the serial EEPROM is via the FPGA, so programming the serial EEPROM is a "bootstrap" process, where the first step is programming the FPGA with a configuration giving host (EPP or USB) access to the serial EEPROM through the FPGA. Both the EPP and USB GPIO demo configurations allow this EEPROM access via a simple SPI interface built into the configuration.

The SCM7I61P program is an example program for writing the serial EEPROM via the EPP port (DOS only), SCM7I61W is a similar example program for writing the serial EEPROM via the USB port (windows only) The SCM programs rely on EPPIOPR8 (for EPP programming or USBIOPR8 (for USB) configuration file being preloaded into the FPGA before writing the serial EEPROM, as the serial EEPROM can only be accessed through the FPGA. EPPIOPR8 and USBIOPR8 have a simple SPI interface to allow EEPROM access.

EXTRA EEPROM SPACE

The serial configuration EEPROM on the 7I61 has a capacity of 1M bytes, but the configuration bit file for the XC6SLX16 Spartan chip is only ~464K bytes, leaving 512 K bytes free for FPGA accessible non volatile storage. The XC6SLX25 Spartan chip uses about 800K bytes of the serial EEPROM leaving 192K bytes free. This storage can be used for non-volatile settings or program storage in stand-alone 7I61 applications. Note that the free space has been rounded down to multiples of 64K bytes because the EEPROMs are only erasable in increments of 64K bytes.

OPERATION

RECONFIGURATION

Once the 7I61 is configured, the CPLD loader is disabled. In order to reconfigure the FPGA, the FPGA must be reset via /PROGRAM. This can be done by having the FPGA assert its /RECONFIG pin (drive it low). If you wish to have the ability to reconfigure the FPGA without cycling the power, the FPGA configuration must include some way of asserting /RECONFIG. /RECONFIG is FPGA ball M10.

CONFIGURATION FILE STARTUP OPTIONS

Important: *Because the 7I61s CPLD stops configuration when DONE is asserted, the configuration file startup options must be set so that asserting DONE is the last configuration step.* Suggested startup options are as follows:

FPGA STARTUP CLOCK: **CCLK**

DONE: **6**

ENABLE OUTPUTS: **6**

RELEASE WRITE ENABLE: **4**

RELEASE DLL: **NO WAIT**

SC7I61P and SC7I61W

Two utility programs, SC7I61P.EXE and SC7I61W are provided to send configuration files to the 7I61. SC7I61P is a DOS only program and SC7I61W is a windows only program.

SC7I61P is invoked with the FPGA configuration file and the Hexadecimal EPP port base address on the command line:

```
SC7I61P FPGAFILE.BIT 378
```

SC7I61W is invoked with the FPGA configuration file and the COM port on the command line:

```
SC7I61W FPGAFILE.BIT COM6
```

SC7I61P and SC7I61W use binary FPGA configuration files. These files can standard Xilinx BIT files or Xilinx PROM format files.

OPERATION

CLOCK SIGNALS

The 7I61 has a 50 MHz crystal controlled clock signal routed to ball M7 on the FPGA. A total of twenty user I/O pins are also GCLK pins. These can be used for single ended clocks or LVDS pairs.

The first two I/O pins of each I/O connectors are available as clocks:

IO BIT	NAME	FPGA BALL	IO BIT	NAME	FPGA BALL
IOBIT0	C24	J4	IOBIT48	C14	E8
IOBIT1	C25	K3	IOBIT49	C15	E7
IOBIT24	C22	H5	IOBIT72	C4	J16
IOBIT25	C23	J6	IOBIT73	C5	J14

Additional pins that can be used as clocks:

IOBIT26	C20	H3	IOBIT56	C18	A9
IOBIT27	C21	H4	IOBIT57	C19	C9
IOBIT30	C26	F1	IOBIT58	C16	A10
IOBIT31	C27	F2	IOBIT59	C17	B10
IOBIT60	C12	C10	IOBIT94	C8	K14
IOBIT61	C13	E10	IOBIT95	C9	J13

OPERATION

EPP-FPGA INTERFACE

The interface from host EPP printer port to the FPGA uses 12 FPGA pins. These consist of an eight bit bidirectional data bus (D0..D7), and four handshake lines. The D bus connects to the FPGA through bus switches for 5V tolerance. The D BUS is shared with the USB data bus so the EPP and USB interfaces cannot operate simultaneously.

P202 PIN	EPPNAME	SPPNAME	FPGA BALL	DIRECTION
1	/WRITE	/STROBE	T4	TO FPGA
2	/DSTROBE	/AUTOFD	P4	TO FPGA
8	/ASTROBE	/SELECTIN	N6	TO FPGA
21	WAIT	BUSY	M6	FROM FPGA
3	D0	D0	P7	BIDIR
5	D1	D1	N12	BIDIR
7	D2	D2	P12	BIDIR
9	D3	D3	N5	BIDIR
11	D4	D4	P5	BIDIR
13	D5	D5	L8	BIDIR
15	D6	D6	L7	BIDIR
17	D7	D7	R5	BIDIR

Signals unused in EPP mode but routed to FPGA:

19	XXX	/ACK	T6	XXX
6	XXX	/PINIT	P6	XXX
23	XXX	PERROR	T7	XXX
25	XXX	SELECT	R7	XXX

OPERATION

EPP-FPGA INTERFACE

The EPP interface implements a simple multiplexed 8 bit data/address bus. The EPPIOPR8 configuration can be used as an example of an EPP interface in the FPGA. This is a simple GPIO interface organized as six eight bit ports.

USB-FPGA INTERFACE

The 7I61 uses a FT2232H high speed USB interface chip (480 Mbps). The FT2232H appears as two serial ports. Only the first port is used by the supplied 7I61 firmware. The supplied configurations support the asynchronous FIFO mode on one port (Port A) The 7I61 has all the connections to support all FT2232 modes including support for the high speed synchronous mode that allows host transfer rate of 25 Mbytes/second and dual(A and B) FIFO mode. The default mode is limited to a 10 M bytes per second transfer rate.

The FPGA interface uses a bidirectional 8 bit data bus that is shared with the EPP interface on the 7I61). Because of this sharing you cannot operate the USB and EPP interfaces simultaneously.

SIGNAL NAME	FPGA BALL	DIRECTION	FUNCTION
USBWRITE	N14	FROM FPGA	XMIT DATA STROBE
/USBRD	N16	FROM FPGA	RECV DATA STROBE
/USBTXE	P15	TO FPGA	XMIT FIFO NOT FULL
/USBRXF	P16	TO FPGA	RECV FIFO HAS DATA
D0	P7	BIDIR	DATA BUS
D1	N12	BIDIR	DATA BUS
D2	P12	BIDIR	DATA BUS
D3	N5	BIDIR	DATA BUS
D4	P5	BIDIR	DATA BUS
D5	L8	BIDIR	DATA BUS
D6	L7	BIDIR	DATA BUS
D7	R5	BIDIR	DATA BUS

OPERATION

ADDITIONAL USB INTERFACE CONNECTIONS

The 7I61 FPGA connects to all of the FT2232H's I/O pins so can support FT2232H's high speed synchronous FIFO interface mode, dual asynchronous FIFO modes, SPI mode etc. Details on these other connections can be found in the 7I61u.ucf file in the hostmot2 source firmware directory.

LEDS

The 7I61 has 8 FPGA driven user LEDS. These green LEDS are located in the top center of the card. They 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 7I61MISC.PIN file for FPGA pin locations of the LED signals.

In addition to the user LEDs there are three other LEDS that display board status information. The LEDS are a red /DONE LED and a red /INIT LED, and a Yellow OK LED.The /DONE and /INIT LED can be used to determine FPGA configuration status. The /INIT LED will be illuminated when /PROGRAM is asserted or when a CRC error has occurred during the configuration process. The /DONE LED will be illuminated when the FPGA is not configured. The Yellow OK led illuminates when power is OK and the FPGA is configured. The two red FPGA LEDS are located in the top left corner of the 7I61 FPGA card. The OK led is located in the bottom right corner of the 7I61 card.

OPERATION

BUS SWITCH MODE

The 7I61 uses bus switch devices in series with all I/O pins. These devices allow the 7I61 inputs to be 5V tolerant and allow the I/O pins to be pulled up to 5V. The bus switch input protection function works by disconnecting the FPGA from the IO pins when the IO pin voltage rises above a preset threshold. This threshold determines the bus switch operational mode. We refer to the modes as 5V tolerant mode and 3.3V mode. Each I/O connector has its own bus switch device, which means that the bus switch mode can be selected on a per connector basis.

When in 5V tolerant mode, the inputs and tri-stated outputs may be pulled up to 5V. This allows driving 5V referred loads such as I/O module racks and connecting inputs to 5V logic. The disadvantage of 5V mode is that the output impedance is higher in the high output state (when the FPGA pins are at 3.3V) as the bus switch is off when the FPGA pin is at 3.3V.

When 3.3V mode is selected, the bus switch is always fully on unless input voltages >4V are applied, at which point the bus switch disconnects the FPGA from the I/O pin. 3.3V mode is suggested for general use. ***Note that 3.3V mode is not 5V tolerant***. When the bus switch mode jumper W2 is in the 'up' position, 5V mode is selected, when 'down', 3.3V bus switch mode is selected.

IO LEVELS

The FPGA used on the 7I61 is a Spartan6. The Spartan6 supports many I/O standards. The 7I61 does not support use of the I/O standards that require input reference voltages, also VCCIO is fixed at 3.3V. The available I/O options for are LVTTL, LVCMOS_33, LVDCI_33, and LVDCI_33_DV2 and LVDS_3.3.

The Spartan6 FPGA chip used on the 7I61 is not 5V tolerant but external bus switch parts are used on the 7I61 to make the I/O pins 5V tolerant. The bus switch parts disconnect the FPGA pins from the I/O pins when the I/O pins are driven to positive voltage levels that would damage the FPGA.

The voltage level that causes disconnect can be selected to be ~4V (3.3V mode) or ~3.3V (5Vmode). For most applications, the 3.3V mode should be used. The 5V tolerant mode is useful when driving 5V referred loads or accepting 5V logic levels.

Note that there is no protection against negative input voltages other than the input clamp diodes in the FPGA and bus switches, so negative input voltages must be limited to -.5V

OPERATION

PULLUP RESISTORS

All I/O pins are provided with pullup resistors to allow connection to open drain, open collector, or OPTO devices. These resistors are 10 pin, 9 resistor SIP networks. The resistor networks are socketed to allow the user to select different values. Pin 10 of all the resistor networks is grounded, allowing the use of 220/330 Ohm termination networks if desired on the receiving end of a high speed bus.

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 results in a safe condition.

DRIVING +5V REFERRED LOADS

When driving external loads like Solid State Relays (SSRs) with an active low output, and the +SSR terminal connected to +5V, the 7I61 output should be configured for 5V tolerance, and the output should be driven in open drain mode. This is because the 7I61 outputs only swing to 3.3V in normal mode, leaving 1.7V (5V -3.3V) driving the SSR when the output is high and the SSR should be off.

TERMINATION

The FPGA used on the 7I61 supports series and parallel termination that can be programmed on a pin-for-pin basis.

SUPPLIED CONFIGURATIONS

EPPIOPR8

GENERAL

The EPPIOPR8 configuration is a simple six port GPIO configuration with EPP interface. The GPIO is organized as six eight bit ports, each with an associated Data Direction Register (DDR).The EPPIOPR8 configuration can be used as a starting point for more complicated user configurations. There are two EPPIOPR8 configuration files, EPPIO8-S.BIT for the XC16SLC16 version and EPPIO8-B.BIT for XC6SLX25 versions of the 7I61.

PORT	DATA REG	DDR	IO BITS	CONNECTOR
0	0x10	0x20	0..7	P4
1	0x11	0x21	8..15	P4
2	0x12	0x22	16..23	P4
3	0x13	0x23	24..31	P5
4	0x14	0x24	32..39	P5
5	0x15	0x25	40..47	P5
6	0x16	0x26	48..55	P3
7	0x17	0x27	56..63	P3
8	0x18	0x28	64..71	P3
9	0x19	0x29	72..79	P2
10	0x1A	0x2A	80..87	P2
11	0x1B	0x2B	88..95	P2

OPERATION

EPPIOPR8

In addition to the GPIO bits, the EPPIOPR8 configuration has a simple SPI interface to the configuration EEPROM and a reconfiguration port.. The SPI port allows the utility program SCM7I61P to write configuration data to the serial EEPROM. These registers are mapped as follows:

REGISTER	ADDRESS	FUNCTION
SPICS	0x7D	Single I/O bit to control SPI Chip Select (bit 0)
SPIDATA	0x7E	Eight bit SPI shift register
RECONFIG	0x7F	Reconfig - Writing 0x5A here resets FPGA

The 7I61 is delivered with the EPPIOPR8 configuration installed for factory and initial user checking.

SUPPLIED CONFIGURATIONS

USBIOPR8

GENERAL

The USBIOPR8 configuration is a simple six port GPIO configuration almost identical to the EPPIOPR configuration. It is different because the USB interface is a simple bidirectional byte-stream without separate address and data. Because of this a Little Binary Protocol (LBP) is used to communicate with standard addressable peripherals in the 7I61 FPGA configuration. The GPIO is organized as six eight bit ports, each with an associated Data Direction Register (DDR).The USBIOPR8 configuration can be used as a starting point for more complicated user configurations. There are two USBIOPR8 configuration files, USBIO8-S.BIT for the XC16SLC16 and USBIO8-B.BIT for XC16SLC25 versions of the 7I61.

PORT	DATA REG	DDR	IO BITS	CONNECTOR
0	0x10	0x20	0..7	P4
1	0x11	0x21	8..15	P4
2	0x12	0x22	16..23	P4
3	0x13	0x23	24..31	P5
4	0x14	0x24	32..39	P5
5	0x15	0x25	40..47	P5
6	0x16	0x26	48..55	P3
7	0x17	0x27	56..63	P3
8	0x18	0x28	64..71	P3
9	0x19	0x29	72..79	P2
10	0x1A	0x2A	80..87	P2
11	0x1B	0x2B	88..95	P2

In addition to the GPIO bits, the USBIOPR8 configuration has a simple SPI interface to the configuration EEPROM, a LED port, and a reconfiguration port.. The SPI port allows the utility program SCM7I61W to write configuration data to the serial EEPROM. These registers are mapped as follows:

SUPPLIED CONFIGURATIONS

USBIOPR8

REGISTER	ADDRESS	FUNCTION
LED	0x07A	8 Status LEDS (‘1' = on)
SPICS	0x07D	Single I/O bit to control SPI Chip Select (bit 0)
SPIDATA	0x07E	Eight bit SPI shift register
RECONFIG	0x07F	Reconfig - Writing 0x5A here resets FPGA

SUPPLIED CONFIGURATIONS

USBIOPR8

LBP

LBP is a simple master slave protocol where the host sends read, write, or RPC commands to the 7I61, and the 7I61 responds. LBP allows the host (master) to efficiently access registers on the slave (7I61) via a simple bidirectional byte oriented protocol.

LBP commands always start with a command header byte. This header specifies whether the command is a read or a write, 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.

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.

SUPPLIED CONFIGURATIONS

USBIOPR8

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

SUPPLIED CONFIGURATIONS

USBIOPR8

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 (dont-care for USB devices)

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

0xC3 Get CRC error count

0xC4 .. 0xC9 Reserved

0xCA Get Enable_RPCMEM access flag

0xCB Get Command timeout (in mS for USB and character times/10 for serial)

0xCC Get Non-volatile memory flag

0xCD Get External memory flag

0xCE.. 0xCF Reserved

SUPPLIED CONFIGURATIONS

USBIOPR8

LOCAL LBP READ COMMANDS

0xD0 .. 0xD3 4 character card name

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)

SUPPLIED CONFIGURATIONS

USBIOPR8

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 Set Non-volatile memory flag

0xED Set External memory flag (non zero for external memory mode)

0xEE .. 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)

SUPPLIED CONFIGURATIONS

USBIOPR8

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 USB serial data stream. RPCs allow significant command compression which improves communication bandwidth.

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 USBIOPR8 configuration, RPCPitch is 0x10 bytes so each RPC command has native size of 0x10 bytes and start 0x10 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 0x10 bytes, RPC0 starts at 0x0000, RPC1 starts at 0x0010, RPC2 starts at 0x0020 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.

SUPPLIED CONFIGURATIONS

USBIOPR8

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 3 LBP commands and uses 11 bytes starting at 0x0050 (RPC5 for 0x10 pitch RPC table)

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

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

Command3. Writes a single byte (0xAA) to port 0x14, data contained in RPC table

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
LBPWrite 2 add 1 data	1	0	1	1	0	1	0	0
Write Address LSB	0	0	0	1	0	1	0	0
Write Address MSB	0	0	0	0	0	0	0	0
Write Data	1	0	1	0	1	0	1	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

AVAILABLE DAUGHTER CARDS

PART NUMBER	FUNCTION
7I29	Dual 20A 165V HBridge
7I30	Quad 3A 36V HBridge
7I31	Debug LED card
7I32	Dual stepper driver (microstepping)
7I33,7I33T	Quad analog servo interface
7I34	8 TX + 8 RX pair RS-422 interface
7I34-R	16 RX pair RS-422 interface
7I37,7I37T	8 output 16 input isolated I/O
7I39	Dual 3 phase H-Bridge
7I40	Dual H-Bridge
7I42	I/O protector
7I44	8 Channel RS-422 to RJ45 breakout
7I46	6 channel SPI breakout
7I47	12 channel encoder oriented RS-422 interface
7I47S	12 channel encoder interface with isolated spindle analog out
7I48	6 channel analog servo interface
7I49	6 channel resolver interface
7I50	SPI I/O expander
7I64	24 input, 24 output isolated I/O
7I65	Octal16 bit A-D analog servo interface
7I66-8	16 input 8 output isolated I/O
7I66-24	24 output isolated I/O

REFERENCE INFORMATION

SPECIFICATIONS

	MIN	MAX	NOTES
3.3V CURRENT TO P2,P3,P4,P5	----	2A	Typically limited to ~200 mA in USB powered case.
5V CURRENTTO P3,P4	----	2A	Typically limited to ~150 mA in USB powered case.
1.2V CORE POWER CURRENT	----	3A	1A = ~300 mA of 5V draw. Depends on FPGA configuration
MAXIMUM I/O I SINK OR I SOURCE	----	24mA	
MAXIMUM I/O INPUT VOLTAGE	-.5V	7V	5V Tolerant mode. I/O 0..95, EPP I/O
MAXIMUM I/O INPUT VOLTAGE	-.5V	4V	3.3V mode. I/O 0..95
TEMPERATURE -C VERSION	0°C	70°C	
TEMPERATURE -I VERSION	-40°C	85°C	