# TEST AND VERIFICATION REPORT

# EVLA Correlator Chip Gate-level Verification Report

RFS Document: **A25082N0004**

Revision: 1.0

B.Carlson, December 22, 2005

*National Research Council Canada*
*Herzberg Institute of Astrophysics*
*Dominion Radio Astrophysical Observatory*

*P.O. Box 248, 717 White Lake Rd*
*Penticton, B.C., Canada*
*V2A 6J9*

# Table of Contents

# List of Figures

# List of Tables

# 1 Revision History

| Revision | Date | Changes/Notes | Author |
|---|---|---|---|
| PRELIM DRAFT | Dec. 16, 2005 | PRELIM DRAFT for ISine review | B. Carlson |
| DRAFT | Dec. 21, 2005 | DRAFT release for final review by ISine and NRC | B. Carlson |
| 1.0 | Dec. 22, 2005 | V1.0 release. This constitutes sign-off of the design for prototype fabrication. | B. Carlson |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

## 2 Executive Summary

The gate-level netlist and timing file contained in the tar'd gzip'd file "**corr_chip_1211.tar"**(file size 150272000 bytes) delivered by the contractor ISine (formerly known as Innotech) on December 12, 2005 **passes all verification tests**. This tar file contains the gate-level netlist and both slow and fast timing model files as follows:

- **corr_chip_full_opt7_rename_flat_fill_1211.vg** – 21805249 bytes.
- **corr_chip_full_opt7_rename_flat_fill_1211_slow.sdf** – 1179711436 bytes.
- **corr_chip_full_opt7_rename_flat_fill_1211_fast.sdf** – 1179711434 bytes.

The RTL model of the chip, contained in the July 28, 2005 release file "**EVLA_Corr_Chip_25082_Release_HDLFiles_Jul28-05.zip"** formed the reference against which the functionality of the chip is compared to verify correctness. The functionality of the chip is defined in [1].

While 100% test coverage was not possible due to excessive simulation times, it is believed that very good test coverage has been achieved with the 21 test cases, and approximately 1.3 msec of simulation time utilized. A detailed verification matrix is included in this report starting on page 15.

Gate-level simulations of the netlist with both the slow and fast timing model were performed at a simulation frequency of 312.5 MHz. Since the chip normally operates at 256 MHz, this test demonstrates a minimum of 22% timing margin over the required operating frequency of 256 MHz. This margin indicates that the chip could possibly operate as low at 0.9 V core and still meeting timing at 256 MHz.

Power dissipation for the chip is estimated at **4.04 W** at an operating frequency of 256 MHz, and 1.02 V. If operation at 0.9 V core is possible, the power dissipation could be as low as 3.3 W. Refer to section 9 for more detailed information on how this number is arrived at.

The gate-level design, represented by the logical and timing model contained in the file **corr_chip_1211.tar** is qualified for the next stage of development—prototype silicon fabrication. Note that this is subject to the chip design meeting input timing parameters according to Table 8-2. Input timing tolerance of the chip has not been probed or rigorously verified by testing—input timing parameters are supplied by ISine and included verbatim in this report. Additionally, chip PLL jitter tolerance, attenuation, and generation were not tested. It is assumed and understood that the PLL attenuates jitter so that it is possible to operate the chip in its intended environment of an 8x8 array, where one chip's PLL generates a clock that feeds the next chip, for a maximum of 8 re-generations in the daisy chain. The PLL's absolute long-term or cycle-to-cycle jitter is not important, only its ability to properly maintain synchronization to input data is of concern.

# 3 Introduction

This document reports the methodology and test results for verification of the gate-level chip model for the EVLA correlator chip. Successful verification of the correctness of the gate-level model, including effects of timing, is the final step before sign-off of the design for prototype silicon fabrication.

Verification entails comparison of the output produced from simulation of the gate-level model of the chip with the output produced from simulation of the original source RTL model on a bit-exact basis. For both the gate-level and RTL models, the chip is treated as a black box; it is stimulated in the way it would be in an actual circuit, and its output is captured in one or more ASCII files for analysis. The files are compared to verify that they match.

Additionally, the output of the gate-level simulation is observed using the simulation wave/timing window to determine output timing parameters and daisy-chained output functionality. X/Y input timing requirements are checked with a separate test bench that consists of two gate-level models—the first model being used to stimulate the second model. This runs very slowly and was used to demonstrate that the chips can be daisy-chained.

# 4 Overview

Verification of the gate-level netlist with timing is the final step in correlator chip design testing before committing to silicon. The process of verification ensures that the detailed gate-level model of the chip that will be used to build the wafer mask, including the effects of real delays, is "correct". Correctness is defined to mean that the chip's gate-level model, treated as a black box, produces the same results as the RTL model when both are stimulated in the same way. The RTL model is the logical design written in synthesizable Verilog code from which the gate-level model, through a process of synthesis and place-and-route, is developed. Figure 4-1 is a simplified flow diagram of the verification process.

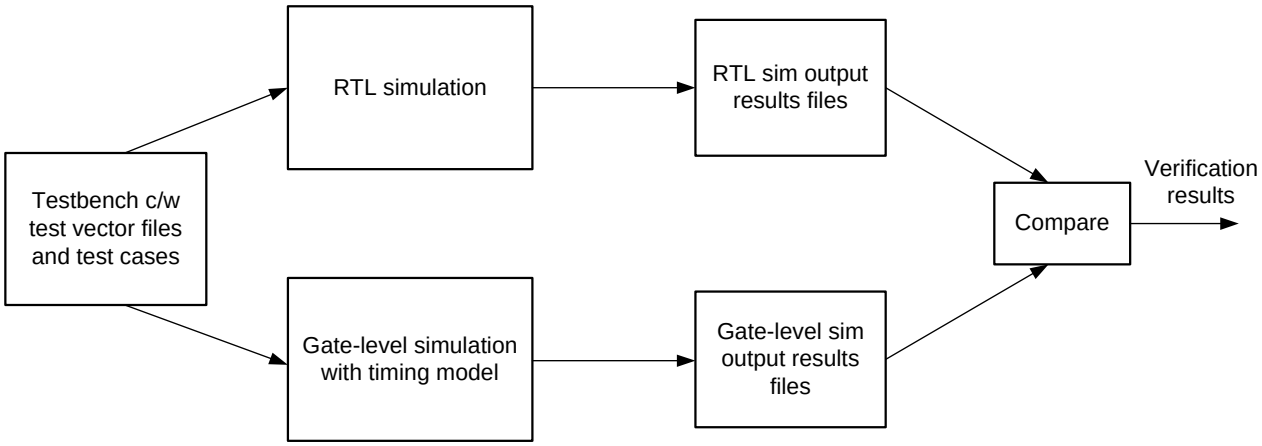


**Figure 4-1 Simplified flow diagram of the gate-level verification process.**

It is, in principle, possible to develop a series of tests that ensures that the gate-level model of the chip functions identically to the RTL model of the chip for any configuration and series of test vectors. In practice, however, it is difficult to do this due to the shear number of test cases and series of test vectors that would be required, and due to the excessive run times that doing so would require. Since the gate-level model is "smashed" down into standard models of low-level gates that will actually be implemented in silicon, run times for gate-level simulations are excessive—typically a factor of 10 slower than the RTL model. For the EVLA correlator chip, and running on the fastest computer available (and affordable), the gate-level model runs at a rate of about 160 nsec/minute. That is, it takes 1 minute to simulate each 160 nsec of chip time.

For the reasons mentioned, it is not possible to completely and exhaustively verify that the gate-level model matches the functionality of the RTL model in all possible configurations and under all possible conditions. It is therefore important to choose the test cases used for verification so that they maximize test coverage while they minimize, or at least keep manageable, the time to run the simulation. A detailed test coverage matrix is included in this document in section 6 to demonstrate that adequate test coverage is obtained, and the test cases chosen for verification have a manageable run time that is ≤ ~1 week.

# 5 Test Methodology

*Verification* of the gate-level netlist is accomplished by performing a bit-by-bit comparison of the output of the gate-level simulation with the output of the RTL simulation, given the same stimulus. The source RTL design has been *validated* by extensive testing as reported in [2] and [3]. Thus, the RTL design forms the reference against which the gate-level model is compared. For verification purposes the actual content of the output is not that important, rather the functionality that was stimulated to produce the output is the important thing.

The Verilog test bench program can put the chip into a suite of desired configurations, each configuration being a "test case". For each test case, the chip is stimulated in a desired way, and output from the chip is captured in a couple of ASCII files. The test bench is designed such that it allows the definition and execution of any number of test cases. Each test case consists of a chip configuration via the microprocessor interface (i.e. MCB), and a Verilog description of how the chip is to be stimulated. Test cases are designed in such a way that as a group they provide as much test coverage as possible within a reasonable simulation time. The output of the chip consists of "lag frames" on the LCI saved in an ASCII "`lag_framesXX.txt`" file, and a final dump of all microprocessor control/status registers in an ASCII "`mcb_regXX.txt`" file, where "`XX`" is the test case number. All test cases can be run with one simulation run, and the chip is completely reset and re-initialized for each test case so that deterministic results can be obtained.

The Mentor Graphics simulation program Modelsim SE (running Linux on NRC's dual AMD Opteron machine "evladual" at DRAO with 16 Gbytes of memory specifically purchased for this purpose) was used to perform the simulations. A dedicated account called "simuser" was built on this machine for verification simulation. The SE version of Modelsim is needed and must run on Linux for performance and memory reasons having to do with the nature of the gate-level netlist as opposed to the RTL design. The RTL design describes the chip in a relatively high-level fashion and it requires relatively little memory and computing power to run the simulations within a reasonable period of time. The gate-level design is a low-level description of the chip and includes modeled timing delays of gates, flip-flops, and wires. The gate-level design thus "smashes" the design into its smallest components and contains at least an order of magnitude or more elements that need to be simulated. Thus, the gate-level design simulation runs much more slowly than the RTL design and uses much more memory. For this chip, the gate-level simulation requires about 4.8 Gbytes of core memory, and runs at a rate of about 160 nsec/minute—a rate that is at least 10X slower than the RTL simulation.

Modelsim SE has the advertised ability to optimize the compilation of the gate-level netlist to speed up simulations. It does this with the "vopt" compile option that, supposedly, collapses logic and timing into bigger chunks that run faster. When this is done, nodes within the "chunk" cannot be probed anymore. Thus, one gains in speed, but loses in the ability to probe internal points of the chip if something doesn't work. The gate-level model was compiled and run with the "vopt" option, but it was found that the

chip no longer functioned properly. This was brought to Mentor Graphics' attention, but there was never any satisfactory answer provided, and the issue was never vigorously pursued since a simulation time of 5 days seemed to be acceptable for verification run time without the "vopt" option.

Verification test cases were built on the foundation of the test cases developed for chip validation. A handful (21) of validation test cases were selected for verification. These test cases were, in some cases, modified slightly from the originals to provide better test coverage and minimize simulation run time. Refer to section 6 for a complete listing of the verification test coverage matrix. Also, extensive testing with preliminary gate-level netlists was done to ensure that the gate-level model of the chip is properly initialized to prevent inadvertent don't cares from propagating to the output (a side-effect when signals cross clock domains that is not a problem with actual chip silicon), and to ensure that functions such as JTAG, that are not part of normal chip operation or tested with simulation, don't factor into or affect normal chip operation.

For a complete verification run on NRC's dual AMD Opteron machine, it takes about 5-7 days. Since two verification runs are required—one for a slow timing model and one for a fast timing model—an additional computer and Modelsim license located at Jodrell Bank Observatory in the UK were used (with the grateful assistance of Jodrell Bank). This computer was set up in an identical fashion to the computer at DRAO (i.e. simuser account and directory structure), and run remotely in the same environment. This computer takes about 7-8 days for a complete verification run. Thus, it takes a total of 7-8 days to run the simulations for both timing models.

The output ASCII files from the RTL simulation and the gate-level simulation were compared with each other using a script that invokes the Unix/Linux "diff" command once for each pair of files that are to be compared. The differenced results, for each pair of files, are saved to a unique file. Thus, a long listing "ll" of the directory containing these "diff output" files very quickly shows which comparison files are different as they have a non-zero length.

In addition, the software program "compare" [4] was used to double-check the results in cases where frames produced by the gate-level model are not identical or in the same order as that produced by the RTL model for a couple of test cases. This is an explainable, expected, and acceptable result and is explained fully in section 7.

A description of the important directories and files in the "simuser" environment on the Opteron machine "evladual" is as follows. The environment is set up so that it is easy to start and run simulations, and to keep all of the various input and output files in well-established and hierarchical locations.

- **`CorrChip_RTL`** – The directory containing the RTL source code and Modelsim scripts.
    - **`vsrc`** – sub-directory containing the Verilog RTL ASCII source files. These files are generated by HDL designer from the design computer (bcarlson on "widar3") operating on Windows.

- **rtl_tc300MHz.do** – The Modelsim do file that compiles all of the source files, the testbench files, the testcases, and loads and starts the simulation. This defines a variable that tells the test bench that the simulation is to run at 312.5 MHz. A similar do file "rtl_tc.do" is used to run the simulation at 256 MHz. 312.5 MHz is used for verification to ensure additional timing margin of the design over the required 256 MHz.
- **TSMC_PD13A12** – This is a PLL initialization file for the PLL. The RTL simulation uses the TSMC PLL model, rather than an idealized model, to ensure that there are no inadvertent differences in the sim results due to some small delays in the output of the PLL[1].
- **corrchip_defines.v** – This is a Verilog file that is part of the RTL code. It defines cmam_rc module pipeline delays and readout structure.

- **CorrChip_GL** – The directory containing the gate-level model and Modelsim scripts.
  - **slow_tc300MHz.do** – This is the modelsim .do file that contains the compile and run script for the gate-level sim slow timing model for 312.5 MHz operation.
  - **fast_tc300MHz.do** – This is the Modelsim .do file that contains the compile and run script for the gate-level sim fast timing model for 312.5 MHz operation.
  - **force.do** – This is a Modelsim .do file that contains gate-level chip initialization to put the JTAG FSM into a known state, and to hold select internal states of the chip at known values until transient don't care states, resulting from MCB configuration, have disappeared.
  - **TSMC_PD13A12** – This is a PLL initialization file for the PLL.
  - **power_tc300MHz.do** – This is Modelsim .do file for running a simulation for power dissipation estimation via the generation of a .vcd file (Value Change Dump file). This .do file also includes the execution of the "vcd_gen.do" file.
  - **slow_daisy_tc.do** – This is a Modelsim .do file for checking X/Y input timing by using two instances of the gate-level model in a separate test bench.
  - **vsrc** – This sub-directory contains the gate level model .vg file, the slow and fast .sdf timing models, and several Verilog TSMC library files provided by the correlator chip place-and-route contractor ISine.

[1] Actually, the use of the TSMC PLL model is historical since at one point it was thought it was needed to eliminate some gate-level chip initialization problems, and it was never changed back to simple and ideal RTL model.

- **testbench_files** – The directory containing the test bench files.
    - **corr_chip_tb.v** – This is the top-level test bench file. It contains one instance of the "uut" correlator chip, and one instance of the tester/stimulator.
    - **corr_chip_top_tester.v** – This is the Verilog tester program that stimulates the "uut" in the test bench. It is the program that also captures chip output and creates and writes to the lag_frames and mcb_reg ASCII files.
    - **corrchip_testcases_tc.v** – This file is included in "corr_chip_top_tester.v" and defines the test cases that will be run. Variables in this file can be set to run some or all of the test cases for a particular simulation run.
    - **force_start.v** – Verilog file that forces some gate-level model internal nodes to known states to prevent propagation of transient don't cares as previously described.
    - **release_start.v** – Verilog file that releases the force gate-level model internal nodes once transient states no longer exist.
    - **corr_chip_daisy_tb.v** – Top-level test bench for daisy-chain timing testing.

- **testcases** – The directory containing the actual test case Verilog files. Each test case Verilog file contains an initialize block that initializes variables for the simulation run.
    - **tcXX_verify.v** – Each of these test case files contains initialization of variables that tell the "corr_chip_top_tester.v" file how to stimulate the chip.

- **testvectors** – The directory that contains the ".vhex" files that contain data, phase, and data valid test vectors for each input stream for each of X and Y. Each test case utilizes one or more of these files, and each file contains the vectors for one of the X or Y inputs. There are a total of 16 files in this directory. Each line of each file contains 3 ASCII characters and a newline. The first ASCII character is 0…f and is the value of a 4-bit data sample, encoded as a hexadecimal ASCII character. This gets interpreted as a 4-bit, 2's complement number. The second ASCII character is 0…f and is the value of phase for that sample, encoded as a hexadecimal ASCII character. The third character is ASCII "0" or "1", and indicates if the sample is valid "1" or invalid "0".

- **output_rtl_tc** – The destination directory for lag_frames and mcb_reg files for the RTL simulation.

- **output_gl_tc** – The destination directory for lag_frames and mcb_reg files for the gate-level simulation.

- **`diffan`** – This directory contains the "diffan" (difference analyzer) script and the program "compare". "diffan" is a simple shell script that performs an exhaustive diff between the files in the output_rtl_tc directory and the output_gl_tc directory, and puts the results for each file diff in an output file, in this directory, appropriately named. "compare" is a more intelligent diff [4].

With this structure in place, it is easy to run a gate-level or an RTL simulation. To run an RTL simulation, start Modelsim, open a new window, change to the "`CorrChip_RTL`" directory, and then execute the `rtl_tc300MHz.do` file (i.e. `do rtl_tc300MHz.do`). Figure 5-1 is a simplified flow diagram of a Modelsim RTL simulation session.

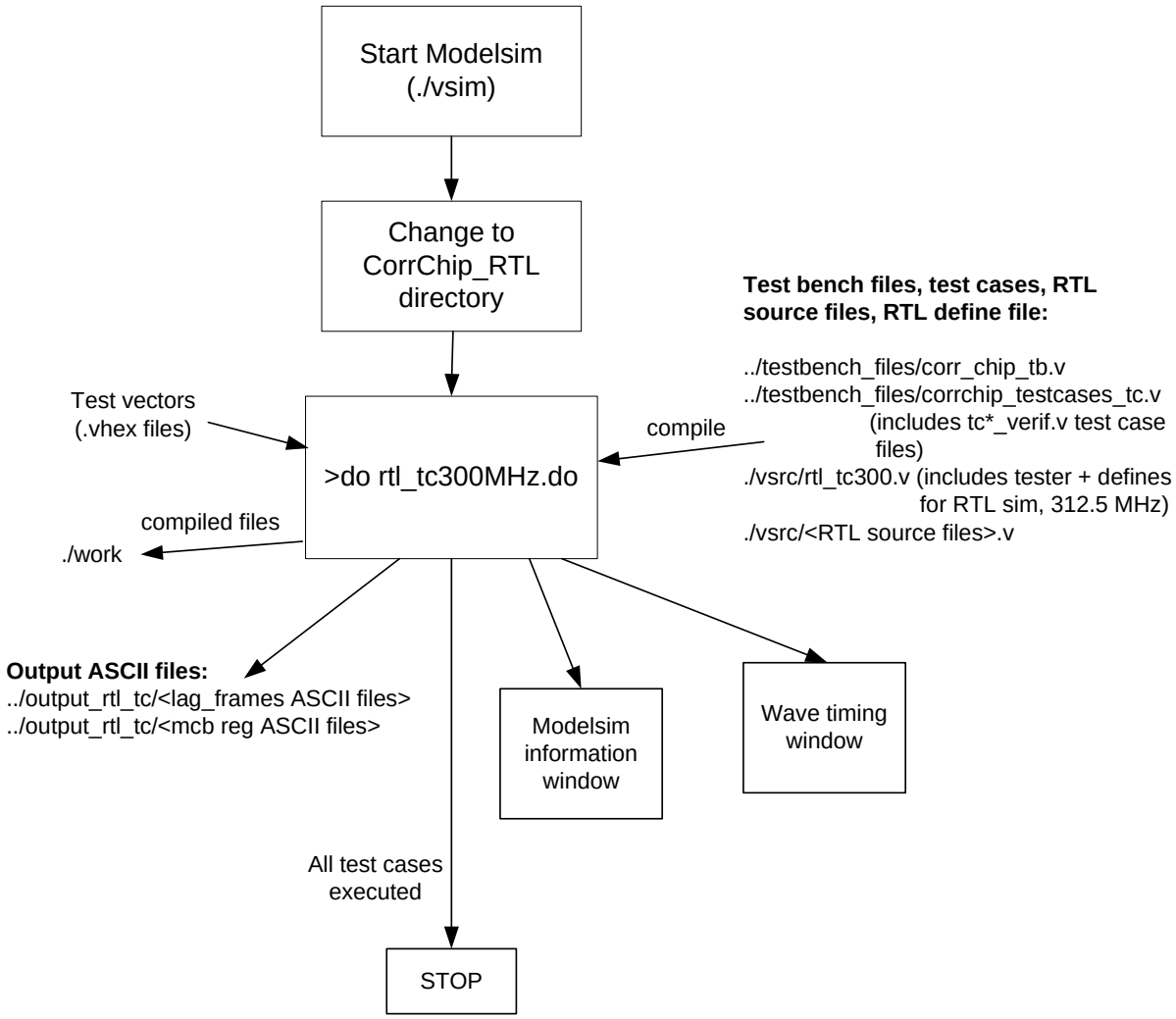


**Figure 5-1 Simplified flow diagram for a Modelsim RTL simulation session.**

When the .do file is executed, all files are compiled, loaded, the wave window is opened, and the simulation starts executing. Execution finishes and stops when all testcases that are set for execution (in the "`corrchip_testcases_tc.v`" file) have completed. The results, consisting of lag_frames and mcb_reg ASCII files, one of each for each test case, are in the `output_rtl_tc` directory. Similarly for a gate-level simulation, except that it

executes at about 1/10$^{th}$ of the speed of the RTL simulation. Figure 5-2 is a simplified flow diagram of a gate-level simulation.

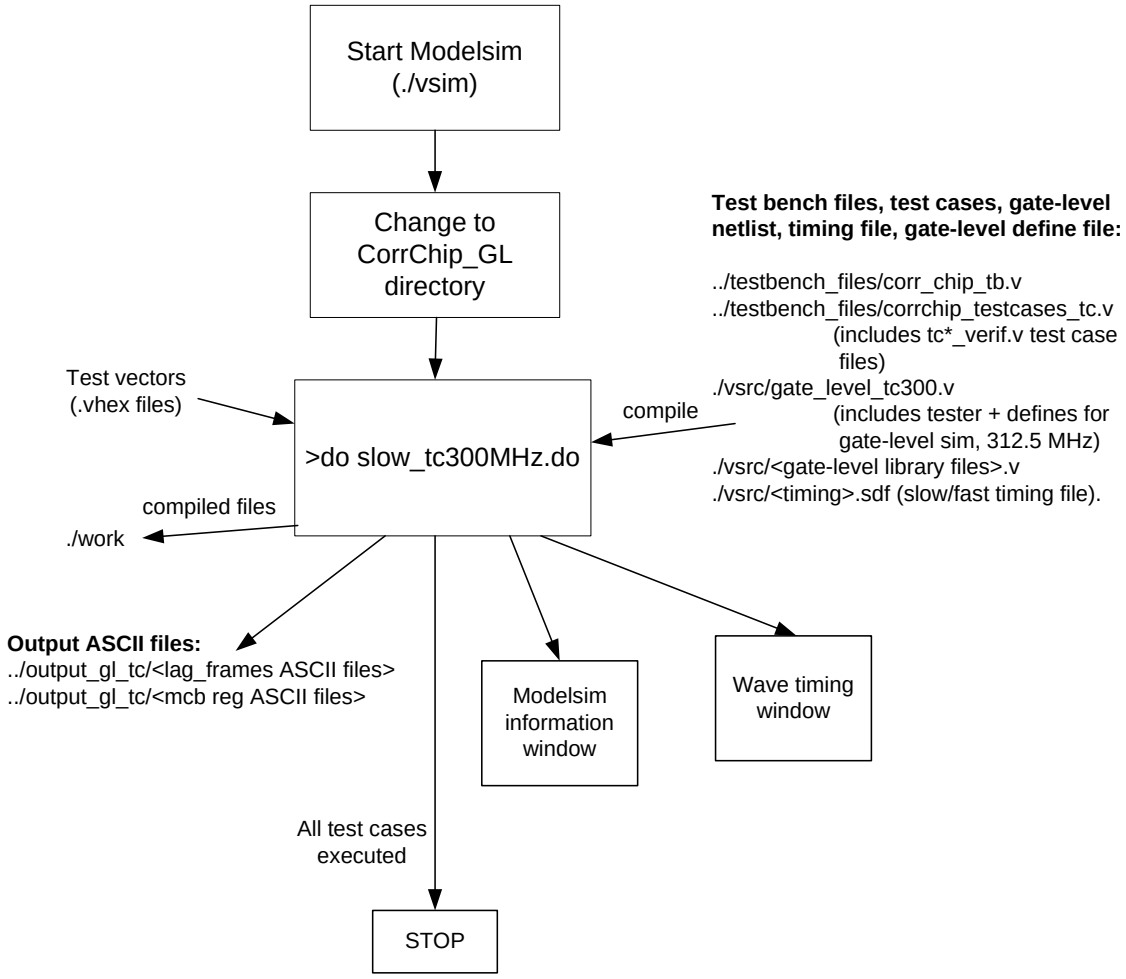


**Figure 5-2 Simplified flow diagram for a Modelsim gate-level simulation session.**

When simulation is complete, "`./diffan`" is run in the "diffan" directory, and the output files are examined for differences. In some cases, such as verifying timing for Table 8-1, it is necessary to examine the Modelsim wave window during or after the gate-level simulation.

# 6 Test Coverage Matrix

Table 6-1 is a simplified test coverage table that indicates, for each test case, the primary purpose(s) of the test. **Duration** is the approximate chip run time for each test case @ 256 MHz.

| Test Case | Primary Purpose of Test | Duration (usec) |
|---|---|---|
| **00** | Toggle all bits in all accumulators in "split accumulator" test mode. | 60 |
| **01** | In "split accumulator" test mode, toggle all bits including the OV bit. | 62 |
| **36** | Normal correlation at full speed, single dump, CCCs chained in pairs | 60 |
| **38** | Test vector mode (TVEN=1), no errors | 15 |
| **39** | Test vector mode (TVEN=1), some errors on some inputs | 15 |
| **40** | Test vector mode (TVEN=1), stuck input errors on "patterned" inputs | 15 |
| **41** | Dump control Cmmd capture, recirc block, phase bin, data IDs | 60 |
| **42** | Dump control Cmmd capture, recirc block, phase bin, data Ids, and corr holdoff | 60 |
| **43** | Frame abort test on LCI | 60 |
| **48** | All products, all inputs active, full speed, DESSR detection | 60 |
| **49** | All products, all inputs active, full speed, DESSR detection (other bits) | 60 |
| **50** | Detection of on-line stuck DATA or PHASE inputs | 60 |
| **51** | Detection of on-line stuck DATA or PHASE inputs (other bits) | 60 |
| **52** | SCHID_FRAME FIFO offset overflow (X/Y SCHID_SYNC detect) positive offset | 60 |
| **53** | SCHID_FRAME FIFO offset overflow (X/Y SCHID_SYNC detect) negative offset | 60 |
| **54** | Long integrations, fast dumping with overruns, all inputs active at different sample rates, and at different dump rates. All pol'n products in all CCCs. Y DUMP_EN error detects, DUMP_SYNC offsets. | 368 |
| **55** | Long test, fast dumping with overruns, all inputs active, all products, multiple IDs and dump Cmmds on inputs. | 100 |
| **56** | Autocorrelation, input 0, normal mode toggle Bit 16 of all lag accumulators | 68 |
| **62** | Autocorrelation, input 6, normal mode toggle Bit 16 of all lag accumulators | 68 |
| **67** | All inputs, all products at different sample rate, no input errors. | 60 |
| **68** | Test connection of X inputs 6 and 7 to CCQ-0; for 16 baselines/chip function. | 30 |

**Table 6-1 Test case coverage summary table.**

A detailed test coverage matrix is included in the following pages that, for each TEST CASE, lists the functionality in the correlator chip that is exercised. If the matrix entry is blank, then the functionality is not exercised. If the matrix entry is "y", then the functionality is exercised. In some cases it is appropriate to include a number or range of numbers in a particular matrix entry to indicate particular parts of the chip or number ranges that are covered. The table on page 20 indicates, for each entry, coverage summary taking all of the test cases together as a whole. As can be seen 100% coverage is obtained for most items, although in some cases it is "aggregate coverage". For aggregate coverage it means that functionality that it replicated in the chip is tested in its entirety, but not for every instance in the chip—the assumption being that synthesis of identical source RTL code will produce identical results, and that if the chip passes static timing analysis (as done by the contractor, ISine), it will work.

Upon careful examination of the matrix, a few holes in test coverage become evident. They are summarized below:

- When in normal operating mode, there is no integration that runs long enough to exercise Bit 16 of the Data Valid accumulators, thereby testing that the test mode's "split accumulator mux" operates properly in normal mode. To do this would require a 260 usec integration time. It is believed that this coverage hole is acceptable for the following reasons:
    - The accumulator source is the same module as the lag accumulators that are tested. Since lag accumulators test ok, it is highly likely that the data valid accumulators are synthesized and function identically.
    - There is no timing constraint on this part of the accumulator since it is a ripple counter.

- Not all CCC input switches are set to the "undef" state (only 25% of the CCCs are tested like this). It is thus conceivable that an untested CCC could be generating frames when set to the "undef" state, even though it is not supposed to. The fall-back position, if this happens, is that the frame can be dumped by the LTA FPGA.

- Autocorrelation is tested only for inputs 0 and 6 active. To test all of the other inputs would require considerably more simulation time (~360 usec). Functionality for all inputs has been tested during validation, and it is unlikely an error in the gate-level model. Also, autocorrelation mode is not a required mode for the EVLA, so even if it didn't function, it would not be a serious problem. Autocorrelation mode was added at the last minute to the RTL design because it was a zero-cost add-on, and extends the potential marketability/use of the chip and the board.

| Description TEST CASE | 0 | 1 | 36 | 38 | 39 | 40 | 41 | 42 | 43 |
|---|---|---|---|---|---|---|---|---|---|
| Toggle all accum bits via split accum test mode | y | y | | | | | | | |
| Toggle all accum OV bits via split accum test mode | | y | | | | | | | |
| Toggle all accum Bit 16, normal mode (accum ≥10000h) list CCC#'s test | | | | | | | | | |
| Correlate all CCCs at 256 Ms/s | y | y | y | | | | y | y | |
| All inputs, all pol'n prods, at different sample rates. | | | | | | | | | |
| Chain all CCCs together | y | y | | | | | | | |
| All CCCs corr inputs 0,1 | | | | | | | | | |
| Chain adjacent CCCs in pairs | | | y | | | | y | y | |
| Active inputs -- X | 0 | 0 | 1-7 | 0-7 | 0-7 | 0-7 | 0-7 | 0-7 | |
| Active inputs -- Y | 7 | 7 | 0-5, 7 | 0-7 | 0-7 | 0-7 | 0-7 | 0-7 | |
| CCC#s: at least 1 wrap of diff phase | all | all | 2-15 | | | | 2-15 | 2-15 | |
| Single dump | y | y | y | | | | y | y | y |
| Multiple rapid dumps, with overrun detection, dump/clear | | | | | | | | | |
| Different dump rates, different streams | | | | | | | | | |
| Correlation enable/disable (CEN bit of MCSR) | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| Correlation HOLDOFF via DUMP_EN lines | y | y | | | | | | y | |
| Phase rotation enable/disable (PhEN bit of MCSR) | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| Test vector enable/disable (TVEN bit of MCSR) | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| OVR status bit of MCSR | | | | | | | | | |
| AOV status bit of MCSR | | y | | | | | | | |
| SyncER status bit of MCSR | | | y | | | | | | |
| X/Y_DS bit of MCSR: X dumping | y | y | | | | | y | | y |
| X/Y_DS bit of MCSR: Y dumping | | | y | | | | | y | |
| CCQ#s: XP0 -- XSw0 primary input | 0 | 0 | 1-3 | | | | 0-3 | 0-3 | |
| CCQ#s: XS0 -- XSw0 secondary input | | | 0 | | | | | | |
| CCQ#s: XP1 -- XSw1 primary input | | | 0-3 | | | | 0-3 | 0-3 | |
| CCQ#s: XS1 -- XSw1 secondary input | | | | | | | | | |
| CCQ#s: YP0 -- YSw0 primary input | | | 1-3 | | | | 0-3 | 0-3 | |
| CCQ#s: YS0 -- YSw0 secondary input | | | 3 | | | | | | |
| CCQ#s: YP1 -- YSw1 primary input | 3 | 3 | 0-3 | | | | 0-3 | 0-3 | |
| CCQ#s: YS1 -- YSw1 secondary input | | | | | | | | | |
| CCC#s: X_A -- adjacent input | 1-15 | 1-15 | 1,3,5,7...,15 | | | | 1,3,5...15 | 1,3,5...15 | |
| CCC#s: X_P -- primary input | 0 | 0 | 0,4,8,12 | | | | 0,4,8,12 | 0,4,8,12 | |
| CCC#s: X_S -- secondary input | | | 2,6,10,14 | | | | 2,4,6,8 | 2,4,6,8 | |
| CCC#s: X_undef -- no connect (CCC not active) | | | | | | | | | |
| CCC#s: Y_A -- adjacent input | 0-14 | 0-14 | 0,2,4,6,...14 | | | | 0,2,4...14 | 0,2,4...14 | |
| CCC#s: Y_P -- primary input | 15 | 15 | 3,7,11,15 | | | | 3,7,11,15 | 3,7,11,15 | |
| CCC#s: Y_S --secondary input | | | 1,5,9,13 | | | | 1,5,9,13 | 1,5,9,13 | |
| CCC#s: Y_undef -- no connect | | | | | | | | | |
| Input ID capture: SID-X | 85 | 85 | 200 | | | | 0 | 200 | 200 |
| Input ID capture: SBID-X | 0 | 0 | 1-7 | | | | 0-7 | 0-7 | 0,2,4,6 |
| Input ID capture: BBID-X | 0 | 0 | 0-3, 5-7 | | | | 0-7 | 0-7 | 1,3,5,7 |
| Input ID capture: SID-Y | 170 | 170 | 100 | | | | 255 | 100 | 100 |
| Input ID capture: SBID-Y | 0 | 0 | 0-3, 5-7 | | | | 24-31 | 0-7 | 0,2,4,6 |
| Input ID capture: BBID-Y | 0 | 0 | 0, 3-7 | | | | 0-7 | 0-7 | 1,3,5,7 |
| Timestamp-0 capture: X | aaaaaaaah | 55555555h | | | | | 222333444 | | 222333444 |
| Timestamp-1 capture: X | 55555555h | aaaaaaaaah | | | | | 555666777 | | 555666777 |
| Timestamp-0 capture: Y | | | aaaaaaaah | | | | | 222333444 | |
| Timestamp-1 capture: Y | | | 55555555h | | | | | 555666777 | |
| Recirc Block capture: X | 233 | 233 | 30, 140, ... 190 | | | | 80-87 | 20, 130...190 | 233 |
| Recirc Block capture: Y | 120 | 120 | 233 | | | | 120, 130...190 | 80-87 | 20,140...180 |
| Phase bin capture: X | aaaah | 5555h | | | | | 21840-21847 | | 556666 |
| Phase bin capture: Y | | | 5555h | | | | | 43690-7 | |
| CMD capture: X | 010 | 010 | | | | | all patterns | | 010 |
| CMD capture: Y | | | 010 | | | | | all patterns | |
| On-line X/Y input error detect | | | Y-DUMP_EN, | | | | | | |
| No errors--test vector mode | | | | y | | | | | |
| Input sync error detect—test vectors | | | | | y | | | | |
| Input stuck error detect—test vectors | | | | | | y | | | |
| Schid_frame + offset sync | | | | | | | | | |
| Schid_frame – offset sync | | | | | | | | | |
| Schid_frame + offset over detect | | | | | | | | | |
| Schid_frame – offset over detect | | | | | | | | | |
| Toggle DESSR_0/7, DESSR_8/15 bits | | | | | | | | | |
| DUMP_SYNC offset detect | | | | | | | | | |
| 16 baselines/chip (X 6,7 into CCQ-0) | | | X-6 | | | | | | |
| Autocorr mode | | | | | | | | | |
| Abort frames on LCI | | | | | | | | | y |
| MCB register configuration (write) | y | y | y | y | y | y | y | y | y |
| MCB register reads | y | y | y | y | y | y | y | y | y |
| Output lag data frames | y | y | y | | | | y | y | y (abort) |
| DATA_BIAS/64 (approx # samples correlated) | 84628 | 2068 | 724 | | | | 724 | 414 | 724 |
| Nominal data valid count | 66140 | 66488 | 670 | | | | 670 | 380 | 670 |

| Description TEST CASE | 48 | 49 | 50 | 51 | 52 | 53 |
|---|---|---|---|---|---|---|
| Toggle all accum bits via split accum test mode | | | | | | |
| Toggle all accum OV bits via split accum test mode | | | | | | |
| Toggle all accum Bit 16, normal mode (accum ≥10000h) list CCC#'s tested | | | | | | |
| Correlate all CCCs at 256 Ms/s | y | y | y | y | y | y |
| All inputs, all pol'n prods, at different sample rates. | | | all prods | all prods | | |
| Chain all CCCs together | | | | | y | y |
| All CCCs corr inputs 0,1 | y | y | | | y | y |
| Chain adjacent CCCs in pairs | | | | | | |
| Active inputs -- X | 0,1 | 0,1 | 0-7 | 0-7 | 0 | 0 |
| Active inputs -- Y | 0,1 | 0,1 | 0-7 | 0-7 | 0 | 0 |
| CCC#s: at least 1 wrap of diff phase | 4,15,2,3,6,7,10,1 | all except 3 | all except 3 | all except 3 | | |
| Single dump | y | y | y | y | y | y |
| Multiple rapid dumps, with overrun detection, dump/clear | | | | | | |
| Different dump rates, different streams | 0,1 | 0,1 | | | | |
| Correlation enable/disable (CEN bit of MCSR) | 1 | 1 | 1 | 1 | 1 | 1 |
| Correlation HOLDOFF via DUMP_EN lines | | | | | | |
| Phase rotation enable/disable (PhEN bit of MCSR) | 1 | 1 | 1 | 1 | 0 | 1 |
| Test vector enable/disable (TVEN bit of MCSR) | 0 | 0 | 0 | 0 | 0 | 0 |
| OVR status bit of MCSR | | | | | | |
| AOV status bit of MCSR | | | | | | |
| SyncER status bit of MCSR | | | y | y | y | y |
| X/Y_DS bit of MCSR: X dumping | y | y | y | y | y | y |
| X/Y_DS bit of MCSR: Y dumping | | | | | | |
| CCQ#s: XP0 -- XSw0 primary input | 0 | 0 | 0-3 | 0-3 | 0 | 0 |
| CCQ#s: XS0 -- XSw0 secondary input | 1,2,3 | 1,2,3 | | | | |
| CCQ#s: XP1 -- XSw1 primary input | 0 | 0 | 0-3 | 0-3 | | |
| CCQ#s: XS1 -- XSw1 secondary input | 1,2,3 | 1,2,3 | | | | |
| CCQ#s: YP0 -- YSw0 primary input | 0 | 0 | 0-3 | 0-3 | | |
| CCQ#s: YS0 -- YSw0 secondary input | 1,2,3 | 1,2,3 | | | 3 | 3 |
| CCQ#s: YP1 -- YSw1 primary input | 0 | 0 | 0-3 | 0-3 | | |
| CCQ#s: YS1 -- YSw1 secondary input | 1,2,3 | 1,2,3 | | | | |
| CCC#s: X_A -- adjacent input | | | | | 1-15 | 1-15 |
| CCC#s: X_P -- primary input | 0,1,4,5,8,9,12,13 | 3,6,7,10,11,14,1 | 3,6,7,10,11,14,12 | 3,6,7,10,11,14,15 | 0 | 0 |
| CCC#s: X_S -- secondary input | 2,3,6,7,10,11,14,1 | 0,1,4,5,8,9,12,13 | 0,1,4,5,8,9,12,13 | 0,1,4,5,8,9,12,13 | | |
| CCC#s: X_undef -- no connect (CCC not active) | | | | | | |
| CCC#s: Y_A -- adjacent input | | | | | 0-14 | 0-14 |
| CCC#s: Y_P -- primary input | 3,2,5,6,9,10,13,14 | 2,5,6,9,10,13,15 | 0,2,4,6,8,10,12,14 | 0,2,4,6,8,10,12,14 | | |
| CCC#s: Y_S --secondary input | 1,3,4,7,8,11,12,15 | 3,4,7,8,11,12,15 | 1,3,5,7,9,11,13,15 | 1,3,5,7,9,11,13,15 | 15 | 15 |
| CCC#s: Y_undef -- no connect | | | | | | |
| Input ID capture: SID-X | 85 | 85 | 136 | 64 | 85 | 85 |
| Input ID capture: SBID-X | 0,1 | 0,1 | 0,2 | 0,1,5,4 | 0 | 0 |
| Input ID capture: BBID-X | 0,1 | 1,2 | 0,1,4,5 | 0,2 | 4 | 4 |
| Input ID capture: SID-Y | 170 | 170 | 68 | 32 | 175 | 175 |
| Input ID capture: SBID-Y | 0,1 | 0,1 | 0,1,4,5 | 0,2 | 0 | 0 |
| Input ID capture: BBID-Y | 0,1 | 1,2 | 0,2 | 0,1,5,4 | 4 | 4 |
| Timestamp-0 capture: X | 222333444-5 | 222333444-5 | 222333444 | 222333444 | 222333444 | 222333444 |
| Timestamp-1 capture: X | 555666777 | 555666777 | 555666777 | 555666777 | 555666777 | 555666777 |
| Timestamp-0 capture: Y | | | | | | |
| Timestamp-1 capture: Y | | | | | | |
| Recirc Block capture: X | 233 | 233 | 233 | 233 | 233 | 233 |
| Recirc Block capture: Y | 0, 120, 130 | 0, 120, 130 | 120, 130…190 | 120, 130…190 | 120 | 120 |
| Phase bin capture: X | 55666 | 55666 | 55666 | 55666 | 55666 | 55666 |
| Phase bin capture: Y | | | | | | |
| CMD capture: X | 010 | 010 | 010 | 010 | 010 | 010 |
| CMD capture: Y | | | | | | |
| On-line X/Y input error detect | DESSR: (1010.. | DESSR: (0101… | ata/phase (1010.. | ata/phase (0101… | Dvalids (101.. | Dvalids (0101.. |
| No errors--test vector mode | | | | | | |
| Input sync error detect—test vectors | | | | | | |
| Input stuck error detect—test vectors | | | y | y | | |
| Schid_frame + offset sync | | | 6 | 6 | | |
| Schid_frame – offset sync | | | | | | |
| Schid_frame + offset over detect | | | | | +70 | |
| Schid_frame – offset over detect | | | | | | -70 |
| Toggle DESSR_0/7, DESSR_8/15 bits | y | y | | | all 1's | all 1's |
| DUMP_SYNC offset detect | | | | | y | y |
| 16 baselines/chip (X 6,7 into CCQ-0) | | | | | | |
| Autocorr mode | | | | | | |
| Abort frames on LCI | | | | | | |
| MCB register configuration (write) | y | y | y | y | y | y |
| MCB register reads | y | y | y | y | y | y |
| Output lag data frames | y | y | y | y | y | y |
| DATA_BIAS/64 (approx # samples correlated) | 224, 724 | 224, 724 | 724 | 724 | 224 | 224 |
| Nominal data valid count | 200, 670 | 200, 670 | 670 | 670 | 200 | 200 |

| Description TEST CASE | 54 | 55 | 56 | 62 | 67 | 68 |
|---|---|---|---|---|---|---|
| Toggle all accum bits via split accum test mode | | | | | | |
| Toggle all accum OV bits via split accum test mode | | | | | | |
| Toggle all accum Bit 16, normal mode (accum ≥10000h); list CCC#s tested | 0,2,4-15 | | 0-15 | 0-15 | | |
| Correlate all CCCs at 256 Ms/s | | | | | | |
| All inputs, all pol'n prods, at different sample rates. | y | | | | y | |
| Chain all CCCs together | | | y | y | | |
| All CCCs corr inputs 0,1 | | | y | | | |
| Chain adjacent CCCs in pairs | | | | | | |
| Active inputs -- X | 0-7 | 0-7 | 0 | 6 | 0-7 | 6,7 |
| Active inputs -- Y | 0-7 | 0-7 | | | 0-7 | 0,1 |
| CCC#s: at least 1 wrap of diff phase | all except 3 | | | | | 0,1,2,3 |
| Single dump | | | y | y | | 1 |
| Multiple rapid dumps, with overrun detection, dump/clear | y | y | | | | |
| Different dump rates, different streams | all inputs | all inputs | | | | |
| Correlation enable/disable (CEN bit of MCSR) | 1 | 1 | 1 | 1 | 1 | 1 |
| Correlation HOLDOFF via DUMP_EN lines | all inputs | all inputs | | | | |
| Phase rotation enable/disable (PhEN bit of MCSR) | 1 | 1 | 1 (off internally) | 1 (off internally) | 1 | 1 |
| Test vector enable/disable (TVEN bit of MCSR) | 1 | 1 | 0 | 0 | 0 | 1 |
| OVR status bit of MCSR | y | y | | | | |
| AOV status bit of MCSR | | | | | | |
| SyncER status bit of MCSR | | y | y | y | | |
| X/Y_DS bit of MCSR: X dumping | | y | y | y | y | y |
| X/Y_DS bit of MCSR: Y dumping | y | | | | | |
| CCQ#s: XP0 -- XSw0 primary input | 0-3 | 0-3 | | | 0-3 | 1,3 |
| CCQ#s: XS0 -- XSw0 secondary input | | | | | | 0,2 |
| CCQ#s: XP1 -- XSw1 primary input | 0-3 | 0-3 | | | 0-3 | 1,3 |
| CCQ#s: XS1 -- XSw1 secondary input | | | | | | 0,2 |
| CCQ#s: YP0 -- YSw0 primary input | 0-3 | 0-3 | | | 0-3 | 0,1 |
| CCQ#s: YS0 -- YSw0 secondary input | | | | | | 2,3 |
| CCQ#s: YP1 -- YSw1 primary input | 0-3 | 0-3 | | | 0-3 | 0,1 |
| CCQ#s: YS1 -- YSw1 secondary input | | | | | | 2,3 |
| CCC#s: X_A -- adjacent input | | | 0-15 | 0-15 | | |
| CCC#s: X_P -- primary input | ,3,4,5,10,11,12,1 | ,3,4,5,10,11,12,14 | | | 2,3,6,7,10,11,14,1 | 0,1,8,9,12,13 |
| CCC#s: X_S -- secondary input | 0,2,6,7,8,9,13,15 | 0,2,6,7,8,9,13,15 | | | 0,1,4,5,8,9,12,13 | 2,3,10,11,14,15 |
| CCC#s: X_undef -- no connect (CCC not active) | | | | | | 4,5,6,7 |
| CCC#s: Y_A -- adjacent input | | | 0-15 | 0-15 | | |
| CCC#s: Y_P -- primary input | 0,2,6,7,8,9,13,15 | 0,2,6,7,8,9,13,15 | | | 0,2,4,6,8,10,12 | 1,3,9,11,13,15 |
| CCC#s: Y_S --secondary input | ,3,4,5,10,11,12,1 | ,3,4,5,10,11,12,14 | | | 1,3,5,7,9,11,13,1 | 0,2,8,10,13,15 |
| CCC#s: Y_undef -- no connect | | | | | | 4,5,6,7 |
| Input ID capture: SID-X | 85 | 70,171,172,175,17 | 85 | 85 | 200 | 200 |
| Input ID capture: SBID-X | 21-28 | 10,11,12,15,17 | 0 | 6 | 0-7 | 0,1,6,7 |
| Input ID capture: BBID-X | 0-7 | 0,1,4,6,7 | 0 | 4 | 0-7 | 2-5 |
| Input ID capture: SID-Y | 170 | 101 | 85 | 85 | 100 | 100 |
| Input ID capture: SBID-Y | 10-17 | 21,22,23,26,28. | 0 | 6 | 0-7 | 0,1 |
| Input ID capture: BBID-Y | 0-7 | 0,1,4,6,7 | 0 | 4 | 0-7 | 4,5 |
| Timestamp-0 capture: X | | …aaa - …abd | 222333444 | 222333444 | 222333444 | 222333444 |
| Timestamp-1 capture: X | | 55555555h | 555666777 | 555666777 | 555666777 | 555666777 |
| Timestamp-0 capture: Y | …aaa - …af9 | | | | | |
| Timestamp-1 capture: Y | 55555555h | | | | | |
| Recirc Block capture: X | 120, 130…190 | 70,171,172,175,17 | 233 | 233 | 233 | 233 |
| Recirc Block capture: Y | 80, 81…87 | 120,130,140,170 | 233 | 233 | 120-170 | 120,130 |
| Phase bin capture: X | | 21841-21847 | 55666 | 55666 | 55666 | 55666 |
| Phase bin capture: Y | 43690 - 43697 | | | | | |
| CMD capture: X | | | 010 | 010 | 010 | 010 |
| CMD capture: Y | all patterns | all patterns | | | | |
| On-line X/Y input error detect | dump_en 1,3,5,7; X | dump_en 0,2,4,6; Y | even X/Y inputs, | odd X/Y inputs, inactive | | inactive se_clk |
| No errors--test vector mode | | | | | | |
| Input sync error detect—test vectors | | | | | | |
| Input stuck error detect—test vectors | | | | | | |
| Schid_frame + offset sync | | +10 | +6 | +6 | +6 | |
| Schid_frame – offset sync | -30 | | | | | |
| Schid_frame + offset over detect | | | | | | |
| Schid_frame – offset over detect | | | | | | |
| Toggle DESSR_0/7, DESSR_8/15 bits | all 1's | | | | | |
| DUMP_SYNC offset detect | y (dump off=+3) | | | | | |
| 16 baselines/chip (X 6,7 into CCQ-0) | | | | | | y |
| Autocorr mode | | | input 0 | input 6 | | |
| Abort frames on LCI | | | | | | |
| MCB register configuration (write) | y | y | y | y | y | y |
| MCB register reads | y | y | y | y | y | y |
| Output lag data frames | y | y (CCC 0,2 no) | y | y | y | y |
| DATA_BIAS/64 (approx # samples correlated) | 420…7406 | 92…3678 | 2724 | 2724 | 724 | 724 |
| Nominal data valid count | 380…6900 | 80…3470 | 2430 | 2430 | 650 | 650 |

| Description | Coverage Comments |
|---|---|
| Toggle all accum bits via split accum test mode | 100% coverage: tc0, tc1 |
| Toggle all accum OV bits via split accum test mode | 100% coverage: tc1 |
| Toggle all accum Bit 16, normal mode (accum ≥10000h); list CCC#s tested | 100% coverage for lags, data_bias; dvcounts NOT COVERED (requires 260 usec integration) |
| Correlate all CCCs at 256 Ms/s | 100% coverage |
| All inputs, all pol'n prods, at different sample rates. | 100% coverage |
| Chain all CCCs together | 100% coverage |
| All CCCs corr inputs 0,1 | 100% coverage |
| Chain adjacent CCCs in pairs | 100% coverage |
| Active inputs -- X | 100% coverage, all X inputs used |
| Active inputs -- Y | 100% coverage, all Y inputs used |
| CCC#s: at least 1 wrap of diff phase | 100% coverage on all CCCs |
| Single dump | 100% coverage |
| Multiple rapid dumps, with overrun detection, dump/clear | 100% coverage |
| Different dump rates, different streams | 100% coverage X and Y inputs |
| Correlation enable/disable (CEN bit of MCSR) | 100% coverage, however, no verif that if 0 it completely blocks frame generation |
| Correlation HOLDOFF via DUMP_EN lines | 100% coverage |
| Phase rotation enable/disable (PhEN bit of MCSR) | 100% coverage 0/1 |
| Test vector enable/disable (TVEN bit of MCSR) | 100% coverage |
| OVR status bit of MCSR | 100% coverage |
| AOV status bit of MCSR | 100% coverage |
| SyncER status bit of MCSR | 100% coverage |
| X/Y_DS bit of MCSR: X dumping | 100% coverage |
| X/Y_DS bit of MCSR: Y dumping | 100% coverage |
| CCQ#s: XP0 -- XSw0 primary input | 100% coverage; all CCQs |
| CCQ#s: XS0 -- XSw0 secondary input | 100% coverage; all CCQs |
| CCQ#s: XP1 -- XSw1 primary input | 100% coverage; all CCQs |
| CCQ#s: XS1 -- XSw1 secondary input | 100% coverage; all CCQs |
| CCQ#s: YP0 -- YSw0 primary input | 100% coverage; all CCQs |
| CCQ#s: YS0 -- YSw0 secondary input | 100% coverage; CCQs 1-3, CCQ-0 is no connect |
| CCQ#s: YP1 -- YSw1 primary input | 100% coverage; all CCQs |
| CCQ#s: YS1 -- YSw1 secondary input | 100% coverage; CCQs 1-3, CCQ-0 is no connect |
| CCC#s: X_A -- adjacent input | 100% coverage; all CCCs |
| CCC#s: X_P -- primary input | 100% coverage; all CCCs |
| CCC#s: X_S -- secondary input | 100% coverage; all CCCs |
| CCC#s: X_undef -- no connect (CCC not active) | 25% coverage; CCCs 4-7; no timing impact, not |
| CCC#s: Y_A -- adjacent input | 100% coverage; all CCCs |
| CCC#s: Y_P -- primary input | 100% coverage; all CCCs |
| CCC#s: Y_S --secondary input | 100% coverage; all CCCs |
| CCC#s: Y_undef -- no connect | 25% coverage; CCCs 4-7; no timing impact, not critical |
| Input ID capture: SID-X | 100% coverage, all bits, aggregate across inputs |
| Input ID capture: SBID-X | 100% coverage, all bits, aggregate across inputs |
| Input ID capture: BBID-X | 100% coverage, all bits, aggregate across inputs |
| Input ID capture: SID-Y | 100% coverage, all bits, aggregate across inputs |
| Input ID capture: SBID-Y | 100% coverage, all bits, aggregate across inputs |
| Input ID capture: BBID-Y | 100% coverage, all bits, aggregate across inputs |
| Timestamp-0 capture: X | Full bit range covered, all bits toggled. |
| Timestamp-1 capture: X | Full bit range covered, all bits toggled. |
| Timestamp-0 capture: Y | Full bit range covered, all bits toggled. |
| Timestamp-1 capture: Y | Full bit range covered, all bits toggled. |
| Recirc Block capture: X | All bits toggled, 100% aggregate coverage. |
| Recirc Block capture: Y | All bits toggled, 100% aggregate coverage. |
| Phase bin capture: X | All bits toggled, 100% aggregate coverage. |
| Phase bin capture: Y | All bits toggled, 100% aggregate coverage. |
| CMD capture: X | All bits toggled, 100% aggregate coverage. |
| CMD capture: Y | All bits toggled, 100% aggregate coverage. |
| On-line X/Y input error detect | 100% aggregate coverage. |
| No errors--test vector mode | 100% coverage |
| Input sync error detect—test vectors | 100% coverage |
| Input stuck error detect—test vectors | 100% coverage |
| Schid_frame + offset sync | 100% coverage |
| Schid_frame – offset sync | 100% coverage |
| Schid_frame + offset over detect | 100% coverage |
| Schid_frame – offset over detect | 100% coverage |
| Toggle DESSR_0/7, DESSR_8/15 bits | 100% coverage |
| DUMP_SYNC offset detect | 100% coverage |
| 16 baselines/chip (X 6,7 into CCQ-0) | 100% coverage |
| Autocorr mode | Inputs 0 and 6 only |
| Abort frames on LCI | 100% coverage |
| MCB register configuration (write) | 100% coverage |
| MCB register reads | 100% coverage |
| Output lag data frames | 100% coverage: lag frames from all CCCs |
| DATA_BIAS/64 (approx # samples correlated) | Information only |
| Nominal data valid count | Information only |

# 7 Test Results

The test results reported here are for the following gate-level model, delivered by ISine on December 11, 2005 with simulation execution starting on December 12, 2005:

- **`corr_chip_1211.tar`** – This 150272000 byte tar file was unpacked and unzipped to produce the following files:
  - **`corr_chip_full_opt7_rename_flat_fill_1211.vg`** – 21805249 bytes.
  - **`corr_chip_full_opt7_rename_flat_fill_1211_slow.sdf`** – 1179711436 bytes.
  - **`corr_chip_full_opt7_rename_flat_fill_1211_fast.sdf`** – 1179711434 bytes.

The RTL simulation was run on December 9, 2005 with the July 28, 2005 release of RTL code (contained in the file **`EVLA_Corr_Chip_25082_Release_HDLFiles_Jul28-05.zip`**). Both the RTL and the gate-level simulations were run with 10 ps timing resolution.

The slow timing model gate-level sim was run on "evladual" starting the morning of December 12, 2005. It finished on the morning of December 19, 2005. The slow timing model was run with the wave window open using Modelsim SE-64 6.1a. The fast timing model gate-level sim was run on the Jodrell Bank computer starting the morning of December 12, 2005 using Modelsim SE-64 Plus 6.0a. It finished on the afternoon of December 19, 2005. The wave window was not opened as it was found that some sort of bug caused Modelsim to crash on the Jodrell Bank computer when the wave window was opened (remotely, or locally by Bryan Anderson).

According to the contractor, the slow and fast models are for the following conditions:

- **slow** model – 1.08 V, slow process, high temperature.
- **fast** model – 1.32 V, fast process, low temperature.

Verification results, for each test case, including notes indicating any anomalies are shown in Table 7-1. For these tests, the correlator chip was run at 312.5 MHz to ensure adequate timing margin (22%) over 256 MHz.

| Test Case | Notes/Anomalies | Pass/ Fail |
|---|---|---|
| **00** | Exact match. | P |
| **01** | Exact match. | P |
| **36** | Exact match. | P |
| **38** | Exact match. | P |
| **39** | Exact match. | P |
| **40** | Exact match. | P |
| **41** | Exact match. | P |
| **42** | Exact match. | P |
| **43** | Exact match (fast model). In the gate-level slow timing model, one more word is produced on the LCI when aborting the frame. This effect is due to XCLOCK input to LCI_DATA_CLKOUT delay and is an acceptable result. It does not show up in the fast timing model simulation | P |
| **48** | Exact match. | P |
| **49** | Exact match. | P |
| **50** | Exact match. | P |
| **51** | Exact match. | P |
| **52** | Exact match. | P |
| **53** | Exact match. | P |
| **54** | Exact match (fast model). In the gate-level slow timing model for this test case with fast dump and overruns, in some cases frames are produced in a slightly different order and different sequence than in the RTL model. This effect is due to XCLOCK input to LCI_DATA_CLKOUT delay of 1.9 nsec, coupled with fast dumping and overruns and is an acceptable and expected result. It does not show up in the fast timing model simulation. | P |
| **55** | Exact match. | P |
| **56** | Exact match. | P |
| **62** | Exact match. | P |
| **67** | Exact match. | P |
| **68** | Exact match. | P |

**Table 7-1 Verification results table.**

# 8 I/O Timing

## 8.1 Output Timing Measurements

The Table 8-1 contains measurements made of the chip's gate-level model output timing using the Modelsim wave window. Slow timing is for 1.08 V core, fast is for 1.32 V core, and both are for 2.5 V I/O voltages.

| **Measurement Description** | **Timing parameter** | **Slow model (nsec)** | **Fast model (nsec)** | **Load cap (pF)** |
|---|---|---|---|---|
| Daisy-chain setup time: X*_O (X outputs) stable relative to ↑ XCLOCK_O | $\mathbf{t_{su\_XO}}$ | 2.19 | 2.05 | 20 |
| Daisy-chain hold time: X*_O (X outputs) stable relative to ↑ XCLOCK_O | $\mathbf{t_{h\_XO}}$ | 0.69 | 1.34 | 20 |
| Daisy-chain setup time: Y*_O (Y outputs) stable relative to ↑ YCLOCK_O | $\mathbf{t_{su\_YO}}$ | 1.77 | 1.88 | 20 |
| Daisy-chain hold time: Y*_O (Y outputs) stable relative to ↑ YCLOCK_O | $\mathbf{t_{h\_YO}}$ | 0.81 | 1.39 | 20 |
| Daisy-chain pin-to-pin input-to-output delay X (measured from X_SCHID_FRAME input mid-cell to X_SCHID_FRAME_O output at XCLOCK_O edge) | $\mathbf{t_{p2p\_X}}$ | 12.2 | 8.55 | 20 |
| Daisy-chain pin-to-pin input-to-output delay Y (measured from Y_SCHID_FRAME input mid-cell to Y_SCHID_FRAME_O output at YCLOCK_O edge) | $\mathbf{t_{p2p\_Y}}$ | 11.4 | 7.86 | 20 |
| LCI data and control signals **max** clock-to-output delay relative to ↑ LCI_DATA_CLKOUT | $\mathbf{t_{co\_max\_LCI}}$ | 1.5 | 0.6 | 15.6 |
| LCI data and control signals **min** clock-to-output delay relative to ↑ LCI_DATA_CLKOUT | $\mathbf{t_{co\_min\_LCI}}$ | 0.31 | -0.17 | 15.6 |
| MCB DATA read **max** clock-to-output delay relative to ↑ of MCB_CLK | $\mathbf{t_{co\_max\_MCB}}$ | 13.5 | 5.8 | 143 |
| MCB DATA read **min** clock-to-output delay relative to ↑ of MCB_CLK | $\mathbf{t_{co\_min\_MCB}}$ | 12.2 | 5.17 | 143 |

| **Duty Cycle** | **Slow model (%)** | **Fast model (%)** |
|---|---|---|
| XCLOCK_O duty cycle | 47.3 | 48.7 |
| YCLOCK_O duty cycle | 47.6 | 48.8 |
| LCI_DATA_CLKOUT duty cycle | 40.1 | 44.7 |

**Table 8-1 Critical timing measurements from the gate-level simulation.**

The chip was run at 256 MHz in the simulation to produce the output timing numbers, and results from both the slow and fast timing models are included. 256 MHz was used instead of 312.5 MHz, because $t_{h\_XO}$ and $t_{h\_YO}$ are lower in the slow timing model at the higher frequency due to the way the outputs are clocked and the output clock is generated. The chip could be run at the higher frequency, but this would require a higher voltage and thus the slow timing model is not applicable.

The load capacitance for which the timing model is defined is as shown, and taken from the .sdc file provided to ISine to define timing constraints. Note that the testbench does not probe the input timing tolerance of the gate-level model. For X and Y inputs, the ideal mid-cell setup and hold times are provided. Similarly for the LCI and MCB interfaces.

***Note:***

> ***Since the clock-to-output delay for the fast timing model on the LCI interface is negative, and provide there is not much spread between $t_{co}$ min and max for both models, it is likely acceptable to set the LTA FPGA so that it clocks data on the falling, rather than rising edge of LCI_DATA_CLKOUT.***

Timing diagrams for the timing parameters of Table 8-1 are shown in Figure 8-1 to Figure 8-4.

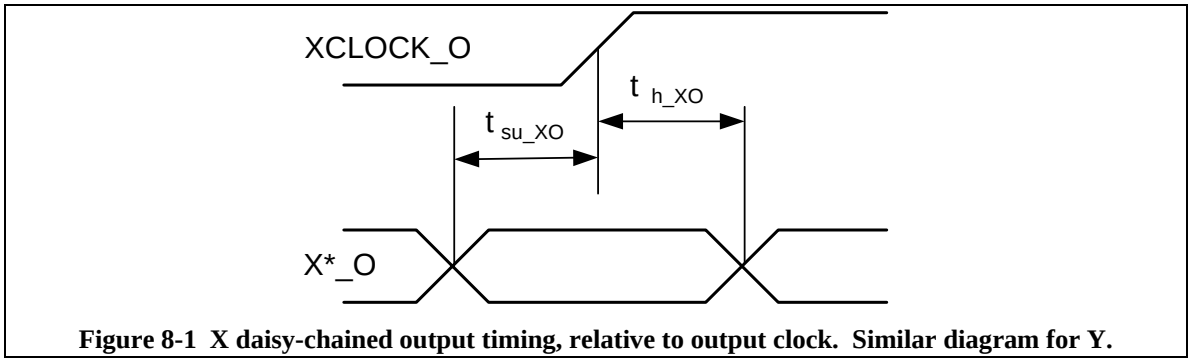


**Figure 8-1 X daisy-chained output timing, relative to output clock. Similar diagram for Y.**

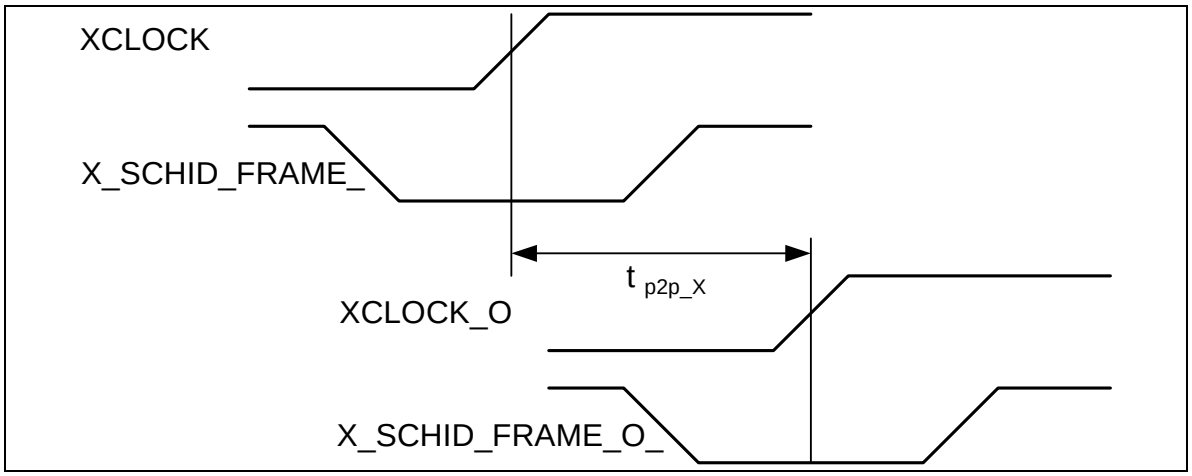

**Figure 8-2 X daisy chained input-to-output pin-to-pin timing. Similar diagram for Y.**

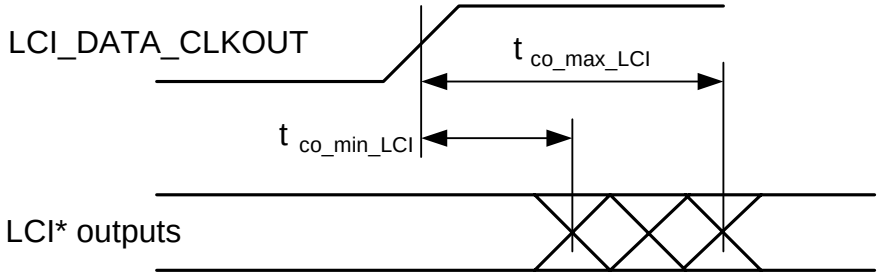


**Figure 8-3 LCI output timing, relative to LCI_DATA_CLKOUT.**

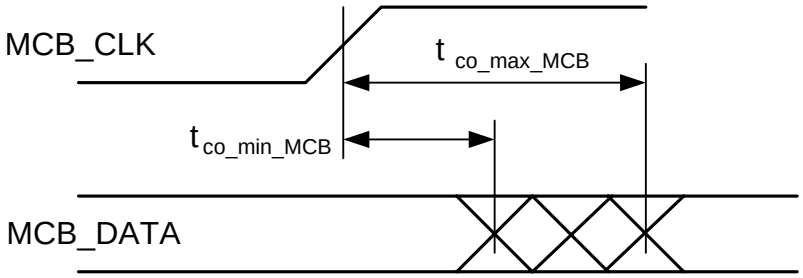


**Figure 8-4 MCB_DATA output timing, relative to MCB_CLK.**

## 8.2 Input Timing Requirements

The input timing requirements of the chip are provided by the contractor ISine and contained in Table 8-2. Slow timing is for 1.08 V core, fast is for 1.32 V core, and both use 2.5 V I/O voltages.

| Description | Timing parameter | Slow model (nsec) | Fast model (nsec) | Input cap (pF) |
|---|---|---|---|---|
| Input setup time, X inputs relative to ↑ X_CLOCK_I | $\mathbf{t_{su\_XI}}$ | 0.66 | 0.15 | 4.7 |
| Input hold time, X inputs relative to ↑ X_CLOCK_I | $\mathbf{t_{h\_XI}}$ | 0.30 | 0.25 | 4.7 |
| Input setup time, Y inputs relative to ↑ Y_CLOCK_I | $\mathbf{t_{su\_YI}}$ | 0.48 | 0.26 | 4.7 |
| Input hold time, Y inputs relative to ↑ Y_CLOCK_I | $\mathbf{t_{h\_YI}}$ | 0.49 | 0.28 | 4.7 |
| Input setup time, LCI inputs (CS, OE, FRAME_ABORT) relative to ↑ LCI_DATA_CLKOUT | $\mathbf{t_{su\_LCI}}$ | 5.1 | 2.48 | 4.7 |
| Input hold time, LCI inputs (CS, OE, FRAME_ABORT) relative to ↑ LCI_DATA_CLKOUT | $\mathbf{t_{h\_LCI}}$ | -4.5 | -2.23 | 4.7 |
| Input setup time, MCB inputs (CS, RD_WR, DATA, ADDR) relative to ↑ MCB_CLK | $\mathbf{t_{su\_MCB}}$ | 3.15 | 1.27 | 4.7 |
| Input hold time, MCB inputs (CS, RD_WR, DATA, ADDR) relative to ↑ MCB_CLK | $\mathbf{t_{h\_MCB}}$ | -0.49 | -0.02 | 4.7 |
| Minimum RESET_ time | $\mathbf{t_{RESET}}$ | 12 | 12 | 4.7 |
| Minimum PLL_RESET_ time | $\mathbf{t_{PLL\_RESET}}$ | 200 | 200 | 4.7 |
| Minimum time from PLL_RESET_ release to RESET_ release | $\mathbf{t_{P\_R}}$ | 500 us | 500 us | |
| Input setup time, JTAG inputs (TDI, TMS) relative to TCK | $\mathbf{t_{su\_JTAG}}$ | 2.56 | 0.96 | 4.7 |
| Input hold time, JTAG inputs (TDI, TMS) relative to TCK | $\mathbf{t_{h\_JTAG}}$ | -0.14 | 0.13 | 4.7 |
| | | | | |
| Minimum frequency, X_CLOCK_I, Y_CLOCK_I | $\mathbf{freq_{min}}$ | 25 (MHz) | 25 (MHz) | |
| Maximum frequency, X_CLOCK_I, Y_CLOCK_I | $\mathbf{freq_{max}}$ | 32 (MHz) | 32 (MHz) | |
| Maximum frequency, MCB_CLK | $\mathbf{MCB_{max}}$ | 33 (MHz) | 33 (MHz) | |

**Table 8-2 Input timing requirements, as provided by ISine**

Input timing tolerance of the chip relative to the output of the previous instance of the gate-level model was checked using the "corr_chip_daisy_tb" test bench, at 256 MHz. This test bench contains two instances of the correlator chip—one chip feeding the next chip. The slow timing model was used since this is the model with the worst-case X/Y output skew relative to the daisy-chained X/Y output clocks.

This test indicated that adequate $t_{su}$ and $t_{hold}$ times are provided for the second chip in the daisy chain, although examination of Table 8-1 and Table 8-2 confirms that all timing requirements are met.

The daisy-chain test bench was run with test case 36 and the results exactly match the RTL simulation for the normal, single-chip, test bench. This comparison, and code built into the test bench tester module that compares the inputs with the outputs indicates that data is properly daisy-chained through the chip.

# 9 Power Dissipation Estimate

Test case 36 was run with the gate-level netlist file provided by ISine named: "`corr_chip_full_opt7_rename_flat_fill_1121_unq.vg`". A .do file was setup to generate a Modelsim .vcd (Value Change Dump) file and it contains a list of every node in the chip that changes for as long as Modelsim is writing to the file. The simulation was run with no timing file, since no timing file was provided or is necessary for vcd file generation. The clock rate for the simulation was 312.5 MHz, as for all other simulation results reported in this document.

Initially, the vcd file was generated for the full duration of the simulation, but this resulted in an unwieldy file size of almost 50 Gbytes. The .do file was changed so that a vcd file was written for the total duration of 1 usec from 20 to 21 usec of the simulation time. During this time, the chip is processing inputs and correlating all CCCs at full speed, all X/Y inputs and outputs active, and generating output data frames on the LCI. Thus, the chip is maximally active[2] for this test.

The vcd file was sent to ISine for power dissipation analysis during the week of December 5th, 2005. The name of the file that was sent is "**`EVLA_power_312dot5MHz_tc36_1usec_20-21_Dec7-05.vcd.gz`**".

ISine reported that due to the way split power planes are setup and named in the chip design, the tools would not allow a top-level power dissipation estimate based on this vcd file.

[2] It is asserted that the chip dissipates less power when operating at ½ sample rate with all SE_CLKs toggling, since all multipliers and phase rotators in the chip, as well as data shifting down the lag shift registers, are operating at ½ speed. Nevertheless, if there is any doubt, a test and vcd file generation could be done at ½ rate if desired.

Thus, the power dissipation estimate of 3.35 W core power, at 1.02 V, and the I/O power of 0.69 W at 2.5 V is based on the following:

- Test bench ccc_tb.v provided by NRC on April 12, 2005. The tester stimulates the CCC with Gaussian data, and nominal phase rates.
- Simulation by ISine of the ccc_rc_0809.vg file for 42 usec, to generate the ccc_rc_0810.vcd file. The .vg file is a gate-level netlist for one CCC in the chip.
- The power estimate of the CCC, at 1.32 V supply and 256 MHz is 331.9 mW.
- The full array of 16 CCCs thus consume 5.31 W. The remainder of non-CCC logic is estimated at 300 mW, for a total of 5.6 W at 1.32 V for the most pessimistic process and temperature.
- Derating power to 1.02 V, yields 3.35 W core power dissipation. ISine indicates that the chip can run at 1.02 V and still meet timing, due to margin in static timing analysis and simulation. Design margin also indicates that the chip could run as low as 0.9 V and still meet timing at 256 MHz. If 0.9 V operation is possible, then power dissipation could be as low as 2.6 W core.
- I/O power estimation is based on a Spice simulation of the output buffers with estimates of capacitive loading and activity levels as provided by NRC. I/O power is estimated at 0.69 W.

The total chip power dissipation is estimated at 3.35 W + 0.69 W = 4.04 W, and may be as low as 3.3 W at 0.9 V core.

## 10 References

[1] Carlson, B., REQUIREMENTS AND FUNCTIONAL SPECIFICATION: EVLA Correlator Chip, RFS Document: A25082N0000, Revision 2.4, February 15, 2005.

[2] Smegal, R. TEST AND VERIFICATION PLAN: EVLA Correlator Chip Functional Test Cases, TVP Document: A25082N0002, Revision DRAFT 4.0, November 30, 2005.

[3] Smegal, R., TEST AND VERIFICATION REPORT: EVLA Correlator Chip Functional Test Results, TVR Document: A25082N0003, Revision DRAFT, November 22, 2005.

[4] Lau, F., REQUIREMENTS AND FUNCTIONAL SPECIFICATION: EVLA Correlator Chip Testing Software, Comparison of Functional Simulation Output with Post-Place-and-Route Simulation Output, RFS Document: A25082N0010, Revision 1.2, May 27, 2003.