ROWE ENGINEERING

ALTERA FPGA DEVELOPMENT BOARD
Q5V1

USER MANUAL VER. 1.1

Oct. 2002

Introduction

Thank you for your purchase of the Q5V1 Altera FPGA Development Board. ***PLEASE READ THIS MANUAL WHICH RESIDES ON THE CD-ROM BEFORE ATTEMPTING TO USE THE BOARD***. This manual will guide you through the usage and settings to make your board function in a minimum of time. The manual is set up in 3 sections. The first section describes the hardware and the interconnections between the 1 chip and the external world. Section #2 describes a quick and easy way to get started using your new Q5V1 board. Please make sure to read this section before attempting down-loads. Section 2 also describes the use of Quartus[1] to down-load directly into the Altera's as well as the Flash Configuration Device on the board. Finally, section 3 discusses other board attributes and a method of using them.

Included with your purchase is a pin configuration file to be inserted into the .csf file necessary for Quartus[1] to configure inputs & outputs for the 1 chip as well as a simple set of diagnostics, written in VHDL, which will allow you to verify proper board operation. This pin configuration file specifies all of the connections possible between the (1) Altera chip regardless of which are being used.

Questions can also be addressed to our technical support department at technicalsupport@roweengineering.net.

Section 1 – Hardware Description

The Q5V1 board has (1) Altera EP1S25F780C5ES FPGA. A block diagram of the board is shown in Figure 1. A board layout is presented in Figure 3. Zooming on this figure shows pin number definitions for all connectors.

1.1 Connectors & I/O Ports

The FPGA has six ports. The first port is labeled LVTTL I/O. There are two instances of it on figure 1. In the sample code provided, it is called "ioport" and "ioport1." Each of these ports are 40 pin I/O ribbon connectors (AMP Part# 102153-9). Each connector supplies 20 I/O bits on the odd numbered connector pins (with pin39 being the LSB) and 20 ground signals on the even numbered connector pins. The user must be careful when connecting ribbon cables on these ports as the grounds are hard-wired within the printed circuit board. The grounds provide for minimal cross-talk between the active signals. The connectors are H3 and H5. Ioport1 feeds H5 and ioport feeds H3.

The second port is labeled "test_out." This is MICTOR connector H4 (AMP Part 2-767004-2). 32 pins carry active signals for the FPGA which the user routes to these pins within their specific designs. As an example, these ports can be used for logic analyzers or generic I/O. The test_out port defined in the pin configuration files and sample code setup a standard MICTOR connector as shown in Figure 2. On figure 2, D0O-D15O represent the 16 bits sent to the odd numbered pod. Similarly, D0E-D15E represent the 16 bits sent to the even numbered pod. The .csf file included with the q5v1 board assigns test_out(0:15) to D0O-D15O and test_out(16:31) to D0E-D15E. Furthermore, there are two clocking possibilities, CLKM1 and CLKM2. CLKM1 is

derived directly from the crystal oscillator (U3) or function generator (J2) sources depending on which is active. This signal is presented to the clk input of the even numbered pod. The CLKM2 signal is derived from the ROBOCLOCK bank 2, clock 2Q5, and has a programmable skew control. The CLKM2 signal is presented to the clk input of the odd numbered pod. For more information on control of this signal, see section 3.1.

The third port is labeled "DIP Switch In." These ports are labeled S1 and S2 and are input only ports. This port was mainly meant as a control port wherein the FPGA could take in 16 bits which the user would set by toggling the DIP switch to VCC or ground. In this way, the user could define certain conditions within their design dependent on constant binary selection. In the sample code provided, S2 will provide switch_in(1:8) while S1 will provide switch_in(9:16) using S1 positions 1-8. It is important to note that placing a switch in the "open" position sends a logic high signal to its' respective input. Conversely, placing a switch in the "closed" position sends a logic low signal to its' respective input.

The fourth port for the FPGA is labeled "led_out" in the sample code provided. The FPGA device has a 4 bit LED port for user defined use. These ports are output only. The lsb of this port is the led that is located closest to the board edge (D4).

The fifth port for the FPGA is labeled "lvds_rx" in the sample code provided. It is connector H1 (3M Part# 10250-1210VE). This port may be used for receiver LVDS functions only. The signal map for the 16 differential inputs and 1 differential clock input is shown in table 1. Naturally, users may ground inputs on their cable if they do not desire to use all 16 inputs. The sample code on the CD in directory q5_1 shows a method for accessing the data within the FPGA. This code was used to test the board at the factory. It is only a sample. The user should feel free to program the lvds receiver port any way that is convenient for them. The 50 pin connector used is manufactured by 3M and is part number 10250-1210VE. If the user does not wish to utilize the lvds interface, this connector can supply the user with 32 general lvttl i/o pins. The basic idea is to assign a single ended lvttl signal to each of the differential signals used for the lvds. The sample code on the CD in directory q5_2 shows sample code with a corresponding pin file in the configuration directory.

The sixth port for the FPGA is labeled "lvds_tx" in the sample code provided. It is connector H2 (3M Part# 10250-1210VE). This port may be used for transmitter LVDS functions only. The signal map is again presented in table 1. Thus, since the same signal definition is used, the LVDS cable can easily tie a transmit end directly to a receiver end. An example is presented in the sample code (q5_1), where it is assumed that the transmit LVDS connector is cabled directly to the receiver LVDS connector on the same board. The diagnostic then checks the rx data against the tx data in a loopback mode to verify LVDS operation. Once again, if the user does not wish to utilize the lvds interface, this connector can supply the user with 34 general lvttl i/o pins (2 more than the rx connector). The sample code on the CD in directory q5_2 shows sample code with a corresponding pin file in the configuration directory.

LVDS Connector Pin	Signal
1	GND
26	GND
2	Clk+
27	Clk-
3	GND
28	D0+
4	D0-
29	D1+
5	D1-
30	GND
6	D2+
31	D2-
7	D3+
32	D3-
8	GND
33	D4-
9	D4+
34	GND
10	D5+
35	D5-
11	GND
36	D6+
12	D6-
37	GND
13	D7+
38	D7-
14	GND
39	D8-
15	D8+
40	GND
16	D9+
41	D9-
17	GND
42	D10-
18	D10+
43	GND
19	D11+
44	D11-
20	D12+
45	D12-
21	GND
46	D13+
22	D13-
47	D14+

23	D14-
48	GND
24	D15+
49	D15-
25	GND
50	GND

Table 1. RX/TX LVDS Connector pin definitions

Besides the FPGA the Q5 board has banana jack power connectors U6 (Johnson Components Inc. Part # 108-0740-001), which must be connected to the 3.3v power supply and U7 (Johnson Components Inc. Part # 108-0740-001), which must be connected to the power supply ground. The SW1 momentary contact switch is connected to the DEV_CLRn signal. If the Quartus[1] user configures the "Enable device-wide reset(Dev_CLRn)" option under the "Compiler Settings…", "Chips & Devices", "Device and Pin Options", then this switch will clear the entire FPGA when depressed.

1.2 Clocking Sources

The SMA connector J2 is for an external clock input while the J1 SMA connector feeds the reference clock for other boards if desired. The U3 socket is meant for an on-board crystal oscillator if external clocking is not used. ***The user should not allow the case where both crystal oscillator & external clock source drive the board simultaneously.***

There are four possible clock sources for the FPGA. The first comes directly from the crystal or J2 connector. It is applied to the FPGA on CLK14, pin K17. It is called clk_xtal_J2 in the sample code provided. The second comes from bank 1, 1Q0, of the Roboclock[2] chip. It is applied to CLK4 on pin AC17. It is called clk_1q0 in the sample code provided. The third clock comes from bank 2, 2Q1, of the Roboclock[2] chip. It is applied to CLK0 on pin P27. It is called clk_2q1 in the sample code provided. The frequency and phase control of the second and third clock is discussed in section 3.1. The fourth and final clock source is the LVDS differential clock signal which accompanies the receiver data. It is applied to CLK11p/CLK11n on pins P2/N2. It is called lvds_rxclk in the sample code provided. A divided down version of this clock could be used to run the rest of the STRATIX chip, not just the LVDS receiver portion. The sample code supplied shows the existence of the four clocks (all connected within the assignment organizer), but only the use of CLK14 (clk_xtal_J2) to run PLL5 which then supplies a differential clock to PLL1 to drive the LVDS transmit clock. CLK11p/CLK11n are supplied to PLL4 (lvds_rxclk) to be used by the receiver. In order to use the other 2 clock sources, the user simply must make reference to them in a process.

The FPGA has it's own 1.5V, 3A regulator to supply VCCINT. H6 and H7 are used to down-load directly into the FPGA or the Flash Configuration Device. Their use is discussed in section 2.

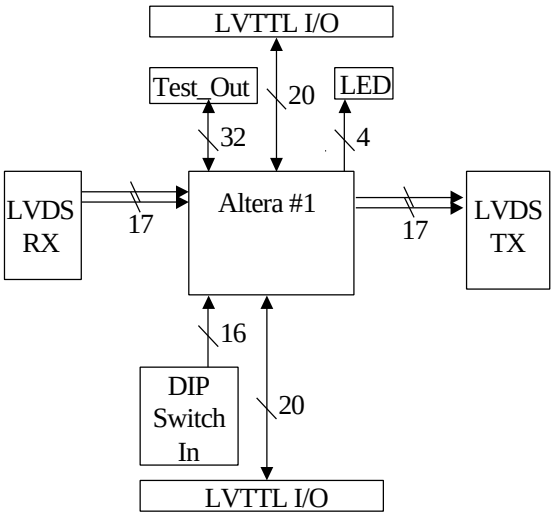
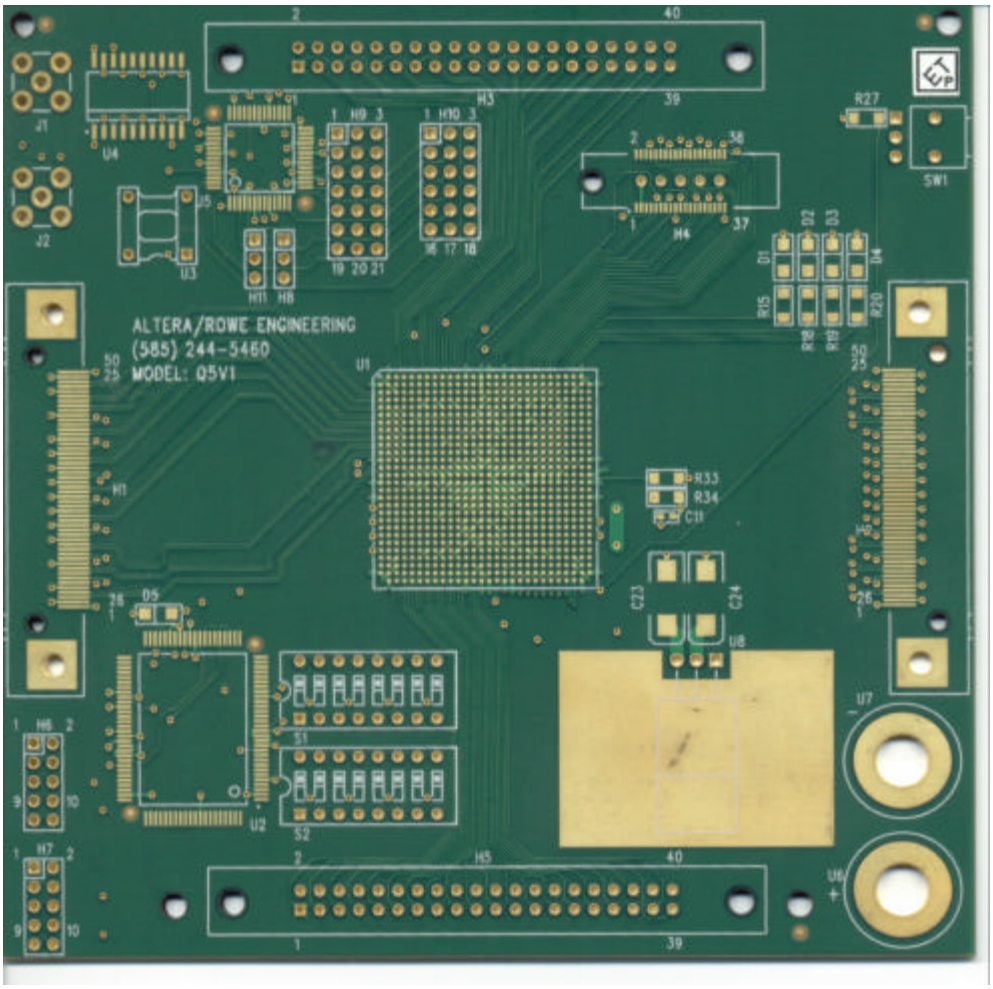


Figure 1. Block Diagram of the Q5V1

NC	1	2	NC
GND	3	4	NC
CLKM1	5	6	CLKM2
D15E	7	8	D15O
D14E	9	10	D14O
D13E	11	12	D13O
D12E	13	14	D12O
D11E	15	16	D11O
D10E	17	18	D10O
D9E	19	20	D9O
D8E	21	22	D8O
D7E	23	24	D7O
D6E	25	26	D6O
D5E	27	28	D5O
D4E	29	30	D4O
D3E	31	32	D3O
D2E	33	34	D2O
D1E	35	36	D1O
D0E	37	38	D0O

Figure 2 MICTOR Connector Assignments

Figure 3



Section 2 – Getting Started

This section will take the new user through a series of steps which will allow the user to down-load into the Altera device. Once you have unpacked your new Q5 board and physically mounted it as desired:

1) Apply (4) mounting supports to the (4) corner holes.
2) Connect banana plug receptacle U6 to the + terminal of a 3.3v power supply. Similarly, connect U7 to the power supply ground.
3) If an external clocking arrangement is to be used, connect the function generator to the J2 SMA connector. For now, we will leave all jumpers H9 and H10 open. These will be defined in section 3 for use with the Cypress Roboclock[2]. Set the function generator to around 20 MHz with a 0 to 3.3V square wave. If a crystal is to be used, do not connect anything to J2 and for this test use a crystal around 20 MHz with a 3.3V output. The Crystal Oscillator U3, has a socket for connection. Pin 1 of U3 is located directly adjacent to the U3 label, which is on the silk screen of the printed circuit board (see Figure 2).
4) The board gives priority to the EPROM's if no download is attempted via the ByteBlasterMV[1]. Further, the EPROMS have been programmed at the factory to run a diagnostic code. Make sure that all the DIP switches are in the closed position. Simply turn on power (3.3v) and make sure that the clock source is connected. The user will see all the LED's flashing a counter. Congratulations, your board is functional!
5) The next step is to download directly into the Altera chip. Connect the ByteBlasterMV[1] from the computers parallel port to H6. Make certain to align pin 1 of the ByteBlasterMV[1] cable to pin 1 of H6, which is marked on the Q5 board.

You are now ready to down-load into the Altera chip. Using Quartus[1], Open a programmer window under the Processing tab. Select "JTAG" for the mode and "ByteBlasterMV" for the programming hardware. If you are running WindowsNT, you must install the ByteBlaster device driver. See the *Quartus Installation and Licensing for PCs* manual available from Altera. Click on the "Add File" button and include the Q5_1.sof file provided to you with the Q5 board in the sof directory on the CD. Your programmer window should look similar to the following table:

File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank-Check	Examine	Security Bit
Q5_1.sof	EP1S25F780	00B85B72	FFFFFFFF	√				

Once you have set up the programmer window, hit the "Start" button & once the down-load is complete, notice a counter flashing the LED ports of the Altera device if all the DIP switches are in the closed position. The counter is also sent to the test_out port. The user can connect a MICTOR adapter (Agilent Part E5346A (not included with the q5)) and select either CLKM1 or CLKM2 to

trigger the state analyzer version (i.e. that which requires an external clock to trigger the analyzer) on the logic analyzer. For this test, use MICTOR CLKM1 on the even # pod with a falling edge trigger to test the MICTOR output on the logic analyzer. Further, the counter is output through ioport. If the user connects the ioport to ioport1 with a (40) pin ribbon cable, the FPGA will compare the received data against the transmitted data. In order to see if error-free data is received, set switch_in(14) [S1-7] to "1" [OPEN] and toggle switch_in(15) [S1-8] from "0"->"1" [CLOSED->OPEN]. The LED, D4, will not be illuminated to show that all bits are received error free. To test the LVDS, the user must use a 80 MHz crystal and loopback the lvds_tx to the lvds_rx signals with a cable. In order to see if error-free data is received, set switch_in(14) [S1-7] to "1" [OPEN] and toggle switch_in(15) [S1-8] from "0"->"1" [CLOSED->OPEN]. The LED, D3, will not be illuminated to show that all bits are received error free.

Also, the configuration done LED located at D5 (near H1) will illuminate RED when the configuration is complete. The VHDL source code and the pin configuration file used to specify I/O ports are also included if the user would like to change, compile & test. We recommend to not use the "smart" compilation feature offered by Quartus to recompile. We have used the option "Power-Up Don't Care=OFF." Also, for QuartusII version 2.1, the .csf, .esf, and .psf files are included to setup the same environment as was used to create the .sof/.pof files. The pin configuration file is especially important as the complete set of interconnections between all ports and the Altera device is specified for the user even though they may not be used for every application. There is a very simple way to re-use these pin configuration files with any new project a user may create. First, after creating a project for Altera #1 for example, close the project and manually edit your_new_project.csf file. Copy the pin assignments from the Q5_1 file in the pin_configuration directory of the CD to your_new_project.csf file via high-lighting, copy & paste into the Chip section of the .csf file. Alternatively, you could insert the Q5_1 pin configuration file if your editor allows this. Your .csf file should look like:

```
CHIP(your_new_project)
{
        clk_1q0 : IO_STANDARD = LVTTL;
        clk_2q1 : IO_STANDARD = LVTTL;
        clk_xtal_J2 : IO_STANDARD = LVTTL;
        LVDS_rx : IO_STANDARD = LVDS;
        LVDS_rxclk : IO_STANDARD = LVDS;
        LVDS_tx : IO_STANDARD = LVDS;
        LVDS_txclk : IO_STANDARD = LVDS;
        clk_xtal_J2 : LOCATION = Pin_K17;
        clk_1q0 : LOCATION = Pin_AC17;
        clk_2q1 : LOCATION = Pin_P27;
        ioport1[0] : LOCATION = Pin_AH13;
         …
```

It is important to do this procedure while the project is closed. After the editing is done, open your project and check for the pin assignments with the Quartus[1] assignment organizer. Consult the Quartus[1] Users manual or call the Altera hot-line at 1-800-800-EPLD for the compilation procedure. Congratulations, you have successfully down-loaded into your new Q5 board.

To down-load the same VHDL code into the Flash Configuration Devices, start with the programmer tool setup as in the following table. Set the mode to "JTAG." The Programming hardware type is still "ByteBlasterMV." Set the programmer up as shown in the table below: (the checksums may be Quartus[1] version dependent. Quartus[1] II version 2.1 is assumed). The .pof file is on the CD in the pof file directory.

File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank-Check	Examine	Security Bit
Q5_1.pof	EPC16	1180FFD6	FFFFFFFF	√				

When this window is set up, the user must now change the ByteBlasterMV[1] cable from H6 to H7, once again matching pin 1 on the board with pin 1 of the ByteBlasterMV[1] cable. Once this is done, hit the "Start" button in the programmer window & the Flash Configuration Devices should be down-loaded. To activate this configuration, cycle power off and then on. Once again, notice the counter output to the LEDS. A factory generated .pof file for these diagnostics is included on the CD.

Section 3 – Other Board Attributes

3.1) Cypress Roboclock[2] timing generator

The Q5 board is equipped with a Cypress CY7B9945V Roboclock[2]. Please refer the datasheet available on the Cypress website for pertinent information to be discussed in this section. The Roboclock[2] datasheet used for this design was dated 8/30/02. The user has several programmable options available via jumper settings. The input clock, whether external via SMA J2 or internal via Crystal Oscillator U3, is connected to the REFA input. REFB is not connected and thus never used. Pin 1 of U3 is located directly adjacent to the U3 label, which is on the silk screen of the printed circuit board (see Figure 3). The Roboclock[2] outputs, 1Q0 and 2Q1, are tied to the Altera CLK4p and CLK0p signals respectively. The pin configuration files given with the board are setup to define both signals. The user should select either one of these signals by referencing the specific clock signal name within a process (see section 1.2). The 1Q0 and 2Q1 clocks have separate frequency and phase control via jumpers H10 and H9. The J1 SMA connector is connected to output 2Q0 or the crystal source to present a reference clock to other boards. As an example, cascading (2) Q5 boards is as easy as connecting an external clock source to J2 of board #1 and connecting J1 of board #1 to J2 of board #2. The jumper H11 is connected to the FS input, pin 34 of the Roboclock[2]. As shown in the Cypress datasheet Table 1, FS has 3 settings, LOW, MID or HIGH. To select FS=LOW,

for a 24-52 MHz frequency range, connect the jumper of H11 between pin 2 and pin 3. Pin 1 is located closest to the Roboclock^2 chip. To select FS=MID, for a 48-100 MHz frequency range, use no jumper on H11 at all. To select FS=HIGH, for a 96-200 MHz frequency range, connect pin 1 to pin2. FS is the frequency obtained after multiplication via the factor in Table 4 of the Cypress Datasheet. That is, if a crystal of 20 Mhz will be multiplied by 4, its feedback frequency is 80 MHz & FS should be set to MID for 80 Mhz, not LOW for 20 MHz.

Another set of jumpers, H9, is available to set the programmable skew as specified in Table 3 of the Cypress datasheet. Each row of 3 pins represents 1 input with an adjacent VCC and an adjacent Ground signal. Starting at pins 1-3 is the signal 1F0. To set it HIGH, connect a jumper between pin1 and pin2. Pin 1 is closest to the Roboclock chip. To set it to MID, make no connection whatsoever. To set it to LOW, connect pin2 and pin3. Similarly, moving to the next row is the signal 1F1. These would be pins 4-6. Set it's value to LOW, MID or HIGH in a similar fashion to 1F0. In a similar fashion, the 3rd row, pins 7-9, would be 1F2. The sequence would continue as 1F3, 2F0, 2F1, FBF0, which would wind up in the final row of pins 19-21. The user can set whatever skew they want as described in the Cypress datasheet Table 3. **Note that leaving all jumpers off of H9 would set the chip into a 0 skew arrangement for all the clock outputs**. Thus, the 2 clocks available to the Altera chip as well as the reference clock on J1 would have no skew between them.

Another set of jumpers, H10, is available to set the programmable frequency multiply/divide ratios as specified in Table 4 of the Cypress datasheet. Each row of 3 pins represents 1 input with an adjacent VCC and an adjacent Ground signal. Starting at pins 1-3 is the signal 1DS0 to control the frequency output of clk_1q0. To set it HIGH, connect a jumper between pin1 and pin2. To set it to MID, make no connection whatsoever. To set it to LOW, connect pin2 and pin3. Similarly, moving to the next row is the signal 1DS1 which is used together with 1DS0 to control the frequency of clk_1q0. These would be pins 4-6. Set it's value to LOW, MID or HIGH in a similar fashion to 1DS0. In a similar fashion, the 3rd row, pins 7-9, would be 2DS0. The sequence would continue as 2DS1, FBDS0, FBDS1, which would wind up in the final row of pins 16-18. The user can set whatever divider they want as described in the Cypress datasheet Table 4 which is reproduced below for clarity. **Note that setting leaving all jumpers off of H10 would set the chip into a divide by 1 transparent mode for all the clock outputs.** Thus, the 2 clocks available to the Altera chip as well as the reference clock on J1 would all be the same frequency as the crystal or J2 source. The FBDS0/FBDS1 select what frequency the Roboclock^2 chip will operate relative to its' reference. The DS0i/DS1i then select what divider to apply (for bank i) to the Roboclock^2 frequency. As an example, using a 20 MHz crystal, set FS to MID for 48-100 MHz feedback frequency range. Also, set FBDS1/FBDS0 to MID/LOW. This is a divide by 4 which forces the Roboclock^2 frequency to 80 MHz. That is, it actually multiplies the reference frequency by 4. Then set 1DS1/1DS0 to LOW/HIGH. This is a divide of the Roboclock^2 frequency by 3 which creates a clock at 1Q0 of 26.66 Mhz. Similarly, set 2DS1/2DS0 to HIGH/MID. This is a divide of the Roboclock^2 frequency by 10 which creates a clock at 2Q0, 2Q1 and 2Q5 of 8 Mhz. The clock 2Q5 is routed to the MICTOR connector for logic analyzer use as CLKM2 mentioned in section 1.1. The clock, 2q0, is routed to J1.

There is one final jumper, H8, which controls which signal is sent to J1, the output reference clock SMA connector. Pin 1 is located nearest to the Roboclock^2 chip as shown in Figure 3. To select an output directly from the crystal or J2 source to serve as a reference frequency to pass onto other boards, set a jumper between pin 2 and pin 3. To select the output from the Roboclock^2 signal 2Q0 to be the reference, set a jumper between pin 1 and pin 2. The idea here is that depending on whether or not the Roboclock^2 is being used as the clock master should determine which reference clock the user should pass onto other boards. However, any combination is allowed. Thus, the user could be using the Roboclock^2 for timing generation on the Q5 board and pass the crystal frequency to the next Q5 board via J1.

Frequency Control of the Roboclock

FBDS1	***FBDS0***	***Frequency Multiplier***	***FS [assuming a 20 Mhz crystal]***
L	L	1	L [20MHz]
L	M	2	L [40MHz]
L	H	3	M [60MHz]
M	L	4	M [80MHz]
M	M	5	H [100MHz]
M	H	6	H [120MHz]
H	L	8	H [160MHz]
H	M	10	H [200MHz]
H	H	12	H [240MHz]
1DS1	***1DS0***	***Frequency Clk_1q0***	***Frequency of Clk_1q0 assuming FS=100MHz***
L	L	FS	100 MHz
L	M	FS/2	50 MHz
L	H	FS/3	33.33 MHz
M	L	FS/4	25 MHz
M	M	FS/5	20 MHz
M	H	FS/6	16.66 MHz
H	L	FS/8	12.5 MHz
H	M	FS/10	10 MHz
H	H	FS/12	8.33 MHz
2DS1	***2DS0***	***Frequency Clk_2q0, Clk_2q1, Clk_2q5***	***Frequency of Clk_2q1 assuming FS=200MHz***
L	L	FS	200 MHz
L	M	FS/2	100 MHz
L	H	FS/3	66.66 MHz
M	L	FS/4	50 MHz
M	M	FS/5	40 MHz
M	H	FS/6	33.33 MHz

H	L	FS/8	25 MHz
H	M	FS/10	20 MHz
H	H	FS/12	16.66 MHz

3.2) On board power regulation.

The Q5 board has a single regulator to supply VCCINT. Only a single external 3.3v supply is required. The 1.5V regulator is rated at 3 amps.

3.3) Multiple I/O ports for easy interfacing

The Q5 board has multiple I/O ports for interfacing to analog converters, logic analyzers, user control bits or additional Q5V1 boards. For logic analyzer support, there is a single MICTOR connector carrying 32 bits and a choice of 2 clocks for sampling. Refer to section 1 for details.

3.4) Internal PLL's

The FPGA device supports up to (6) internal PLL's. The Q5V1 board does not support external clock driving capability (from within the FPGA) at this time. PLL4 and PLL1 are the fast PLL's meant for LVDS control. PLL4 is used for the receiver LVDS ports. It is routed to CLK11p/CLK11n. It can be instantiated through the megawizard plug-in manager as was done in the example provided in the code supplied. It requires a differential clock input which is derived directly from the LVDS differential receiver clock. PLL1 is used for the transmitter LVDS. It also requires a differential clock input. As such, for the Q5 board, the differential output of PLL5 is routed to the clock input of PLL1 to drive the transmitter LVDS. Thus, the PLL5 is used as a divide by 1 on a single ended clock to provide a differential clock to PLL1. Use the megawizard plug-in manager to set up these LVDS ports as per user application. An example is given in the vhdl code supplied. These three PLL's should be used first within each device as these PLL's have the separate 20 mil power and ground traces supplied to them as specified by Altera. Further, PLL4 and PLL1 should be used for LVDS control as they are in the same bank as the LVDS signals routed to the LVDS connectors. PLL2, PLL3, and PLL6 have power and ground as well, but do not have separate 20 mil traces. Thus, these could be used with caution. Refer to Altera application note AN115 for more information (p17). Even though the AN115 refers to the APEX devices, similar requirements are placed on the STRATIX devices.

3.5) LVDS support.

The Q5 board supports (16) LVDS channels for receiving and (16) LVDS channels for transmitting. Separate receive and transmit LVDS clocks are also supported. If LVDS is not desired, the 16 LVDS RX channels can be used as 32 LVTTL signals. The 16 LVDS TX channels can be used as 32 LVTTL signals and the LVDS TX clock can be used as 2 LVTTL signals. Thus, 66 general LVTTL I/O signals can be used on the LVDS connectors if the LVDS is not required.

3.6) Standalone Mode

Given the flash configuration device and U3 crystal, your Q5 board can operate in a standalone fashion without the need for a download cable or function generator.

3.7) Fast I/O Registers

It is usually a good idea to assign Fast I/O registers when communicating between the Altera FPGA and the connectors. Set these options for input and output

registers in the assignment organizer of Quartus[1]. Also, make certain that the vhdl code specifies latches for these signals. An example vhdl code is supplied in Q5_1.vhd in the source code directory of the CD.

[1]A registered trademark of Altera Corporation.
[2]A registered trademark of Cypress Corporation