LASER INTERFEROMETER GRAVITATIONAL WAVE OBSERVATORY
-LIGO-

LIGO Laboratory /LIGO Scientific Collaboration

Technical Document LIGO-T1100515 3/13/2011

Test Procedure for the

Timing Comparator/Frequency Counter Chassis of

Advanced LIGO Timing System

Ilya Belopolski, Stefan Countryman, Ilya Belopolski, Peter Raffai, Zsuzsa Marka
Columbia University

LIGO Hanford Observatory
P.O.Box. 1970
Mail Stop S9-02
Richland,WA 99352
Phone (509) 372-8106
Fax (509) 372-8137

LIGO Livingston Observatory
P.O.Box. 940
Livingston, LA 70754
Phone (225) 686-3100
Fax (225) 686-7189

Columbia University
Columbia Astrophysics Laboratory
Pupin Hall - MS 5247
New York NY 10027
Phone (212) 854-8209
Fax (212) 854-8121
E-mail: geco.cu@gmail.com

WWW: http://www.ligo.caltech.edu

Comparator daughterboard board document LIGO DCC # D070568 – revision E

Slave board document LIGO DCC# D070071 - revision C

Board Serial #s (Comparator/Slave):

Test Engineer:

Test Date:

Overall Comparator/Frequency Counter chassis testing: ☐ PASSED ☐ FAILED

Testing schedule:

1. Hardware Tests:
    a. Power up
    b. Test point voltages
    c. FPGA programming
    d. Synchronization
2. Comparator testing
    1PPS Input ports (1-7)
3. Testing 1PPS outputs (8-11)
4. Frequency Counter testing
    Ports 12-17

Hardware:

1 Altium USB JTAG Adapter (refer to figure 2 in Appendix)
2 12V power supply, cord with 3-pin D-sub connectors
3 Voltmeter
4 2 fiber-optic multi-mode transceivers and LC-LC fiber pair (transceiver e.g. HFBR 57E0PZ)
5 Windows®-operated PC with Altium Summer09 edition and Xilinx ISE WebPack (12.1 or lower edition) installed
6 Reference clocking (i.e. 1PPS) source w. optical output. e.g. a working aLIGO MFO chassis LIGO D070011/D08094

7 BNC cable

Software:

1 Altium® Designer v Summer 09

2 This test procedure was written using RFCounterComparator FPGA code (FPGARFCounterComparator - Altium FPGA project) available from https://svn.ligo.caltech.edu/svn/Altium-D2D at revision 84.

A brief introduction to the Master/FanOut board functionality and the advanced LIGO timing system can be found online at Quick Start Guide (LIGO DCC # E080541)

---

1. Hardware Tests: Power Up & Programming

a. **Power-up Lights**. Upon power-up, do the unmarked circular LED and the timing interface "On" LED (both on the front panel) light up?

Circular LED: ☐ "On" LED: ☐

b. **Test Points**. What voltages are observed at each test point on the Comparator Daughterboard?

TP1. Nominal: 12 ±0.5V Measured:

TP2. Nominal: 5.0 ±0.5V Measured:

TP4. Nominal: 3.3 ±0.5V Measured:

c. **FPGA Programming**. Connect the Timing Comparator through the JTAG interface using a JTAG-USB adapter. Using Altium, load the FPGA program (see appendix).

Was the FPGA chip programmed successfully? ☐

d. **Synchronization**. Connect the Timing Comparator to one of the 1-16 outputs of an advanced LIGO Master/Fanout chassis via fiber optic cable.

Does the Timing Slave "Up-link" LED exhibit a 1/2 Hz blink (indicating synchronization with the MFO) within 3 minutes? ☐

On the board itself, do DS14, DS3, DS7 and DS11 (located next to one another adjacent to the front panel) start blinking with a frequency of 1/2 Hz (1 second on, 1 second off)?

DS14: ☐ DS3: ☐ DS7: ☐ DS11: ☐

2. 1 Pulse per Second Comparator

The main purpose of the Timing Comparator is to measure the time difference between 1 PPS input signals (routed through BNC inputs on the front panel) and the 1PPS signal sent from a Timing MFO through the fiber optic cable. During normal operation, this data is sent back to the MFO; for the purpose of this test, Altium virtual instruments are used.

The time difference is reported in *clock cycles (clock frequency: 2^{26} Hz)*. The sign of the time difference is negative if the PPS signal associated with the BNC input arrives after the reference pulse (from MFO via fiber optic); it is positive if the PPS signal associated with the BNC input arrives before the reference pulse. (Please note, while checking the same BNC input PPS pulse using an oscilloscope and triggering on the reference (e.g. on 1 PPS output from back panel of the MFO) we measure time *delay* – exactly the opposite definition of signs.)

Digital I/O Reference: U_Comparator

Inputs

COMP1_DIFF[31..0]: Measures the delay in clock cycles for 1st 1 PPS Input
COMP2_DIFF[31..0]: Measures the delay in clock cycles for 2nd 1 PPS Input
COMP3_DIFF[31..0]: Measures the delay in clock cycles for 3rd 1 PPS Input
COMP4_DIFF[31..0]: Measures the delay in clock cycles for 4th 1 PPS Input
COMP5_DIFF[31..0]: Measures the delay in clock cycles for 5th 1 PPS Input
COMP6_DIFF[31..0]: Measures the delay in clock cycles for 6th 1 PPS Input
COMP7_DIFF[31..0]: Measures the delay in clock cycles for 7th 1 PPS Input

COMP_SIGN is defined as *0 - positive, 1- negative* for all (1st to 7th) 1 PPS Inputs

Test Procedure

1. In Altium, open the Digital I/O called U_COMPARATOR. You should see the above fields under "INPUTS", while "OUTPUTS" should contain no fields. Is each of the above values nonzero?

COMP1	COMP2	COMP3	COMP4	COMP5	COMP6	COMP7
☐	☐	☐	☐	☐	☐	☐

2. Take a BNC Cable and attach one end to a 1 PPS signal source, such as the one found on the back panel of any Timing Master/Fanout.

3. Connect the other end of the BNC cable to each of the 1 PPS Inputs on the front board, starting with No. 1 and ending with No. 7; make sure to leave the cable connected to each for several seconds. Does the value of each of the Digital I/O field change? What is the new value?

1 PPS Input No:	Digital I/O Value Changes?	New Digital I/O Value:
1	☐	
2	☐	
3	☐	
4	☐	
5	☐	
6	☐	
7	☐	

Are all new Digital I/O Values the same (within a few clock cycles)? ☐

Comments:

3. Testing 1 PPS outputs

Outputs 8-11 are 1PPS ticks synchronized to the 1PPS received from the Master/Fanout via the optical fiber. Their delays compared to the Master/Fanout 1PPS taken from the 1PPS output on the Master/Fanout should be within 50 ns.

Test Procedure:

a.Using an oscilloscope *measure the time delay* between rising edge of the 1PPS output from the back panel of the Master/Fanout chassis (trigger) and the rising edge of outputs 8-11 on the Timing Comparator/RF Counter chassis. Indicate with a checkmark, in the table below, whether the 1PPS is present at each output and within 50 ns of the reference.

	OUTPUT8	OUTPUT9	OUTPUT10	OUTPUT11
1PPS present and within 50ns of reference	☐	☐	☐	☐

b. *Offset 1PPS output test*: Outputs 9-11are also capable of producing a 1PPS that is delayed or ahead of the reference tick. Offset for outputs 9-11 is set using U_GENERATE_OFFSET_1PPS digital I/O.

Digital I/O Reference: U_GENERATE_1PPS_OFFSET (only for outputs 9-11)

GENERATE_1PPS_OFFSET: Set 'high' (i.e. 1) to generate offset, otherwise set to 'low'.

SUBSECOND_DIFF: Output time offset, in clock cycles (1 clock cycle is 2^{-26}s, approx. 15 ns).

SUBSECOND_DIFF_SIGN*:* Output offset sign - *0 is positive, 1 is negative.* Just as before, the sign is negative if the PPS signal associated with the BNC output is after the reference pulse (from MFO via fiber optic); and it is positive if the PPS signal associated with BNC output is before the reference pulse. (Please note, while checking the same BNC output PPS pulse using an oscilloscope and triggering on the reference (e.g. on 1 PPS output from back panel of the MFO) we measure time *delay* – exactly the opposite definition of signs.)

BASELINE_ DELAY: There is a delay between the output PPS and reference PPS signals which is inherent in the hardware. This register allows the user to compensate for this inherent delay.

Procedure:

1. Using a BNC cable, connect the Output 9 (10,11) port to any of the input ports (1-7).

2. Set GENERATE_1PPS_OFFSET to '1'.

3. Set Baseline Delay: input the value of the delay at startup, as shown in the COMP register of the corresponding BNC input port in the U_COMPARATOR digital I/O (Nominal: a few clock cycles at startup), into the BASELINE_ DELAY_9(10,11) register. This value will be automatically subtracted from the SUBSECOND_DIFF output register.

4. Add an arbitrary time offset: set SUBSECOND_DIFF to an arbitrary large number of clock cycles (try on the order of 10^4). Record the entered values and their signs. (Note: the timing clock has frequency 2^{26}Hz.)

5. Offset measured: record the measured time offset from the COMP register of the corresponding BNC input port in the U_COMPARATOR digital I/O.

	OUTPUT9	OUTPUT10	OUTPUT11
Baseline Delay in clock cycles			
Time offset added in clock cycles			
Offset measured in clock cycles			
Do they agree (within a few cycles)?	☐	☐	☐

Comments:

4. Testing of the RF Counter

The Advanced LIGO Timing System is particularly well-adept for absolute frequency measurements because it produces a highly precise one pulse per second (1PPS) signal that can be used as a benchmark. The RF Counter input frequency range is indicated in LIGO Document T1200086.

Test Procedure:

1. Use a function generator (e.g.) to inject sinusoids of given frequencies (see table below) with 1V amplitudes into the RF Counter input ports 12-17 at the back panel of the Timing Comparator/ RF Counter chassis.

2. Read the frequency measurement of the RF counter through Digital I/O code programmed into the FPGA chip using Altium Designer (Digital I/O called U_FREQUENCY_COUNT).

Note: The RF Counter test is successful if the frequency as shown on the digital I/O port was stable within 10Hz of the frequency reported by the function generator.

RF Input No:	RF Input	Is observed RF within 10Hz?	RF Input	Is observed RF within 10Hz?
12	1MHz	☐	25MHz	☐
13	1MHz	☐	25MHz	☐
14	1MHz	☐	25MHz	☐
15	1MHz	☐	25MHz	☐
16	1MHz	☐	25MHz	☐
17	1MHz	☐	25MHz	☐

Are all frequency values in required range? ☐

Comments:

APPENDIX Programming FPGA Firmware onto Timing Module

The following step by step instruction describes the programming of the FPGA chip and Flash PROM on a Timing Module, such as the Master, FanOut or Slave modules. The guideline assumes that one has the following done/ready prior to going through the steps:
- Computer with Altium Designer is installed on it. (Summer09 version is preferred.)
- Xilinx ISE WebPACK installed on the same computer (version 11 or earlier)
- Altium USB-JTAG adapter.

1. Open Altium Designer.
2. Open 'My Account' from the 'DXP' menu bar (see Figure 1).
3. Press 'Sign In', provide your user name and password, then press 'Sign In' (see Figure 1).
4. Activate the product by clicking on the 'Activate' button (see Figure 1).

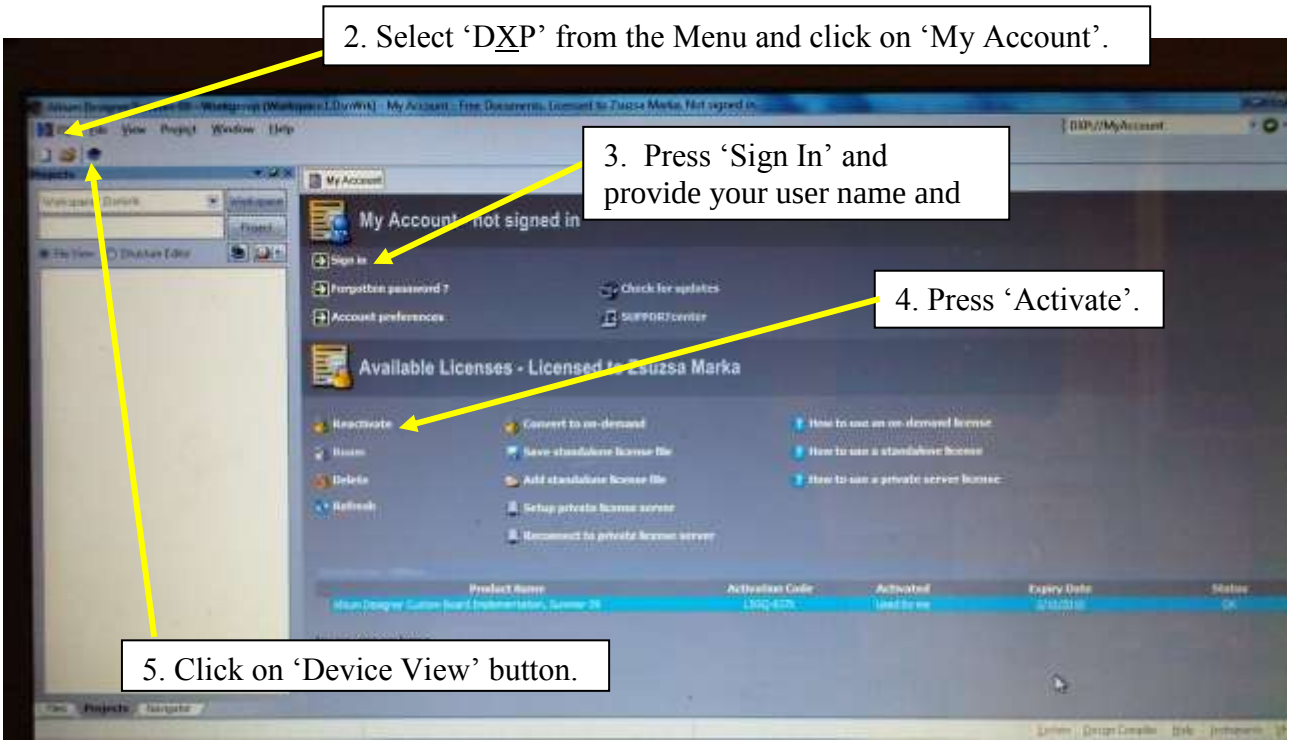

Figure 1. My Account window. The steps of signing in and activating an account that allows the use of the software are indicated. We also indicated the position of the Device View icon that opens the window from where FPGA programming will be done.

5. Click on 'Device View' button (see Figure 1).
6. Connect the computer to the desired board (Slave, Master or FanOut) using Altium USB-JTAG adapter.
7. Make sure that 'Live' is checked on the upper right corner of the 'Devices' screen in Altium Designer (See Figure 1).

8. Make sure that ‘Connected’ is shown on the upper left corner of the screen (see Figure 1).
9. Load Project. In the ‘File’ menu bar of Altium Designer, click on ‘Open Project’. Select the project you want to open and open it. For example, for slaves with comparator daughterboards:
   ….\ RFCounterComparator\FPGARFCounterComparator.PrjFpg

10. Make sure that the correct Project is selected for the FPGA:
    From the menu below the FPGA chip icon select the project you need:
    RFCounterComparator/ Spartan3

11. Reset Flash PROM and the FPGA. Right-click on Flash PROM (or the FPGA) icon and choose ‘Reset Hard Device’. Note that resetting can take up to a few minutes for the PROM. Progress is indicated on the lower left corner of the screen.
12. Upload FPGA program to Flash PROM. Right-click on Flash PROM icon as in the previous step, and click on ‘Choose File and Download…’
    The file you need to select and download for testing IRIG-B is given below:
    …\ RFCounterComparator\ProjectOutputs\Spartan3\fpgarfcountercomparator.mcs
    Note that uploading the code can take up to a few minutes. Progress is indicated on the lower left corner of the screen.

13. Program FPGA chip. Above the FPGA chip icon, click on ‘Program FPGA’.