

Introduction

The Virtex™-5 GTP Aurora core implements the Aurora protocol using the high-speed serial GTP transceivers in Virtex-5 LXT and SXT devices. The core can use up to 24 GTP transceivers running at any supported line rate to provide a low cost, general purpose, data channel with throughput range from 500 Mb/s to 90 Gb/s.

Aurora is a scalable, lightweight, link-layer protocol for high-speed serial communication. The protocol is open and can be implemented by Aurora protocol licensees using any technology. The protocol is typically used in applications requiring simple, low cost, high rate, data channels.

The CORE Generator™ tool produces source code for Aurora cores with variable datapath width. The cores can be simplex or full-duplex, and feature one of two simple user interfaces and optional flow control.

Aurora cores are verified for protocol compliance using the Aurora bus functional model (BFM) and an array of automated hardware and simulation tests. The core comes with an example design. Several reference designs are available that show the use of the Aurora core and provide additional plug-in features.

Features

- General purpose data channels with throughput range from 500 Mb/s to 90 Gb/s
- Supports line rates 500 Mb/s to 3.75 Gb/s
- Supports up to 24 Virtex-5 RocketIO™ GTP transceivers
- Aurora v2.0 compliant (8B/10B encoding)
- Low resource cost (see [Resource Utilization](#))
- Easy-to-use framing and flow control
- Automatically initializes and maintains the channel
- Full-duplex or simplex operation
- LocalLink (framing) or streaming user interface

| LogiCORE Facts | | | | |
|-----------------------------|--|------|-----|------------|
| Core Specifics | | | | |
| Supported Device Family | Virtex-5 LXT/SXT ¹ | | | |
| Resources Used | I/O | LUTs | FFs | Block RAMs |
| | Varies with channel size See " Resource Utilization " on page 7 | | | 0 |
| Special Features | Open source; Core is free | | | |
| Provided with Core | | | | |
| Documentation | Product Specification User Guide Getting Started Guide | | | |
| Design File Formats | Verilog and VHDL | | | |
| Constraints File | .ucf (user constraints file) | | | |
| Verification | Aurora Bus Functional Model | | | |
| Design Tool Requirements | | | | |
| Xilinx Implementation Tools | ISE 9.2i ² | | | |
| Verification | ModelSim 6.1e | | | |
| Simulation | ModelSim 6.1e | | | |
| Synthesis | Synplify 8.8 XST 9.2i | | | |
| Support | | | | |
| Provided by Xilinx, Inc. | | | | |

1. For more information on the Virtex-5 LXT/SXT platform, see [DS100: Virtex-5 Family Overview](#)
2. ISE Service Packs can be downloaded at <http://www.xilinx.com/support/download.htm>

© 2006-2007 Xilinx, Inc. All rights reserved. XILINX, the Xilinx logo, and other designated brands included herein are trademarks of Xilinx, Inc. All other trademarks are the property of their respective owners. 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.

Functional Overview

Aurora is a lightweight, serial communications protocol for multi-gigabit links. It is used to transfer data between devices using one or many GTP transceivers. Connections can be *full-duplex* (data in both directions) or *simplex*.

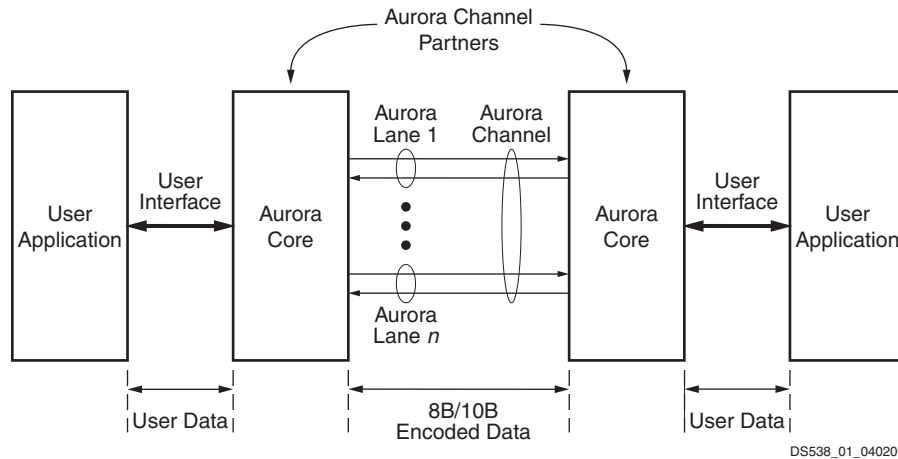


Figure 1: Aurora Channel Overview

Aurora cores automatically initialize a channel when they are connected to an Aurora channel partner. After initialization, applications can pass data freely across the channel as *frames* or *streams* of data. Aurora *frames* can be any size, and can be interrupted at any time. Gaps between valid data bytes are automatically filled with *idles* to maintain lock and prevent excessive electromagnetic interference. *Flow control* is optional in Aurora, and can be used to reduce the rate of incoming data, or to send brief, high-priority messages through the channel.

Streams are implemented in Aurora as a single, unending frame. Whenever data is not being transmitted, idles are transmitted to keep the link alive. The Aurora core detects single-bit, and most multi-bit errors using 8B/10B coding rules. Excessive bit errors, disconnections, or equipment failures cause the core to reset and attempt to initialize a new channel.

Applications

Aurora cores can be used in a wide variety of applications because of their low resource cost, scalable throughput, and flexible data interface. Examples of Aurora core applications include:

- **Chip-to-chip links:** Replacing parallel connections between chips with high-speed serial connections can significantly reduce the number of traces and layers required on a PCB. The Aurora core provides the logic needed to use GTP transceivers, with minimal FPGA resource cost.
- **Board-to-board and backplane links:** Aurora uses standard 8B/10B encoding, making it compatible with many existing hardware standards for cables and backplanes. Aurora can be scaled, both in line rate and channel width, to allow inexpensive legacy hardware to be used in new, high-performance systems.
- **One-way connections:** In some applications there is no need for a high-speed back channel. The Aurora simplex protocol provides several ways to perform unidirectional channel initialization, making it possible to use the GTP transceivers when a back channel is not available, and to reduce costs due to unused full-duplex resources.

- **ASIC connections:** Aurora is not limited to FPGAs, and can be used to create scalable, high-performance links between programmable logic and high-performance ASICs. The simplicity of the Aurora protocol leads to low resource costs in ASICs as well as in FPGAs, and design resources like the Aurora bus functional model with automated compliance testing make it easy to get an Aurora connection up and running.

Functional Description

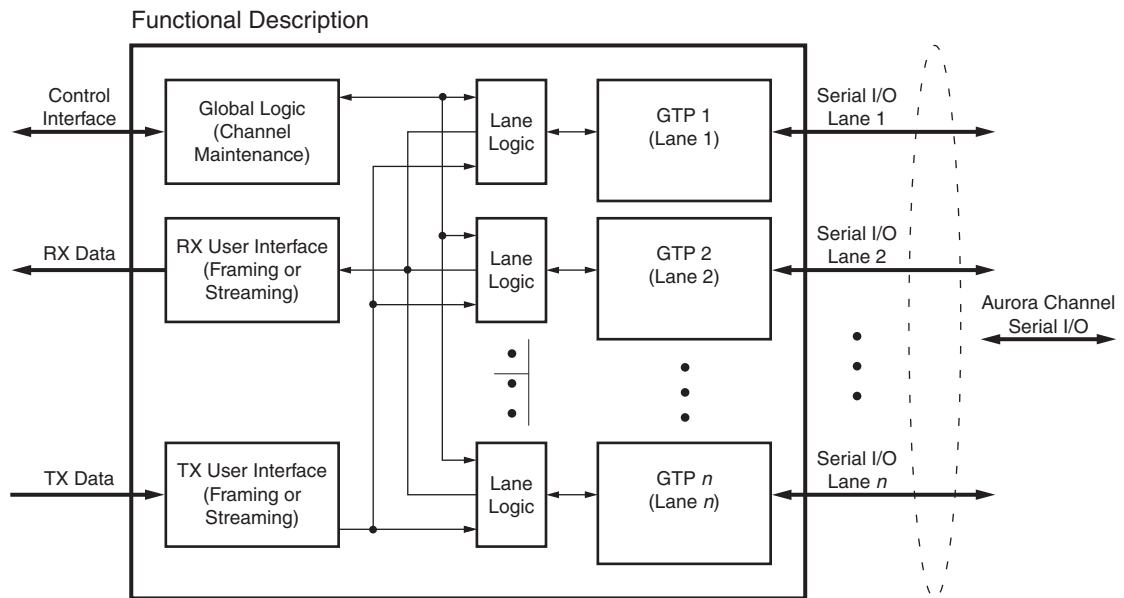


Figure 2: Aurora Core Block Diagram

Figure 2 shows a block diagram of the implementation of the Aurora core. The major functional modules of the Aurora core are:

- **Lane logic:** Each GTP_DUAL tile is driven by an instance of the lane logic module, which initializes each individual GTP transceiver and handles the encoding and decoding of control characters and error detection.
- **Global logic:** The global logic module in each Aurora core performs the bonding and verification phases of channel initialization. While the channel is operating, the module generates the random idle characters required by the Aurora protocol and monitors all the lane logic modules for errors.
- **RX user interface:** The RX user interface moves data from the channel to the application. Streaming data is presented using a simple stream interface equipped with a data bus and a data valid signal. Frames are presented using a standard LocalLink interface. This module also performs flow control functions.
- **TX user interface:** The TX user interface moves data from the application to the channel. A stream interface with a data valid and a ready signal is used for streaming data. A standard LocalLink interface is used for data frames. The module also performs flow control TX functions. The module has an interface for controlling clock compensation (the periodic transmission of special characters to prevent errors due to small clock frequency differences between connected Aurora cores). This interface is normally driven by a standard clock compensation manager module provided with the Aurora core, but it can be turned off, or driven by custom logic to accommodate special needs.

Core Parameters

CORE Generator users can customize Aurora cores for a given application by setting the parameters for the core. [Table 1](#) (continues on next page) describes the customizable parameters.

Table 1: Core Parameters

| Parameter | Description | Values Supported by the GTP Aurora Core |
|---------------------------|--|---|
| Lanes | The number of GTP transceivers used in the channel. | From 1 to 24 GTP transceivers on the chosen device |
| Lane Width | The GTP transceivers in the core are set to use 2-byte SERDES. For a given line rate, this 2-byte mode (vs. 1-byte mode) has the effect of reducing the required clock rate of application logic connected to the Aurora core. | 2 bytes |
| Device | The type of FGPA used to implement the design. | Virtex-5 LXT/SXT devices with RocketIO GTP transceivers |
| HDL | The language in which the source code for the core will be generated. | Verilog or VHDL |
| Direction | The type of channel to be generated by the CORE Generator tool. Can be full-duplex, simplex in the TX direction, simplex in the RX direction, or two separate simplex modules (one TX and one RX) sharing the same GTP transceivers. | Full-Duplex Simplex-TX Simplex-RX Simplex-Both |
| Flow Control | Enables optional Aurora flow control. There are two types: Native Flow Control (NFC): NFC allows full-duplex receivers to control the rate of incoming data. Completion mode NFC forces idles when frames are complete. Immediate mode NFC forces idles as soon as the flow control message arrives. User Flow Control (UFC): UFC allows applications to send each other brief high priority messages through the channel. | None NFC Immediate NFC Completion UFC UFC and NFC Immediate UFC and NFC Completion |
| Interface | The user can specify one of two types of interfaces: Framing: The framing user interface is LocalLink compliant. After initialization, it allows framed data to be sent across the Aurora channel. Framing interface cores tend to be larger because of their comprehensive word alignment and control character stripping logic. Streaming: The streaming user interface allows users to start a single, infinite frame. After initialization, the user writes words to the frame using a simple register style interface that has a data valid signal. | Framing (LocalLink) Streaming |
| Line Rate | The line rate for Aurora cores that use GTP transceivers can be set from 500 Mb/s to 3.75 Gb/s using the CORE Generator tool. See the <i>Virtex-5 GTP Aurora v2.8 User Guide</i> for detailed instructions. | 500 Mb/s to 3.75 Gb/s |
| Reference Clock Frequency | The CORE Generator tool accepts parameters to set the reference clock rate for Virtex-5 devices. See the <i>Virtex-5 GTP Aurora v2.8 User Guide</i> for detailed instructions. | A selection of legal rates based on the selected line rate and available clock multipliers in the Virtex-5 GTP transceivers |

Table 1: Core Parameters (Continued)

| Parameter | Description | Values Supported by the GTP Aurora Core |
|-----------------|---|---|
| Reference Clock | GTP transceivers can be fed a reference clock from a variety of dedicated and non-dedicated clock networks. See the <i>Virtex-5 GTP Aurora v2.8 User Guide</i> for instructions to select the best reference clock network for a given application. | REFCLK and GREFCLK |
| GTP Placement | The CORE Generator tool provides a graphical interface that allows users to assign lanes to specific GTP transceivers. The <i>Virtex-5 GTP Aurora v2.8 User Guide</i> includes guidelines for placing GTP transceiver for best timing results. | Any combination of GTP transceivers can be selected |

Core Interfaces

The parameters used to generate each Aurora core determine the interfaces available (Figure 3) for that specific core. The Aurora cores have three to six interfaces:

- "User Interface" on page 6
- "User Flow Control Interface" on page 6
- "Native Flow Control Interface" on page 6
- "GTP Interface" on page 6
- "Clock Interface" on page 6
- "Clock Compensation Interface" on page 6

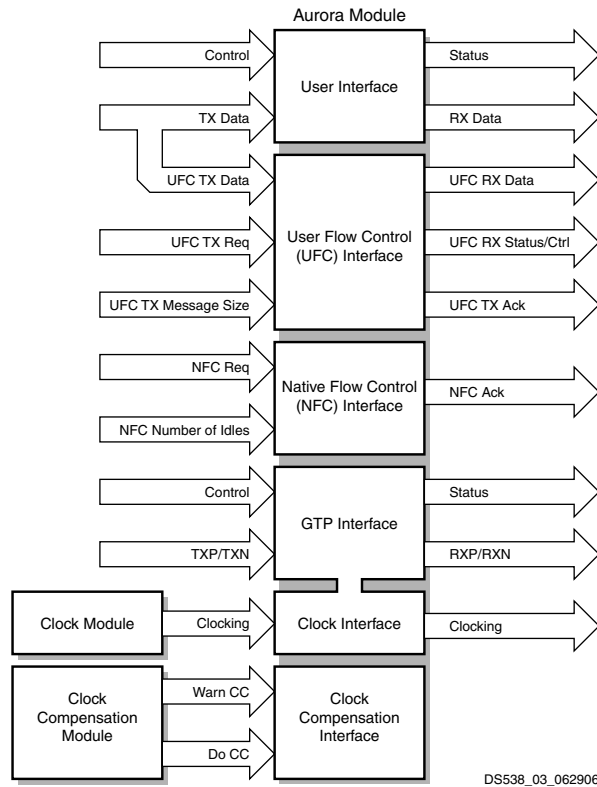


Figure 3: Top-Level Interface

User Interface

This interface includes all the ports needed to read and write *streaming* or *framed* data to and from the Aurora core. LocalLink ports are used if the Aurora core is generated with a framing interface; for streaming modules, the interface consists of a simple set of data ports and data_valid ports. Full-duplex cores include ports for both transmit and receive; simplex cores use only the ports they require to send data in the direction they support. The width of the data ports in all interfaces depends on the number of GTP transceivers in the core, and on the width selected for the transceivers.

User Flow Control Interface

If the core is generated with user flow control (UFC) enabled, a UFC interface is created. The TX side of the UFC interface consists of a request and an acknowledge port that are used to start a UFC message, and a port to specify the length of the message. The user supplies the message data to the data port of the user interface; immediately after a UFC request is acknowledged, the user interface indicates it is no longer ready for normal data, thereby allowing UFC data to be written to the data port.

The RX side of the UFC interface consists of a set of LocalLink ports that allows the UFC message to be read as a frame. Full-duplex modules include both TX and RX UFC ports; simplex modules retain only the interface they need to send data in the direction they support.

Native Flow Control Interface

If the core is generated with native flow control (NFC) enabled, an NFC interface is created. This interface includes a request and an acknowledge port that are used to send NFC messages, and a 4-bit port to specify the number of idle cycles requested.

GTP Interface

This interface includes the serial I/O ports of the GTP transceivers, and the control and status ports of the Aurora core. This interface is the user's access to control functions such as reset, loopback, and powerdown. Status information about the state of the channel, and error information is also available here.

Clock Interface

This interface is most critical for correct Aurora core operation. The clock interface has ports for the reference clocks that drive the GTP transceivers, and ports for the parallel clocks that the Aurora core shares with application logic.

Clock Compensation Interface

This interface is included in modules that transmit data, and is used to manage clock compensation. Whenever the DO_CC port is driven high, the core stops the flow of data and flow control messages, then sends clock compensation sequences. For modules with UFC, the WARN_CC port prevents UFC messages and CC sequences from colliding. Each Aurora core is accompanied by a clock compensation management module that is used to drive the clock compensation interface in accordance with the *Aurora Protocol Specification*. When the same physical clock is used on both sides of the channel, DO_CC should be tied low.

Resource Utilization

Table 2 through Table 5 show the number of look-up tables (LUTs) and flip-flops (FFs) used in selected Aurora modules. The Aurora core is also available in configurations not shown in the tables; the estimated resource usage for these other modules can be extrapolated from the tables. These tables do not include the additional resource usage for flow control. These tables include the additional resource usage for the example design.

Table 2: Virtex-5 LXT Resource Usage for Streaming

| Virtex-5 LXT | | | Streaming | | | |
|--------------|------------|---------------|-------------|---------|---------|------|
| Lanes | Lane Width | Resource Type | Duplex | Simplex | | |
| | | | Full-Duplex | TX Only | RX Only | Both |
| 1 | 2 | LUTs | 242 | 96 | 140 | 236 |
| | | FFs | 275 | 141 | 152 | 287 |
| 2 | 2 | LUTs | 440 | 143 | 277 | 417 |
| | | FFs | 519 | 246 | 300 | 538 |
| 4 | 2 | LUTs | 788 | 230 | 508 | 733 |
| | | FFs | 965 | 444 | 552 | 988 |
| 8 | 2 | LUTs | 1495 | 788 | 979 | 1378 |
| | | FFs | 1865 | 965 | 1058 | 1890 |
| 16 | 2 | LUTs | 2898 | 758 | 1913 | 2653 |
| | | FFs | 3657 | 1633 | 2062 | 3686 |

Table 3: Virtex-5 LXT Resource Usage for Framing

| Virtex-5 LXT | | | Framing | | | |
|--------------|------------|---------------|-------------|---------|---------|------|
| Lanes | Lane Width | Resource Type | Duplex | Simplex | | |
| | | | Full-Duplex | TX Only | RX Only | Both |
| 1 | 2 | LUTs | 344 | 201 | 157 | 328 |
| | | FFs | 382 | 177 | 200 | 395 |
| 2 | 2 | LUTs | 666 | 243 | 398 | 639 |
| | | FFs | 751 | 324 | 453 | 769 |
| 4 | 2 | LUTs | 1200 | 369 | 779 | 1147 |
| | | FFs | 1449 | 636 | 845 | 1473 |
| 8 | 2 | LUTs | 2609 | 619 | 1874 | 2486 |
| | | FFs | 2779 | 1169 | 1663 | 2826 |
| 16 | 2 | LUTs | 6607 | 1109 | 5258 | 6348 |
| | | FFs | 6072 | 2233 | 3877 | 6101 |

Table 4: Virtex-5 SXT Resource Usage for Streaming

| Virtex-5 SXT | | | Streaming | | | |
|--------------|------------|---------------|-------------|---------|---------|------|
| Lanes | Lane Width | Resource Type | Duplex | Simplex | | |
| | | | Full-Duplex | TX Only | RX Only | Both |
| 1 | 2 | LUTs | 241 | 94 | 142 | 233 |
| | | FFs | 275 | 140 | 149 | 283 |
| 2 | 2 | LUTs | 441 | 142 | 280 | 419 |
| | | FFs | 519 | 243 | 294 | 531 |
| 4 | 2 | LUTs | 795 | 230 | 514 | 736 |
| | | FFs | 965 | 441 | 542 | 977 |
| 8 | 2 | LUTs | 1492 | 397 | 968 | 1357 |
| | | FFs | 1865 | 837 | 1046 | 1877 |
| 16 | 2 | LUTs | 2888 | 756 | 1884 | 2598 |
| | | FFs | 3657 | 1630 | 2046 | 3670 |

Table 5: Virtex-5 SXT Resource Usage for Framing

| Virtex-5 SXT | | | Framing | | | |
|--------------|------------|---------------|-------------|---------|---------|------|
| Lanes | Lane Width | Resource Type | Duplex | Simplex | | |
| | | | Full-Duplex | TX Only | RX Only | Both |
| 1 | 2 | LUTs | 344 | 200 | 156 | 329 |
| | | FFs | 382 | 179 | 197 | 391 |
| 2 | 2 | LUTs | 657 | 243 | 396 | 630 |
| | | FFs | 751 | 322 | 447 | 763 |
| 4 | 2 | LUTs | 1203 | 373 | 782 | 1140 |
| | | FFs | 1442 | 627 | 833 | 1454 |
| 8 | 2 | LUTs | 2583 | 617 | 1862 | 2475 |
| | | FFs | 2763 | 1159 | 1623 | 2776 |
| 16 | 2 | LUTs | 6541 | 1124 | 5189 | 6275 |
| | | FFs | 6064 | 2224 | 3859 | 6076 |

Performance

The GTP Aurora cores shown in [Table 2 on page 7](#) and [Table 3 on page 7](#) run at 190.625 MHz in devices with speed grades ranging from -1 to -3.

Verification

The Aurora core is verified using the Aurora BFM and proprietary custom testbenches. An automated test system runs a series of simulation tests on the most widely used set of cores, as well as cores chosen at random. Aurora cores are also tested in hardware for functionality, performance, and reliability using Xilinx GTP demonstration boards. Aurora verification tests suites for all possible modules are continuously being modified to increase test coverage across the range of possible parameters for each individual module.

Table 6: Boards Used for Verification

| Test Boards |
|-------------|
| ML523 |
| ML505 |
| ML506 |

References

1. Xilinx Aurora Web site, <http://www.xilinx.com/aurora>
 - SP002, *Aurora Protocol Specification*
 - UG058, *Aurora Bus Functional Model User Guide*
 - UG223, *Virtex-5 GTP Aurora v2.8 Getting Started Guide*
 - UG224, *Virtex-5 GTP Aurora v2.8 User Guide*
2. **UG196**, *Virtex-5 RocketIO GTP Transceiver User Guide*

Support

For technical support, go to <http://www.xilinx.com/support>. Xilinx provides technical support for this LogiCORE product when used as described in the product documentation. Questions are routed to a team of engineers with expertise using the Aurora core.

Xilinx will provide technical support for use of this product as described in the *Virtex-5 GTP Aurora v2.8 User Guide* and the *Virtex-5 GTP Aurora v2.8 Getting Started Guide*. Xilinx cannot guarantee timing, functionality, or support of this product for designs that do not follow the guidelines in these documents, or for modifications to the source code.

Any feedback, questions, or concerns about the design can be reported to the Xilinx Technical Support or by opening a WebCase.

Ordering Information

The Aurora core is provided free of charge to licensed users. The licence for the Aurora core is also free and can be obtained by visiting www.xilinx.com/aurora.

There are three steps required to obtain the core:

1. Install Xilinx ISE 9.2i SP3. See the [ISE product page](#) for instructions if ISE software is not already installed.
2. Install Xilinx ISE 9.2i IP Update 2 to add version 2.8 of the GTP Aurora core to the list of cores available in the Core Selection window in the CORE Generator tool. Instructions for this step are available in the *Virtex-5 GTP Aurora v2.8 Getting Started Guide*.
3. Electronically sign the Aurora Core License Agreement to obtain a license file for the Aurora core. Instructions for this step and the link to the page with the license and the CORE Generator license file are also at www.xilinx.com/aurora. You must be a registered user on www.xilinx.com to sign the license.

Revision History

| Date | Version | Revision |
|----------|---------|--|
| 11/30/06 | 1.1 | Initial Xilinx release. |
| 03/01/07 | 1.2 | LogiCORE GTP Aurora v2.6 release. |
| 05/17/07 | 1.3 | LogiCORE GTP Aurora v2.7 release. Added Virtex-5 SXT resource usage tables (Table 4 and Table 5). |
| 10/10/07 | 1.4 | LogiCORE GTP Aurora v2.8 release. |