



XAPP660 (v2.2) February 4, 2004

Dynamic Reconfiguration of RocketIO MGT Attributes

Author: Derek R. Curd

Summary

This application note describes a pre-engineered design module for Virtex-II Pro™ devices that enables dynamic reconfiguration of RocketIO™ Multi-Gigabit Transceiver (MGT) attributes. This solution is ideal for any application in which these attributes must be modified to customize the MGT behavior for various system conditions while leaving the rest of the FPGA design unchanged. The MGT reconfiguration module uses less than 75 slices of FPGA logic and can be added to any Virtex-II Pro design simply by including the HDL (Verilog or VHDL) source file and a device-specific User Constraints File (UCF) in a standard Integrated Software Environment (ISE) project. The associated reference design files provide support for all members of the Virtex-II Pro family.

Introduction

Virtex-II Pro devices contain between four and twenty RocketIO MGTs that allow the creation of high-speed serial links (up to 3.125 Gb/s per channel) between devices. The MGTs have several user-defined settings that are selected through RocketIO primitive attributes. These attributes allow the MGT transmit and receive characteristics to be customized for a specific design. For more details on these attributes or the MGTs in general, refer to [UG024, RocketIO Transceiver User Guide](#).

In many cases it might be desirable to modify one or more of the MGT attributes on a Virtex-II Pro device, while leaving the rest of the FPGA design unchanged. For example, when using the MGTs to create high-speed serial links across a backplane, the distance the signals must travel can change significantly depending on which slot the board is plugged into. Adjusting the attribute settings for pre-emphasis (TX_PREEMPHASIS) and/or differential swing control (TX_DIFF_CTRL) to compensate for the change in distance allows for the highest quality signal transmission at the intended baud rate.

This application note describes an MGT reconfiguration design module that allows on-the-fly modification of the MGT primitive attributes through a straightforward register interface. The provided design files implement a completely self-contained solution that can easily be added to any standard ISE project.

© 2004 Xilinx, Inc. All rights reserved. All Xilinx trademarks, registered trademarks, patents, and further disclaimers are as listed at <http://www.xilinx.com/legal.htm>. All other trademarks and registered trademarks are the property of their respective owners. All specifications are subject to change without notice.

NOTICE OF DISCLAIMER: Xilinx is providing this design, code, or information "as is." By providing the design, code, or information as one possible implementation of this feature, application, or standard, Xilinx makes no representation that this implementation is free from any claims of infringement. You are responsible for obtaining any rights you may require for your implementation. Xilinx expressly disclaims any warranty whatsoever with respect to the adequacy of the implementation, including but not limited to any warranties or representations that this implementation is free from claims of infringement and any implied warranties of merchantability or fitness for a particular purpose.

Using the MGT Reconfiguration Module

Table 1 shows the six MGT attributes that are supported in version 2.0 of this design. Note that this is only a small subset of the total number of MGT attributes. Refer to [UG024](#) for complete descriptions of all MGT attributes.

Table 1: Supported MGT Attributes

Attribute	Settings	Description
TX_DIFF_CTRL	400 mV, 500 mV, 600 mV, 700 mV, 800 mV	Sets the voltage difference or swing between the differential lines of the MGT output driver
TX_PREEMPHASIS	10%, 20%, 25%, 33%	Sets the pre-emphasis value of the MGT output driver
CHAN_BOND_MODE	OFF, MASTER, SLAVE_1_HOP, SLAVE_2_HOPS	Sets channel bonding mode for each MGT
CHAN_BOND_ONE_SHOT	TRUE, FALSE	Determines repeated execution of channel bonding
RX_DECODE_USE	TRUE, FALSE	Determines if 8B/10B decoding is bypassed
REF_CLK_V_SEL	0, 1	Selects between REFCLK and BREFCLK paths as the MGT reference clock

Hardware Module

Figure 1 depicts the MGT reconfiguration module that can be instantiated into any Virtex-II Pro based design for on-the-fly reconfiguration of MGT attributes through a simple register interface. Both Verilog (MGT_cntlr.v) and VHDL (MGT_cntlr.vhd) versions of this module are provided in the reference design files.

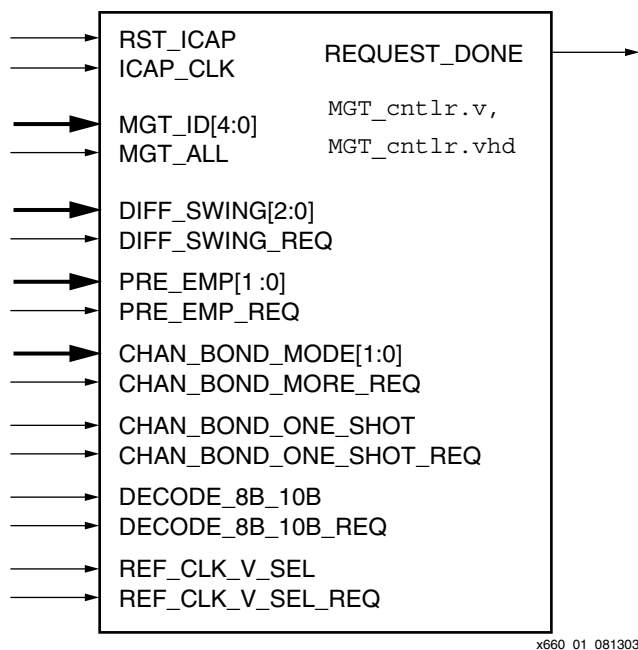
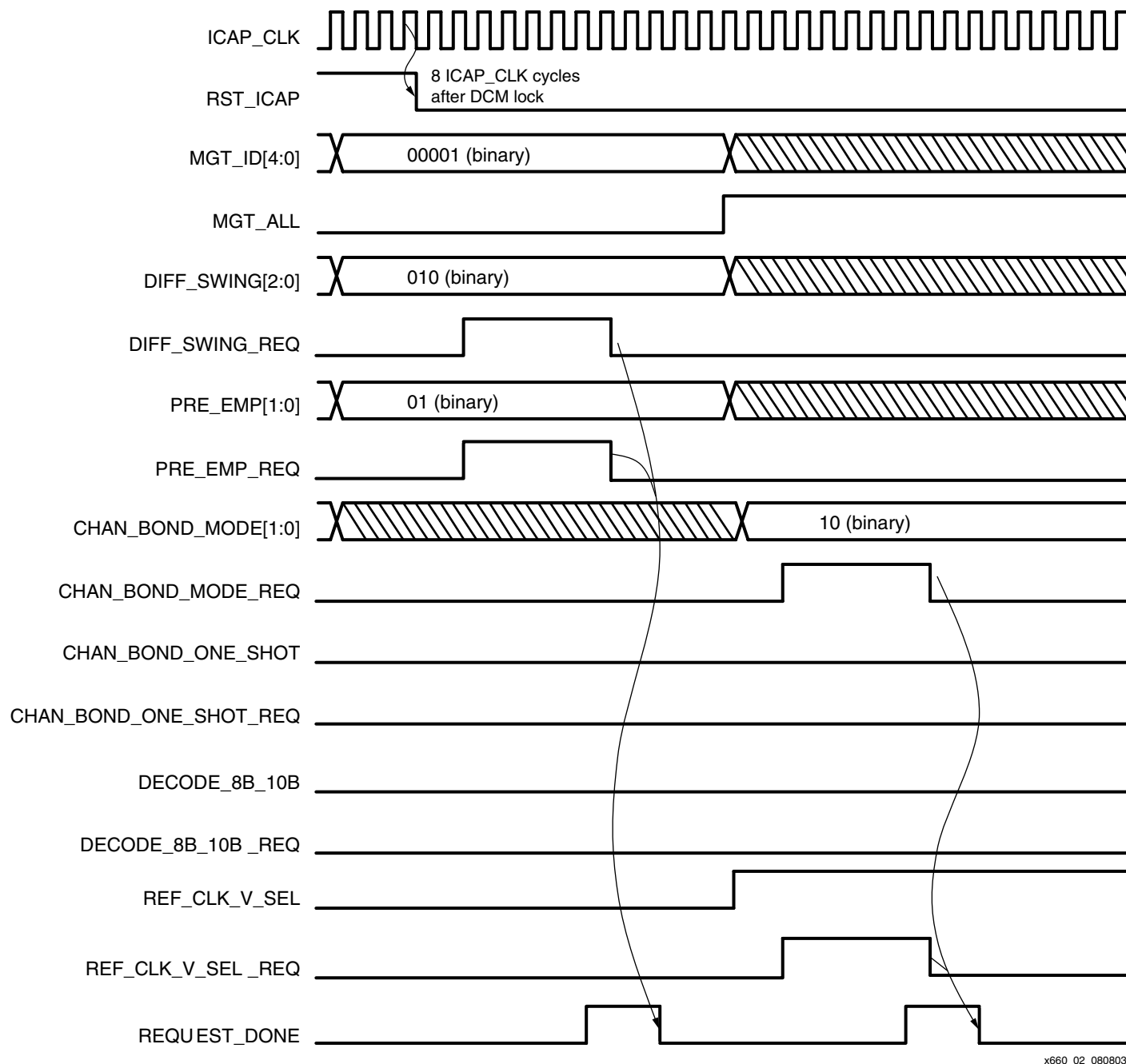


Figure 1: MGT Reconfiguration Module

In addition to the signals shown in Figure 1, the MGT reconfiguration module also has a set of five JTAG-related signals that can be connected to add the PPC405 into the device JTAG chain

for debug purposes. (Refer to the [PowerPC 405 Processor Block Reference Guide](#) for details). This may be required when using the MGT reconfiguration module in a Virtex-II Pro device with two PPC405 cores. Otherwise, the three JTAG inputs (JTGC405TCK, JTGC405TDI, and JTGC405TMS) can be tied to ground.

The MGT reconfiguration module includes attribute value ports (for example, DIFF_SWING[2:0]) and attribute request signals (for example, DIFF_SWING_REQ) for each of the six supported MGT attributes. To modify one or more attributes, simply apply a chosen value to the attribute value port(s) and strobe the associated attribute request signal(s). A specific MGT can be addressed through the MGT_ID[4:0] bus, limiting the reconfiguration of attributes to a single MGT. Alternatively, the same attribute changes can be applied to all MGT blocks by asserting the MGT_ALL input (active High) during a reconfiguration request.



x660_02_080803

Figure 2: MGT Reconfiguration Module Waveform Example

Figure 2 shows an example of modifying the DIFF_SWING and PRE_EMP values on a single MGT followed by a change to the CHAN_BOND_MODE and REF_CLK_V_SEL values on all MGTs. Due to the nature of the configuration logic in Virtex-II Pro devices, individual MGT attribute settings can be modified on-the-fly, without interrupting the operation of other MGTs or the rest of the FPGA.

All inputs to the MGT reconfiguration module are captured on the rising edge of ICAP_CLK, and setup and hold times for these signals must be relative to this edge. When multiple attributes are selected for reconfiguration during the same reconfiguration cycle, all associated request signals must be brought active within the same ICAP_CLK cycle to ensure the requests are captured simultaneously on the next rising edge. All request signals must be held active (High) until the REQUEST_DONE output goes active (High), indicating that the reconfiguration is complete. All active request signals then are brought Low, causing the REQUEST_DONE output to return to an inactive (Low) state. Then another request can be issued. The module interface requires that no new requests are made until the REQUEST_DONE signal returns to a Low state.

If certain attributes are always modified simultaneously, then multiple request signals can be driven by a single collective request signal. In addition, the request signals for any attributes that are never modified must be tied to a static Low.

In cases where the MGT attributes are modified only once after initial device power-up and configuration, the attribute value and request signals can be tied to static levels to make a one-time request. For example, when it is desired to set different DIFF_SWING and PRE_EMP values for all MGT blocks on a device based on a backplane slot location, a Slot ID code from the backplane can be decoded to set static values on the DIFF_SWING and PRE_EMP ports. The associated request signals for these attributes and the MGT_ALL signal also must be set to a static High. When the RST_ICAP signal goes inactive, the MGT reconfiguration module performs a one-time modification to the selected attributes on all MGT blocks, after which the REQUEST_DONE signal goes active (High) permanently or until another RST_ICAP strobe occurs.

Table 2 defines the clock (ICAP_CLK) and reset (RST_ICAP) specifications for the MGT reconfiguration module. The ICAP_CLK frequency must not exceed 33 MHz, and the RST_ICAP signal must remain active (High) during initial device power-up and configuration and also for eight clock cycles after ICAP_CLK becomes stable, as in the case where it is driven by a Digital Clock Manager (DCM) that has just achieved phase lock.

Table 2: Clock and Reset Specifications for MGT Reconfiguration Module

Signal	Frequency Range (Min. to Max.)	Details
ICAP_CLK	0.0 to 33.0 MHz	Continuous clock signal for registering inputs and synchronizing reconfiguration operations. The maximum frequency must not exceed 33 MHz.
RST_ICAP	-	Asynchronous reset (Active High). This signal must be active throughout initial device power-up and configuration, and for at least eight clock cycles after ICAP_CLK becomes stable. Assertion of this signal does <i>not</i> reset any device configuration logic or memory cells.

Table 3 defines the MGT_ID[4:0] values required to address the different MGT blocks based on their physical (X, Y) location and device size. To ensure a consistent relationship between the MGT_ID value and a specific MGT instance name, LOC statements are recommended in the UCF to tie MGT instances to specific physical MGT block locations.

Table 3: Association of MGT_ID[4:0] Value to MGT Physical Block Location

MGT_ID [4:0]	MGT Physical Location (X,Y)	XC2VP2	XC2VP4	XC2VP7	XC2VP20	XC2VP30	XC2VP40	XC2VP50	XC2VP70	XC2VP100
00000	GT_X0Y0	X	X	X	X	X	X	X	X	X
00001	GT_X0Y1	X	X	X	X	X	X	X	X	X
00010	GT_X1Y0	X	X	X	X	X	X	X	X	X
00011	GT_X1Y1	X	X	X	X	X	X	X	X	X
00100	GT_X2Y0			X	X	X	X	X	X	X
00101	GT_X2Y1			X	X	X	X	X	X	X
00110	GT_X3Y0			X	X	X	X	X	X	X
00111	GT_X3Y1			X	X	X	X	X	X	X
01000	GT_X4Y0						X	X	X	X
01001	GT_X4Y1						X	X	X	X
01010	GT_X5Y0						X	X	X	X
01011	GT_X5Y1						X	X	X	X
01100	GT_X6Y0							X	X	X
01101	GT_X6Y1							X	X	X
01110	GT_X7Y0							X	X	X
01111	GT_X7Y1							X	X	X
10000	GT_X8Y0								X	X
10001	GT_X8Y1								X	X
10010	GT_X9Y0								X	X
10011	GT_X9Y1								X	X

Table 4 shows the bit definitions for each attribute value port on the MGT reconfiguration module interface, along with the associated attribute settings.

Table 4: Relationship between Port Bit Definitions and Attribute Settings

DIFF_SWING[2:0]	TX_DIFF_CTRL
0 0 0	400 mV
0 0 1	500 mV
0 1 0	600 mV
0 1 1	700 mV
1 0 0	800 mV

PRE_EMP[1:0]	TX_PREEMPHASIS
0 0	10%
0 1	20%
1 0	25%
1 1	33%

Table 4: Relationship between Port Bit Definitions and Attribute Settings (Continued)

CHAN_BOND_MODE[1:0]	CHAN_BOND_MODE
0 0	OFF
0 1	MASTER
1 0	SLAVE_1_HOP
1 1	SLAVE_2_HOPS

CHAN_BOND_ONE_SHOT	CHAN_BOND_ONE_SHOT
0	FALSE
1	TRUE

DECODE_8B_10B	RX_DECODE_USE
0	FALSE
1	TRUE

REF_CLK_V_SEL	REF_CLK_V_SEL
0	0 (REFCLK)
1	1 (BREFCLK)

Instantiating the MGT Reconfiguration Module

The reference design files include both Verilog and VHDL versions of the Xilinx optimized MGT reconfiguration module. This module uses only two block RAMs (BRAMs) and less than 75 slices of logic, in addition to the PPC405 core, which makes it possible to add the module to almost any Virtex-II Pro based design. The following implementation flows apply to all Virtex-II Pro devices with the exception of the smallest family member, the XC2VP2. For specific instructions on reconfiguring MGT attributes for the XC2VP2 device, see the section entitled “XC2VP2 Design Specifics,” below.

Verilog Flow

The following steps list the Verilog implementation flow:

1. Unzip the reference design files into a safe directory and navigate to the newly created “Verilog” subdirectory.
2. Add the `MGT_cntl0.v` file from the “Verilog” directory to your ISE project, instantiate it (preferably at the top level), and connect it to the proper signals in your design.
3. If you are not using XST for synthesis, uncomment the ``define SYN_TOOL_NOT_XST` line at the top of the `MGT_cntl0.v` file. When using XST, make sure that the “Keep Hierarchy” option is enabled in the synthesis properties.
4. Append the device-specific UCF file (for example, `2vp4_Verilog.ucf` for the XC2VP4 device) to the UCF file for the entire design. The appended UCF file includes the BRAM initialization statements for the PPC405 instructions. It assumes that the `MGT_cntl0.v` module has been instantiated at the top level of the design with the instance name “MGT_cntl0”. If this is not the case, then the hierarchical paths to the BRAM in the UCF file need to be modified to reflect the correct path.

VHDL Flow

The following steps list the VHDL implementation flow:

1. Unzip the reference design files into a safe directory and navigate to the “VHDL” subdirectory that is created.

2. Add the `MGT_cntl_r.vhd` file from the “VHDL” directory to your ISE project, instantiate it (preferably at the top level), and connect it to the proper signals in your design.
3. If you are using XST for synthesis, make sure that the “Keep Hierarchy” option is enabled in the synthesis properties.
4. Append the device specific UCF file (for example, `2vp4_VHDL.ucf` for the XC2VP4 device) to the UCF file for the entire design. The appended UCF file includes the BRAM initialization statements for the PPC405 instructions. It assumes that the `MGT_cntl_r.vhd` module has been instantiated at the top level of the design with the instance name “MGT_cntlr0”. If this is not the case, then the hierarchical paths to the BRAM in the UCF file need to be modified to reflect the correct path.

When the HDL source file and UCF file have been added to the ISE project, the MGT reconfiguration module is integrated automatically into the overall design through the standard synthesis and FPGA implementation tool flow. Note that it is not possible to accurately simulate the behavior of the module because there is no simulation model for the device configuration logic and memory.

XC2VP2 Design Specifics

The XC2VP2 device does not contain a PPC405 core, and, therefore, cannot be supported with the standard MGT reconfiguration module (`MGT_cntl_r.v`, `MGT_cntl_r.vhd`). However, the reference design files contain a functionally identical solution for the XC2VP2 based on Xilinx’s PicoBlaze™ soft processor core. This solution uses two BRAMs and 182 slices of the FPGA logic. As with the other family members, both Verilog and VHDL versions of the design files are available for the XC2VP2 device. The implementation flow for both Verilog and VHDL versions is the same, as follows:

Verilog and VHDL Flow

1. Unzip the reference design files into a safe directory and navigate to either the newly created “Verilog” or “VHDL” subdirectory.
2. Add either the `MGT_cntl_r_2VP2.v` or `MGT_cntl_r_2VP2.vhd` file from this directory to your ISE project, instantiate it, and connect it up to the proper signals in your design.

A UCF file is not required for the XC2VP2 solution. When the HDL source file has been added to the ISE project, the MGT reconfiguration module is integrated automatically into the overall design through the standard synthesis and FPGA implementation tool flow. Note that the XC2VP2 HDL files are specific to this device and will not function properly when added to a design project for a different Virtex-II Pro family member. In addition, XST must be used to synthesize the XC2VP2 version of the MGT reconfiguration module. Other synthesis tools are not supported.

ISE Design Example

The “ISE_project” subdirectory within either the “Verilog” or “VHDL” directory contains a complete ISE design example for demonstrating the use of the MGT reconfiguration module in a simple design. This design example targets the XC2VP4 device and contains the necessary constraints file statements to target the Virtex-II Pro 2VP4-FG456 Development Board available from Insight (Memec) Electronics. The pinout constraints in the UCF file can be modified to target other development platforms.

A basic clock and reset module is included to demonstrate one method of generating the required ICAP_CLK and RST_ICAP signals from a DCM. In addition, the 16-bit TXDATA parallel interfaces to the MGT blocks are tied to constant values to generate continuous K28.5 characters for demonstration purposes. The design example as provided only demonstrates modification of the pre-emphasis attribute (TX_PREEMPHASIS). It can be modified to demonstrate modification of the other attributes supported by the MGT reconfiguration module by simply changing the top-level connectivity and pinouts.

Important Notes

1. The MGT reconfiguration module uses the Internal Configuration Access Port (ICAP) found inside Virtex-II Pro devices to modify the MGT attributes. The ICAP port interface is similar to the SelectMAP interface, but is accessible from general interconnect rather than the device pins. The "Boundary Scan" configuration mode pin setting (M2:M0 = 101) will disable the ICAP interface. Therefore, when using the MGT reconfiguration module, another mode pin setting must be used. If JTAG will be used as the primary configuration method, select another mode pin setting (recommendation is Slave Serial, M2:M0 = 111) to avoid disabling the ICAP interface. JTAG configuration will still be available because it overrides other means of configuration, and the MGT reconfiguration module will function as intended.
2. Designs incorporating the MGT reconfiguration module should NOT be downloaded to Virtex-II Pro devices using the ChipScope or ChipScope Pro software tools due to potential conflict with the ICAP interface. All other configuration tools / methods are supported (e.g. iMPACT, loading from Serial PROM / SystemACE, etc.). This issue will be resolved in Version 6.1i, Service Pack 3 of ChipScope / ChipScope Pro. In addition, all other features of ChipScope (e.g. the Logic Analyzer functions) CAN be used with designs incorporating the MGT reconfiguration module. It is only the bitstream download function of ChipScope that should be avoided.
3. The ICAP interface used by the MGT reconfiguration module accesses the same device configuration logic as the standard configuration pins. Therefore, all requests to the MGT reconfiguration module should be suspended if the Virtex-II Pro device is going to be completely reconfigured with a new bitstream without powering down or otherwise clearing the contents of the device (for example, by strobing the PROG_B pin).
4. The BitGen PERSIST option must be set to "NO" in order for the ICAP interface to take control of the device configuration logic. If PERSIST=YES, then the configuration logic is NOT accessible through the ICAP interface and the MGT reconfiguration module will not function properly.

Conclusion

This application note has shown how to incorporate a pre-engineered solution for Virtex-II Pro devices to enable dynamic reconfiguration of MGT attributes. The provided MGT reconfiguration module is completely self-contained and uses very few device resources, allowing it to be added to nearly any Virtex-II Pro based application. The module can be integrated easily into a standard ISE development flow and allows Virtex-II Pro MGT blocks to meet the demands of changing system environments and varying signal topologies.

Reference Design Code

The reference design code is available on the Xilinx web site at:

<http://www.xilinx.com/bvdocs/appnotes/xapp660.zip>

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
01/13/03	1.0	Initial Xilinx release.
08/21/03	2.0	Major revision describing new module interface. Simplified implementation flow.

Date	Version	Revision
11/25/03	2.1	<ul style="list-style-type: none">Section “XC2VP2 Design Specifics”: Added requirement that XST be used to synthesize XC2VP2 version of MGT reconfiguration module.Replaced existing design files archive with updated files.Relocated design files archive from ftp server to web server in compliance with current practice.
02/04/04	2.2	<ul style="list-style-type: none">Section “ISE Design Example”: Added three implementation notes addressing potential conflicts with the ICAP interface.Reference design files updated to support larger Virtex-II Pro devices (XC2VP70 and up).