UJTAG Applications in ProASIC3/E Devices

Introduction

In ProASIC3/E devices, there is bidirectional access from the JTAG port to the core VersaTiles during normal operation of the device. The UJTAG tiles, located at the southeast area of the die, are directly connected to the ProASIC3/E JTAG Test Access Port (TAP) Controller in normal operating mode. As a result, all the functional blocks of the ProASIC3/E architecture, such as Clock Conditioning Circuits (CCC) with PLLs, SRAM blocks, embedded FlashROM (FROM) blocks, and I/O tiles, can be reached via the JTAG ports. The UJTAG functionality is available by instantiating the UJTAG macro directly in the source code of a design. Access to the FPGA core VersaTiles from the JTAG ports enables users to implement different applications using the ProASIC3/E TAP Controller (JTAG port). This application note introduces the UJTAG tile functionality and discusses a few application examples. However, the possible applications are not limited to what is presented in this document. UJTAG can serve different purposes in many designs as an elementary or auxiliary part of the design.

UJTAG Macro

The UJTAG tiles can be instantiated in a design using the UJTAG macro from the ProASIC3/E library. Note that "UJTAG" is a reserved name and cannot be used for any other user-defined blocks. A block symbol of the UJTAG tile macro is presented in Figure 1 on page 2. In this figure, the ports on the left side of the block are connected to the JTAG TAP Controller, while the right-side ports are accessible by the FPGA core VersaTiles. The TDI, TMS, TDO, TCK, and TRST ports of UJTAG are only provided for design simulation purposes and should be treated as external signals in the design netlist. However, these ports must NOT be connected to any I/O buffer in the netlist. Figure 2 on page 2 illustrates the correct connection of the UJTAG macro to the user design netlist. Actel Designer software will automatically connect these ports to the TAP during place-and-route. The following are the port descriptions for the rest of the UJTAG ports:

UIREG [7:0]: This eight-bit bus carries the contents of the JTAG Instruction Register of ProASIC3/E devices. Instruction Register values 16 to 127 are not reserved and can be employed as user-defined instructions.

URSTB: The URSTB is an active low signal and will be asserted when the TAP Controller is in test-logic-reset mode. The URSTB is asserted at power-up, and a power-on-reset signal resets the TAP Controller. The URSTB will stay asserted until an external TAP access changes the TAP Controller state.

UTDI: This port is directly connected to the TAP's TDI signal.

UTDO: This port is the user TDO output. Inputs to the UTDO port are sent to the TAP TDO output MUX when the IR address is in user range.

UDRSH: Active high signal enabled in ShiftDR TAP state.

UDRCAP: Active high signal enabled in the CaptureDR TAP state.

UDRCK: This port is directly connected to the TAP's TCK signal.

UDRUPD: Active high signal enabled in the UpdateDR TAP state.

UJTAG Operation

There are a few basic functions of the UJTAG macro that users must understand before designing with it. The most important fundamental concept of the UJTAG design is its connection with the TAP Controller state machine.

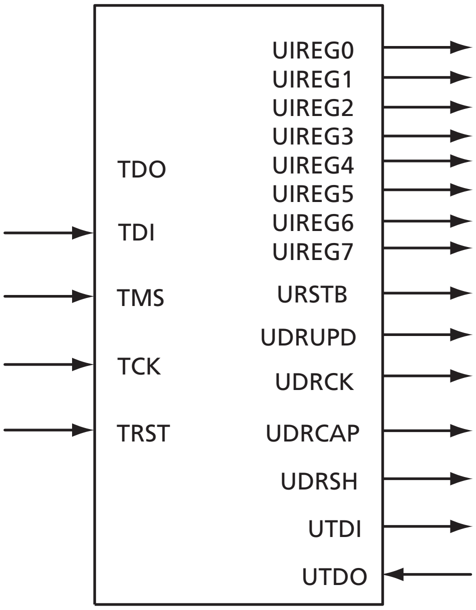
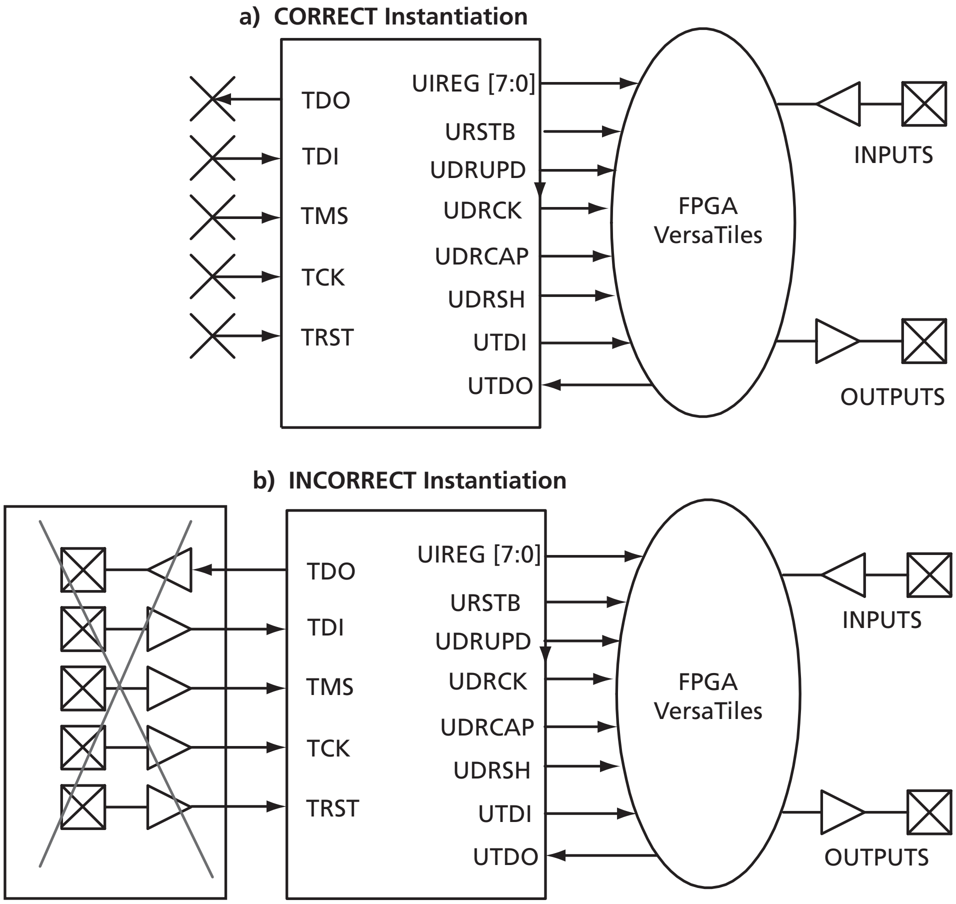


Figure 1 • **UJTAG Tile Block Symbol**



Note: *Do not connect JTAG pins (TDO, TDI, TMS, TCK, or TRST) to I/Os in the design.*

Figure 2 • **Connectivity Method of UJTAG Macro**

TAP Controller State Machine

The 16 states of the Tap Controller state machine are shown in Figure 3. The 1s and 0s, shown adjacent to the state transitions, represent the TMS values that must be present at the time of a rising TCK edge for a state transition to occur. In the states that include the letters IR, the instruction register operates; in the states that contain the letters DR, the test data register operates. The TAP Controller receives two control inputs, TMS and TCK, and generates control and clock signals for the rest of the test logic.

On power-up (or the assertion of TRST), the TAP Controller enters the Test-Logic Reset state. To reset the controller from any other state, TMS must be held high for at least five TCK cycles. After reset, the TAP state changes at the rising edge of TCK, based on the value of TMS.

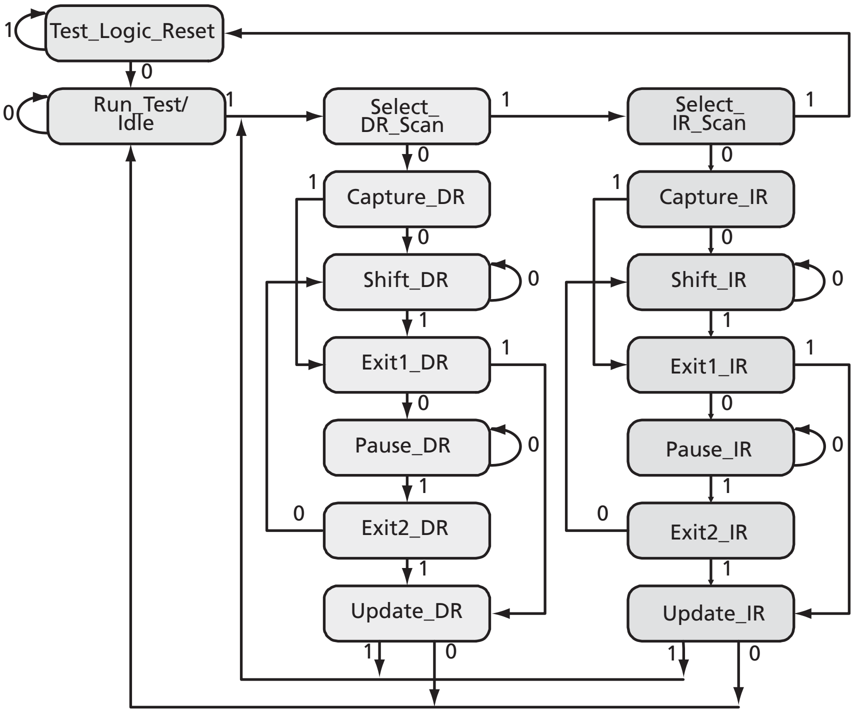


Figure 3 • **TAP Controller State Diagram**

UJTAG Port Usage

UIREG[7:0] hold the contents of the JTAG instruction register. The UIREG vector value will be updated when the TAP Controller state machine enters the Update_IR state. Instructions 16 to 127 are user-defined and can be employed to encode multiple applications and/or commands within an application. Loading new instructions into the UIREG vector requires users to send appropriate logic to TMS in order to put the TAP Controller in a full IR cycle starting from the "Select IR_Scan" state and ending with the "Update_IR" state.

UTDI, UTDO, and UDRCK are directly connected to the JTAG TDI, TDO, and TCK ports, respectively. The TDI input can be used to provide either data (TAP Controller in "Shift_DR" state) or the new contents of the instruction register (TAP Controller in "Shift_IR" state).

UDRSH, UDRUPD, and UDRCAP are high when the TAP Controller state machine is in the Shift_DR, Update_DR, and Capture_DR states, respectively. Therefore, they act as flags to indicate the stages of the data shift process. These flags are useful for applications in which blocks of data are shifted into the design from JTAG pins. For example, an active UDRSH can indicate that the UTDI contains the data bitstream, while UDRUPD is a candidate for the end-of-data-stream flag.

As mentioned earlier, users should not connect the TDI, TDO, TCK, TMS, and TRST ports of the UJTAG macro to any port or net of the design netlist. The Designer software will automatically handle the port connection.

Typical UJTAG Applications

Bidirectional access to the JTAG port from VersaTiles—without putting the device into test mode— creates flexibility to implement many different applications. This section describes a few of these. All are based on importing/exporting data through the UJTAG tiles.

Clock Conditioning Circuitry—Dynamic Reconfiguration

ProASIC3/E clock conditioning circuits (CCCs), which include PLLs, can be configured dynamically through either an 81-bit embedded shift register or static Flash programming switches. These 81 bits control all the characteristics of the CCC: routing MUX architectures, delay values, divider values, etc. Table 1 lists the 81 configuration bits in the CCC.

Table 1 • **Configuration Bits of ProASIC3/E CCC Blocks**

Bit Number	Control Function
80	RESET ENABLE
79	DYNCSEL
78	DYNBSEL
77	DYNASEL
<76:74>	VCOSEL [2:0]
73	STATCSEL
72	STATBSEL
71	STATASEL
<70:66>	DLYC [4:0]
<65:61>	DLYB {4:0]
<60:56>	DLYGLC [4:0]
<55:51>	DLYGLB [4:0]
<50:46>	DLYGLA [4:0]
45	XDLYSEL
<44:40>	FBDLY [4:0]
<39:38>	FBSEL
<37:35>	OCMUX [2:0]
<34:32>	OBMUX [2:0]
<31:29>	OAMUX [2:0]
<28:24>	OCDIV [4:0]
<23:19>	OBDIV [4:0]
<18:14>	OADIV [4:0]
<13:7>	FBDIV [6:0]
<6:0>	FINDIV [6:0]

The embedded 81-bit shift register (for the dynamic configuration of the CCC) is accessible to the VersaTiles which, in turn, have access to the UJTAG tiles. Therefore, the CCC configuration shift register can receive and load the new configuration data stream from JTAG.

Dynamic reconfiguration eliminates the need to reprogram the device when reconfiguration of the CCC functional blocks is needed. The CCC configuration can be modified while the device continues to operate. Employing the UJTAG core requires the user to design a module to provide the configuration data and control the CCC configuration shift register. In essence, this is a user-designed TAP Controller requiring chip resources.

Similar reconfiguration capability exists in the Actel ProASIC$^{\underline{PLUS}}$ family. The only difference is the number of shift register bits controlling the CCC (27 in ProASIC$^{\underline{PLUS}}$ vs. 81 in ProASIC3/E).

Fine Tuning

In some applications, design constants or parameters need to be modified after programming the original design. The tuning process can be done using the UJTAG tile without reprogramming the device with new values. If the parameters or constants of a design are stored in distributed registers or embedded SRAM memory blocks, the new values can be shifted onto the JTAG TAP Controller pins, replacing the old values. The UJTAG tile is used as the "bridge" for data transfer between the JTAG pins and the FPGA VersaTiles or SRAM logic. Figure 4 shows a flow chart example for fine tuning application steps using the UJTAG tile.

In Figure 4, the TMS signal sets the TAP Controller state machine to the appropriate states. The flow mainly consists of two steps: a) shifting the defined instruction and b) shifting the new data. If the target parameter is constantly used in the design, the new data can be shifted into a temporary shift register from UTDI. The UDRSH output of UJTAG can be used as a shift-enable signal, while UDRCK is the shift clock to the shift register. Once the shift process is completed, and the TAP Controller state is moved to the Update_DR state, the UDRUPD output of the UJTAG can latch the new parameter value from the temporary register onto a permanent location. This avoids any interruption or malfunctioning during the serial shift of the new value.

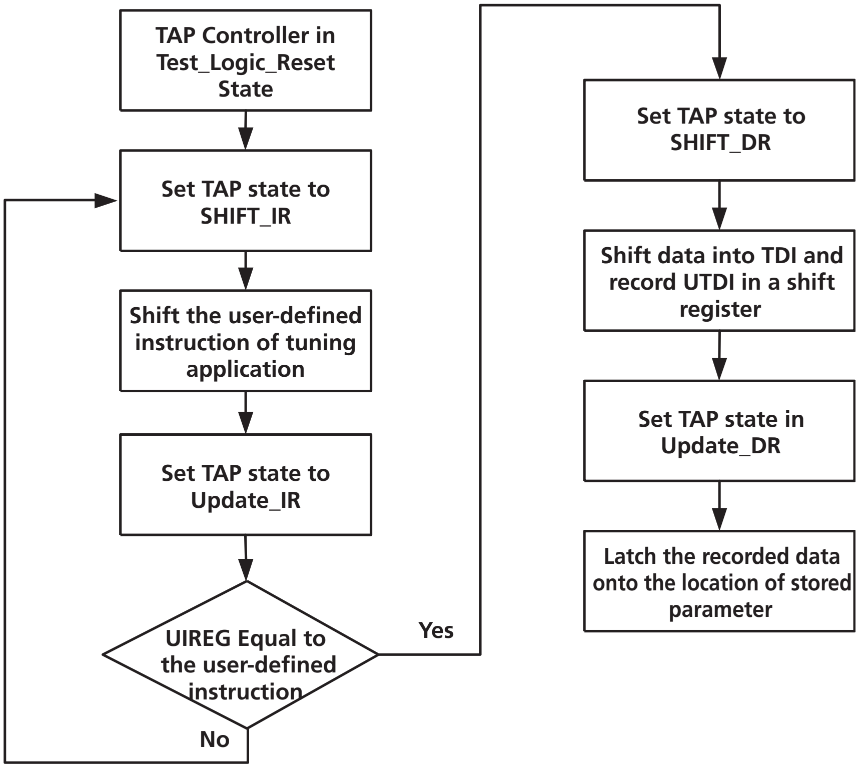

TAP Controller in Test_Logic_Reset State
Set TAP state to SHIFT_IR
Shift the user-defined instruction of tuning application
Set TAP state to Update_IR
UIREG Equal to the user-defined instruction
Yes
No
Set TAP state to SHIFT_DR
Shift data into TDI and record UTDI in a shift register
Set TAP state in Update_DR
Latch the recorded data onto the location of stored parameter

Figure 4 • **Flow Chart Example of Fine-Tuning an Application Using UJTAG**

Silicon Testing and Debugging

In many applications, the design needs to be tested, debugged, and verified on real silicon or in the final embedded application. In order to debug and test the functionality of designs, users may need to monitor some internal logic (or nets) during device operation. The approach of adding design test pins to monitor the critical internal signals has many disadvantages, such as limiting the number of user I/Os. Furthermore, adding external I/Os for test purposes may require additional or dedicated board area for testing and debugging.

The UJTAG tiles of ProASIC3/E devices offer a flexible and cost-effective solution for silicon test and debug applications. In this solution, the signals under test are shifted out to the TDO pin of the TAP Controller. The main advantage is that all the test signals are monitored from the TDO pin; no pins or additional board-level resources are required. Figure 5 illustrates this technique. Multiple test nets are brought into an internal MUX architecture. The selection of the MUX is done using the contents of the TAP Controller instruction register, where individual instructions (values from 16 to 127) correspond to different signals under test. The selected test signal can be synchronized with the rising or falling edge of the TCK (optional) and sent out to UTDO to drive the TDO output of JTAG.

The test and debug procedure is not limited to the example in Figure 4 on page 5. Users can customize the debug and test interface to make it appropriate for their applications. For example, multiple test signals can be registered and then sent out through UTDO, each at a different edge of TCK. In other words, n signals are sampled with an FTCK/n sampling rate. The bandwidth of the information sent out to TDO is always proportional to the frequency of TCK.

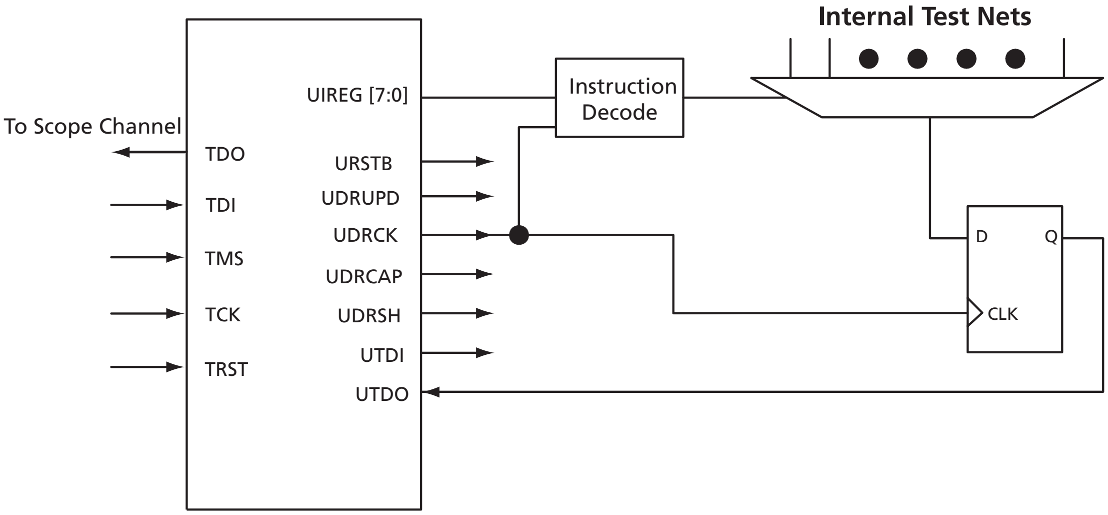


Figure 5 • **UJTAG Usage Example in Test and Debug Applications**

SRAM Initialization

Users can also initialize embedded SRAM memories of the ProASIC3/E devices. The initialization of the embedded SRAM blocks of the design can be done using UJTAG tiles where the initialization data is imported using the TAP Controller. Similar functionality is available in ProASIC^{PLUS} devices using JTAG. The guidelines for implementation and design examples are given in *RAM Initialization and ROM Emulation in ProASIC^{PLUS} Devices*.

Users can apply the same approach to initialize the embedded SRAM blocks of ProASIC3/E devices. SRAM memories are volatile by nature; data is lost in the absence of power. Therefore, the initialization process should be done at each power-up, if necessary.

FlashROM (FROM) Read Back Using JTAG

The ProASIC3/E architecture contains a dedicated nonvolatile FROM memory block which is formatted into eight 128-bit pages. For more information on FROM, please refer to the *ProASIC3/E FlashROM (FROM)* application note. The contents of FROM memory are available to the VersaTiles during normal operation through a read operation. As a result, the UJTAG macro can be used to provide the FROM contents to the JTAG port during normal operation. Figure 6 illustrates a simple block diagram of using UJTAG to read the contents of FROM during normal operation.

The FROM read address can be provided from outside the FPGA through the TDI input or can be generated internally using the core logic. In either case, data serialization logic is required (Figure 6) and should be designed using the VersaTile core logic. FROM contents are read asynchronously in parallel from the Flash memory and shifted out in a synchronous serial format to TDO. Shifting the serial data out of the serialization block should be performed while the TAP is in UDRSH mode. The coordination between the TCK and the data shift procedure can be done using the TAP state machine by monitoring UDRSH, UDRCAP, and UDRUPD.

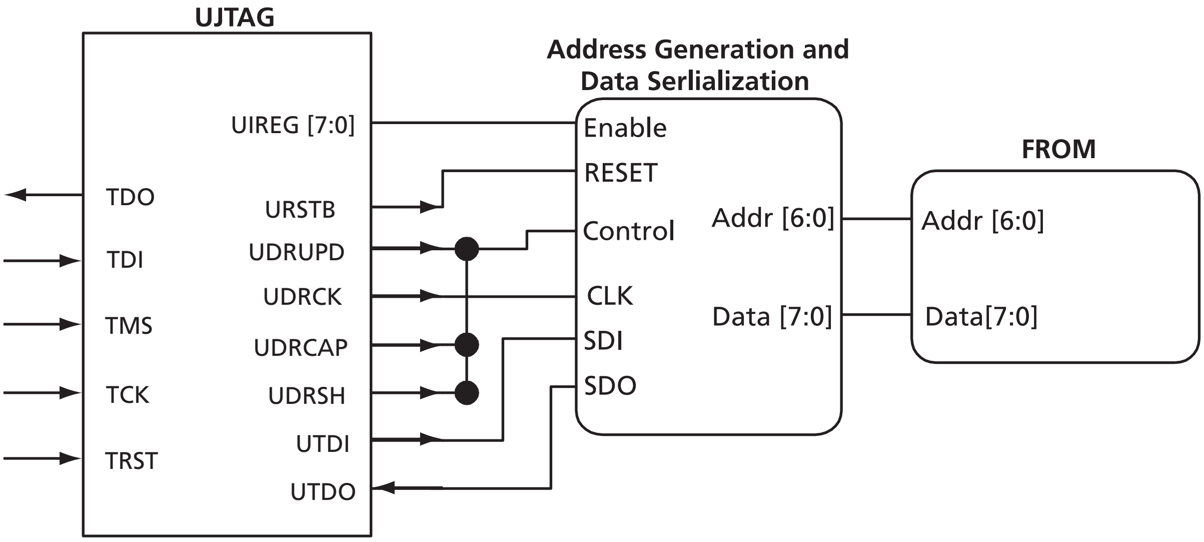


Figure 6 • **Block Diagram of Using UJTAG to Read FROM Contents**

Conclusion

Actel Flash FPGAs offer many unique advantages such as security, nonvolatility, reprogrammablity, and low power—all in a single chip. In addition, ProASIC3/E devices provide access to the JTAG port from core VersaTiles, while the device is in normal operating mode. A wide range of available user-defined JTAG OPCODEs allows users to implement various types of applications, exploiting this feature of ProASIC3/E devices. The connection between the JTAG port and core tiles is implemented through an embedded and hardwired UJTAG tile. A UJTAG tile can be instantiated in designs using the UJTAG library cell. This application note presents multiple examples of UJTAG applications, such as dynamic reconfiguration, silicon test and debug, fine-tuning of the design, and RAM initialization. Each of these applications offers many useful advantages.

Related Documents

Application Notes

RAM Initialization and ROM Emulation in ProASIC$^{\underline{PLUS}}$ Devices

http://www.actel.com/documents/APA_RAM_Initd_AN.pdf

ProASIC3/E FlashROM (FROM)

http://www.actel.com/documents/PA3_E_FROM_AN.pdf

http://www.actel.com

Actel Corporation
2061 Stierlin Court
Mountain View, CA
94043-4655 USA
Phone 650.318.4200
Fax 650.318.4600

Actel Europe Ltd.
Dunlop House, Riverside Way
Camberley, Surrey GU15 3YL
United Kingdom
Phone +44 (0) 1276 401 450
Fax +44 (0) 1276 401 490

Actel Japan
EXOS Ebisu Bldg. 4F
1-24-14 Ebisu Shibuya-ku
Tokyo 150 Japan
Phone +81.03.3445.7671
Fax +81.03.3445.7668

Actel Hong Kong
39th Floor, One Pacific Place
88 Queensway, Admiralty
Hong Kong
Phone +852.227.35712
Fax +852.227.35999