LASER INTERFEROMETER GRAVITATIONAL WAVE OBSERVATORY
-LIGO-
CALIFORNIA INSTITUTE OF TECHNOLOGY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Technical Note LIGO-T080083-00 01/04/09

Testing procedure for the Timing Master/FanOut board Advanced LIGO

Imre Bartos, Rolf Bork, Maxim Factourovich, Jay Heefner, Szabolcs Márka, Zoltán Raics, Paul Schwinberg and Daniel Sigg

This is an internal working note
of the LIGO Project.

California Institute of Technology
LIGO Project – MS 51-33
Pasadena CA 91125
Phone (626) 395-2129
Fax (626) 304-9834
E-mail: info@ligo.caltech.edu

Massachusetts Institute of Technology
LIGO Project, MIT NW22-295,
185 Albany St.,Cambridge, MA 02139 USA
Phone (617) 253 4824
Fax (617) 253 7014
E-mail: info@ligo.mit.edu

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

Board document LIGO DCC # **070011**

Board Revision **A**

Board Serial # _____

Board Type **Master [ ]**
( ) on-board GPS receiver
( ) OCXO included

FanOut [ ]

Test Engineer _____

Test Date: _____

Overall MFO board testing: **PASSED** **FAILED**

Signature: _____

Testing schedule:

1. Power Supply Unit (PSU) & on-board voltages
2. Flash PROM & JTAG Interfaces
3. Fiber I/O channels
4. The VCXO circuitry
5. Front-panel LEDs
6. Serial interface (RS422)
7. GPS receiver, antenna & interface*
8. Fiber-delay claculator
9. OCXO interface & PLL circuitry**
10. BNC ports

Physical requirements:

Hardware:

1	GPS-antenna*
2	JTAG interface board, rev A. (D060291) with 10-pin flat-ribbon cable and 26-pin flat-ribbon cable
3	12V, 2A PSU and 24V, 0.25A PSU and 2 cords with 3-pin D-sub connectors
4	Voltmeter
5	17 fiber-optic multi-mode (e.g. Agilent® HFBR 57E0) and at least 1 single-mode (e.g. Avago® AFCT-5760) transceiver modules
6	Windows®-operated PC with serial and parallel ports
7	RS232 to RS422 converter
8	Reference clocking (i.e. 1PPS) source w. optical output
9	An LC-LC fiber pair and a long (≥ 20m) optical fiber loop
10	BNC patch cable

Software:

1	MS Windows® 2000, XP
2	Altium® Designer v6.9
3	Motorola® WinOncore v1.2*
4	HHD Software Free Serial Port Monitor v3.31 http://www.serial-port-monitor.com/index.html
5	GPS clock from LIGOTools*

A brief introduction to the Master/FanOut board functionality is given in the Appendix.
More detailed description of the proposed timing system can be found online at
http://www.ligo.caltech.edu/docs/T/T070218-00/

---

*Master-board with GPS-module only

**Master-board with OCXO module only

1. Testing the on-board PSU voltages at various testpoints.

External Power Supply Unit:
manufacturer: _____

model: _____

Digital Multimeter:
manufacturer: _____

model: _____

(1) Measure the applied external voltages and write them below:

Nominal: 12.0V ± 1.0V Measured*: _____ At the board (TP9): _____
*no electrical connections to the MFO

(2) Measure the voltages at the testpoints and the current through the board and write them below:

TP5	*nominal*	3.0V ± 0.15V	*measured:* _____
TP6	*nominal:*	1.2V ± 0.06V	*measured:* _____
TP7	*nominal:*	5.0V ± 0.25V	*measured:* _____
TP8	*nominal:*	5.0V ± 0.25V	*measured:* _____
TP9	*nominal:*	12.0V ± 1.2V	*measured:* _____
TP10	*nominal:*	2.5V ± 0.12V	*measured:* _____
TP11	*nominal:*	10.0V ± 0.5V	*measured:* _____
TP12	*nominal:*	1.8V ± 0.09V	*measured:* _____
TP18	*nominal:*	-10.0V ± 0.5V	*measured:* _____
TP19	*nominal:*	3.3V ± 0.16V	*measured:* _____
The current:	*nominal:*	0.5A ± 10%	*measured:* _____

(3) Now, insert pluggable transceivers into all available slots simultaneously, measure the voltages and the current again and write them down below:

TP5	*nominal*	3.0V ± 0.15V	*measured:* _____
TP6	*nominal:*	1.2V ± 0.06V	*measured:* _____
TP7	*nominal:*	5.0V ± 0.25V	*measured:* _____
TP8	*nominal:*	5.0V ± 0.25V	*measured:* _____

TP9	*nominal:*	12.0V ± 1.2V	*measured:* _____
TP10	*nominal:*	2.5V ± 0.12V	*measured:* _____
TP11	*nominal:*	10.0V ± 0.5V	*measured:* _____
TP12	*nominal:*	1.8V ± 0.09V	*measured:* _____
TP18	*nominal:*	-10.0V ± 0.5V	*measured:* _____
TP19	*nominal:*	3.3V ± 0.16V	*measured:* _____
The current:	*nominal:*	1.5A ± 10%	*measured:* _____

All the measurements are within the nominal ranges: **( ) YES** **( ) NO**

---

2. Testing Flash PROM & JTAG interfaces

Use Altium® Designer;
bit-files: **fpgamasterfanout_cclk.bit** (for "Spartan3E XC3S1600E-4FG320C"),
fpgamasterfanout.mcs (for "XCF XCF08PFSG48C")

The procedure:

(a) Turn OFF the power to the MFO, wait a few seconds;
(b) Turn the power ON;
(c) Open Altium® Designer, go to Devices View, download the indicated bit-file and record the programming status of the FPGA;
(d) Open the "Spartan3E XC3S1600E-4FG320C" instrument (the FPGA core), click on JTAG viewer panel, check the "Live Update" box and watch the FPGA pin indicators; record the observations
(e) Right-click on the "XCF XCF08PFSG48C" instrument (the Flash PROM), select "Choose File and Download" and choose the indicated bit-file
(f) Upon completion of the download, turn off the power to the board, wait for a few seconds, then restore the power
(g) The FPGA should be automatically reprogrammed from the Flash PROM
(h) Indicate the success/failure of the reprogramming

FPGA programming status (circle one):	successful*	failed
FPGA pin indicators (circle one):	all steady	changing*
Flash PROM reprogramming (circle one):	successful*	failed

*indicates normal JTAG and/or PROM operation

---

3. Test fiber I/O channels

Use Altium® Designer;
download bit-file: **fpgamasterfanout_cclk.bit**

Altium® virtual instrument I/O reference:

FO_IO_ROUTER - routes the signal through chosen optical I/O FanOut (FO);

BIN[15..0] - monitors the active/inactive status of the FanOuts;
MONITOR - toggles when 1PPS signal is received;
INT_FGEN - when enabled, switches to internal frequency generator;
THRU_FIBER - when enabled, routes the 8MHz signal through the chosen FO;
CHOOSE_OUT[3..0] - allows to choose FO output;
CHOOSE_IN[3..0] - allows to choose FO input;

The instrument setup:
Enable THRU_FIBER (the indicator turns *green*)

DELAY_MEASURE - shows the time required for the signal to go through the loop of fiber connected to an optical I/O port;

CLK_CNT[31..0] counts the number of clock-cycles in one second
DEL_CNT[31..0] shows the propagation delay in units of clock-cycles
ALT_CNT[31..0] same function as DEL_CNT[31..0] but alternative implementation; may be used for verification
NO_SIGNAL displays the presence of the attached fiber-loop
LED_ON when enabled, sends the toggling 1PPS signal to the LED of the FanOut-1, overriding other signals for this LED.

The instrument setup:
Disable LED-ON, if enabled (the indicator turns *red*)

BNC_FIBER_FREQUENCIES - measures the frequency of the incoming signal through a given port;

COUNTER_CHANNEL_A the signal frequency of the BNC "1PPS output" port;
COUNTER_CHANNEL_B the signal frequency of the chosen FanOut port.

The instrument setup:
In the "Counter Options" set the Counter Time Base to 67.108 MHz.

The procedure:

(a) Open Altium® Designer; open FO_IO_ROUTER instrument; set the TEST_INPUT to '1' and the INT_GEN and the THRU_FIBER to '0';
(b) Open the BNC_FIBER_FREQUENCIES instrument in the Counter Options, set the Counter Time Base to 67.108 MHz; set the COUNTER CHANNEL B to the FREQUENCY mode;

(c) Observe and record the frequency displayed in the COUNTER CHANNEL B with and without reference clock source connected to the front-panel INPUT port; write the results in the table below;
(d) Connect the reference clock source to the INPUT port and a fiber-optic loop to the FO port #1;
(e) Confirm that the front-panel FanOut LEDs follow the values of CHOOSE_OUT[3..0] and CHOOSE_IN[3..0];
(f) Set both, CHOOSE_OUT[3..0] and CHOOSE_IN[3..0] to 0;
(g) In the space below, record the frequency observed in the COUNTER_CHANNEL_B of the BNC_FIBER_FREQUENCIES instrument;
(h) Circle PASS if the nominal and observed frequencies match or FAIL if they don't;
(i) Move the fiber-loop to the next port (FO #2);
(j) Increase the values of CHOOSE_OUT and CHOOSE_IN by 1;
(k) Repeat the recording;
(l) In the same fashion, proceed with the rest of the ports;
(m) Write the results in the provided table

Port #	Frequency while disconnected		Frequency while connected		PASS (match)	FAIL (mismatch)
	nominal	observed	nominal	observed		
INPUT	0 MHz		8.389 MHz			
FO #1	0 MHz		8.389 MHz			
FO #2	0 MHz		8.389 MHz			
FO #3	0 MHz		8.389 MHz			
FO #4	0 MHz		8.389 MHz			
FO #5	0 MHz		8.389 MHz			
FO #6	0 MHz		8.389 MHz			
FO #7	0 MHz		8.389 MHz			
FO #8	0 MHz		8.389 MHz			
FO #9	0 MHz		8.389 MHz			
FO #10	0 MHz		8.389 MHz			
FO #11	0 MHz		8.389 MHz			
FO #12	0 MHz		8.389 MHz			
FO #13	0 MHz		8.389 MHz			
FO #14	0 MHz		8.389 MHz			
FO #15	0 MHz		8.389 MHz			
FO #16	0 MHz		8.389 MHz			

All nominals were met: ( ) YES ( ) NO

4. Testing the VCXO circuitry:

Additional equipment necessary: voltmeter, reference clocking source

The following testing is done by measuring voltages at the VCXO-related testing points on the MFO board. The hardware-implemented VCXO phase-locking loop automatically synchronizes the on-board VCXO to the 8MHz signal from the front-panel INPUT port. The voltages at the testpoints indicate whether or not the proper synchronization takes place. ***Prior to taking the measurements, a master timing source MUST be connected to the front-panel INPUT channel.***

VCXO voltages:

	Without synchronization		With synchronization	
	nominal	measured	nominal	measured
TP1 (ERR)	**~10V**		**5.0 ± 0.5V**	
TP2 (CTRL)	**~0V**		**2V...8V**	

The voltages are within the nominal ranges: **( ) YES** **( ) NO**

---

5. Test front-panel LEDs.

This test checks the operational status of the front-panel LEDs.

Use Altium® Designer;
download bit-file: **fpgamasterfanout_cclk.bit**

Altium® virtual instrument I/O reference:

SERIAL_LED_ROUTER routes the signals between serial interfaces (on-board GPS receiver & RS422 port) as well as front-panel LEDs.

A[7..0]	not used
GPS_THRU	connects GPS and RS422 interfaces directly, bypassing all the FPGA logic
OCXO_GPS_TO_RS422	chooses between the OCXO and the GPS data to be sent through the RS422 port
TEST_LEDS	for LED testing purposes, overrides all other FPGA LED-driving circuitry and sets all the front-panel LEDs in test mode
LED_RATE[7..0]	sets the blinking rate for the front-panel LEDs

The procedure:

(a) Open Altium® Designer, download the bit-file to the FPGA
(b) Open the SERIAL_LED_ROUTER instrument
(c) Set TEST_LEDS flag to high (green)
(d) Increase the LED_RATE value from 0x00 to 0x3#, considering *only* the highest digit (# can be any number)
(e) Record the blinking rates in the provided table

Parameter	LED rate at 0x0#		LED rate at 0x3#	
	nominal	**observed**	**nominal**	**observed**
ON	steady		2/sec.	
1PPS	steady		2/sec.	
GPS	steady		2/sec.	
OCXO	steady		2/sec.	
INPUT	steady		2/sec.	
#1	steady		2/sec.	
#2	steady		2/sec.	
#3	steady		2/sec.	
#4	steady		2/sec.	
#5	steady		2/sec.	
#6	steady		2/sec.	
#7	steady		2/sec.	
#8	steady		2/sec.	
#9	steady		2/sec.	
#10	steady		2/sec.	
#11	steady		2/sec.	
#12	steady		2/sec.	
#13	steady		2/sec.	
#14	steady		2/sec.	
#15	steady		2/sec.	
#16	Steady		2/sec.	

All LEDs blink simultaneously: **( ) YES** **( ) NO**

6. Testing the serial interface – RS422 output

This test verifies the operational status of the on-board serial interfaces.

Use Altium® Designer, Motorola® WinOncore;
download bit-file: **fpgamasterfanout_cclk.bit**

Altium® virtual instrument I/O reference:

SERIAL_LED_ROUTER routes the signals between serial interfaces (on-board GPS receiver & RS422 port) as well as front-panel LEDs.

A[7..0]	not used
GPS_THRU	connects GPS and RS422 interfaces directly, bypassing all the FPGA logic
OCXO_GPS_TO_RS422	chooses between the OCXO and the GPS data to be sent through the RS422 port
TEST_LEDS	for LED testing purposes, overrides all other FPGA LED-driving circuitry and sets all the front-panel LEDs in test mode
LED_RATE[7..0]	sets the blinking rate for the front-panel LEDs

The procedure:

(f) Open Altium® Designer, download the bit-file to the FPGA
(g) Open the SERIAL_LED_ROUTER instrument
(h) Set GPS_THRU flag to high (green)
(i) Open WinOncore, open Command Monitor Window
(j) Open GPS-receiver setup wizard (refer to the WinOncore manual), set up the GPS-receiver with default parameters
(k) During the setup procedure, observe both, [TX] and [RX] data packets in the Command Monitor Window
(l) Upon completion of the setup, look for the long strings of data in the Command Monitor Window, with the "[RX]@@Ha" header
(m) Record the observations below

[TX] commands (circle one):	present*	absent
[RX] commands (circle one):	present*	absent
[RX]@@Ha headers (circle one):	present*	absent
GPS setup (circle one):	successful*	unsuccessful

---

*indicates normal JTAG operation

7. GPS antenna & receiver module test

Use Altium® Designer, Motorola® WinOncore, GPSclock from LIGOtools;
download bit-file: **fpgamasterfanout_cclk.bit**

Altium® virtual instrument I/O reference:

OSCILLATOR_FREQUENCIES - shows the frequencies of the onboard/external oscillators;

COUNTER CHANNEL A - GPS 1PPS signal frequency, nominally 1Hz
COUNTER CHANNEL B - The frequency of the external OCXO, in MHz.

The instrument setup:

(a) make sure each channel window is in the "Frequency" mode; to change the mode, press the Mode button below each screen as many times as necessary, until the windows gets into the desired mode.
(b) press the "Counter Options" button under the left screen; set up the Counter Time Base to 67.108 MHz.
(c) make sure the Run button is pressed, in which case it should have yellow color.

GPS_I0 - reads and displays various GPS-receiver data parameters;

Inputs:

GPS_SECOND[31..0] - The 32-bit value of the current GPS-second
VISIBLE[7..0] - The number of visible satellites
TRACKED[7..0] - The number of tracked satellites
GPS_STATUS[15..0] - GPS receiver status
ID1_TAG[31..0] - GPS-receiver ID-tag, highest 16 bits
ID2_TAG[31..0] - GPS-receiver ID-tag, lowest 16 bits
LOCKED - Confirms locking to GPS satellites
GPS_1PPS - Changes state each time the 1PPS is received
PPS_INDICATOR[15..0] - Scopes the GPS_1PPS line with 1/8 sec. resolution
ERROR - The software error status.

Outputs:

INITIALIZE - sends initialization signal
RS232_SEL - selects the source for the serial output
LOOPBACK - loops the output signal back to the software receiver
CLKD[31..0] - sets the clock division factor for the PPS_INDICATOR

The procedure:

(a) open Altium® Designer, download the bit-file, open the GPS_IO instrument
(b) set the INITIALIZE output to '0'
(c) set the RS422_SEL output to '1'
(d) set the LOOPBACK output to '0'
(e) make sure the CLKD[31..0] output is set to 0x800000 or 8,388,608 decimal.
(f) wait a few seconds for the GPS_1PPS and PPS_INDICATIOR input indicators to begin toggling
(g) wait for additional 15 minutes for the GPS receiver to acquire data
(h) open WinOncore and perform GPS self-test; fill-up the table below
(i) in Altium® Designer, open the GPS_IO instrument again and record the status in the provided table

GPS self-test results:

Parameter	Nominal value	Factual value
Antenna Status	OK	
RTC comm. & time	PASS	
Temperature Sensor	PASS	
RAM	PASS	
ROM	PASS	
1 kHz Presence	PASS	
Temp. Sensor Data Checksum	PASS	
Oscillator Data Checksum	PASS	
Manufacturing Data Checksum	PASS	
Channel 12 correlation test	PASS	
Channel 11 correlation test	PASS	
Channel 10 correlation test	PASS	
Channel 9 correlation test	PASS	
Channel 8 correlation test	PASS	
Channel 7 correlation test	PASS	
Channel 6 correlation test	PASS	
Channel 5 correlation test	PASS	
Channel 4 correlation test	PASS	
Channel 3 correlation test	PASS	
Channel 2 correlation test	PASS	
Channel 1 correlation test	PASS	

GPS_IO instrument status:

Parameter	Nominal value	Factual value
# of tracked GPS satellites	≥ 3	
# of visible GPS satellites	≥ (# of tracked)	
LOCKED value	TRUE	
Highest 3 bits of GPS_STATUS	111	
ERROR value	FALSE	
Value of GPS_SECOND	Use reference*	
GPS_SECOND discrepancy with the reference*	0	

All the nominal values have been met: **( ) YES** **( ) NO**

Notes:

8. Test fiber-delay calculator

Use Altium® Designer;
download bit-file: **fpgamasterfanout_cclk.bit**

Altium® virtual instrument I/O reference:

DELAY_MEASURE - shows the time required for the signal to go through the loop of fiber connected to an optical I/O port;

CLK_CNT[31..0]	counts the number of clock-cycles in one second
DEL_CNT[31..0]	shows the propagation delay in units of clock-cycles
ALT_CNT[31..0]	same function as DEL_CNT[31..0] but alternative implementation; may be used for verification
NO_SIGNAL	displays the presence of the attached fiber-loop
LED_ON	when enabled, sends the toggling 1PPS signal to the LED of the FanOut-1, overriding other signals for this LED.

The instrument setup:

(a) Open Altium Designer, download the bit-file, open the instrument from the Devices menu;
(b) Connect the longest possible fiber loop (up to $\mathbf{10^8}$ m) to the FanOut-port #**1**
(c) For the given fiber length, calculate* the nominals and write them in the spaces provided below
(d) Observe and record the actual measured delays

*For every **1000m** of the fiber, the measured delay shown in DEL_CNT[31..0] and ALT_CNT[31..0] should be about **370** units (i.e. clock-cycles). E.g. the measured delay for the 4km+ loop at LIGO-Hanford was 1489 at both counters.

For the fiber-loop of the _____ length:

CLK_CNT[31..0]: nominal: **67,108,854** measured:_____

DEL_CNT[31..0]: nominal: _____ measured:_____

ALT_CNT[31..0]: nominal: _____ measured:_____

Notes:

---

9. Testing the OCXO interface

Use Altium® Designer, HHD Software Free Serial Port Monitor;
download bit-file: **fpgamasterfanout_cclk.bit**

Altium® virtual instrument I/O reference:

FILTER_INPUT_SWITCH - controls the input of the OCXO-regulating filter.

OUT_VAL[31..0]	displays the combined output value of the COARSE[10..0], FINE[11..0] and SFINE[11..0] controls
FOUT[31..0]	graphically shows the output value of the OCXO-regulating filter
FONUM[31..0]	numerical equivalent of FOUT[31..0]
OVF	indicates the overflow status of the filter
IOLD[31..0]	reflects the error value previously stored in the filter
TO_DAC[15..0]	shows the value sent to the OCXO-controlling DAC
SIGN	shows the sign of the DAC value
O_F	indicates the overflow status of the DAC value
COARSE[10..0]	simulates the filter error input with full range but minimal resolution
FINE[11..0]	same as COARSE[10..0] but with medium range and resolution
SFINE[11..0]	same as COARSE[10..0] but with minimal range and highest resolution
INP_SEL	toggles the filter input between simulation (disabled, red) and actual (enabled, green) inputs
SUPER_GAIN[3..0]	applied additional gain (i.e. bit-shift) to the filter output (currently disabled and non-functional)
SG_BYPASS	allows bypassing the applied super-gain (currently disabled and non-functional)
RESET_FILTER	resets the memory within the filter
SYNC_OVERRIDE	forces synchronization of the VCXO to the optical input

OCXO_MONITOR - monitors various OCXO parameters.

OCXO_PRESENT	detects the active presence of the OCXO
ONEPPS_EXT	reflects the presence of an external 1PPS signal
ONEPPS_INT	toggles along with the internal 1PPS generator
OCXO_LOCKED	goes high whenever the OCXO error is below the threshold
OCXO_ERR[31..0]	displays the OCXO error value
FM[31..0]	monitors the input value of the OCXO filter engine
OCXO_CTRL[15..0]	displays the OCXO-adjusting value sent to the DAC circuit, 0x8000 corresponding to zero-adjustment
OCXO_DEV[15..0]	reflects the absolute value of the OCXO deviation from the reference clock in terms of clock-cycles per-second
OCXO_DEV_SIGN	GREEN indicates the POSITIVE sign of the OCXO deviation

The procedure:

(a) Open Altium® Designer, download the bit-file;
(b) Open the OCXO_MONITOR instrument
(c) Unscrew the OCXO manual adjustment cap on the OCXO chassis
(d) Using small flat screwdriver, adjust the OCXO so that the value in the OCXO_DEV[15..0] is no greater than 3. ***Be very gentle,*** *as the OCXO's internal adjustment knob is very easy to break. Use counter-clockwise motion for the OCXO_DEV to go towards the positive side.*
(e) Open the FILTER_INPUT_SWITCH instrument
(f) Set INP_SEL flag to HIGH (green); make sure the RESET_FILTER flag is LOW(red)
(g) Open Free Serial Port Monitor, go to Console Window
(h) Open a new connection (COM1, 9600-baud, 8-bits, no-parity)
(i) Allow ~20min for the filter to settle down, then go to the Port Monitor's Console Window and record the string value below

---

OCXO status serial data format:

The OCXO status string has the following structure:

[flags][hex_error][hex_correction][dec_error][dec_correction][hex_fiber_delay]

where the fields are separated by spaces.

Char. 1	OCXO LOCKED flag
Char. 2	FILTER OVERFLOW flag
Char. 3	field separator (space)
Char. 4	OCXO error sign
Char. 5..12	OCXO error value in hex base (8-digit number)
Char. 13	field separator (space)
Char. 14	OCXO control sign
Char. 15..18	OCXO control value in hex base (4-digit number)
Char. 19	field separator (space)
Char. 20	OCXO error sign
Char. 21..28	OCXO error value in decimal base (8-digit number)
Char. 29	field separator (space)
Char. 30	OCXO control sign
Char. 31..35	OCXO control value in decimal base (5-digit number)
Char. 36	field separator (space)
Char. 37..44	fiber-delay (clock-cycles in hex base; not used for OCXO testing)

Allowing ~20min. for the OCXO synchronization, fill in the table below:

OCXO LOCKED flag	nominal: 1	observed:_____
FILTER OVERFLOW flag	nominal: 0	observed:_____
OCXO error (decimal)	nominal: 0 ± 4	observed:_____

10. Testing BNC ports

Use Altium® Designer;
download bit-file: **fpgamasterfanout_cclk.bit**

Altium® virtual instrument I/O reference:

BNC_FIBER_FREQUENCIES - measures the frequency of the incoming signal through a given port;

COUNTER_CHANNEL_A the signal frequency of the BNC "1PPS output" port;
COUNTER_CHANNEL_B the signal frequency of the chosen FanOut port.

FO_IO_ROUTER - routes the signal through chosen optical I/O FanOut (FO);

BIN[15..0] - monitors the active/inactive status of the FanOuts;
MONITOR - toggles when 1PPS signal is received;
INT_FGEN - when enabled, switches to internal frequency generator;
THRU_FIBER - when enabled, routes the 8MHz signal through the chosen FO;
CHOOSE_OUT[3..0] - allows to choose FO output;
CHOOSE_IN[3..0] - allows to choose FO input;

The procedure:

(a) Connect the 1PPS_OUT and 1PPS_IN ports via coaxial cable;
(b) Open Altium® Designer, download the bit-file;
(c) Open the BNC_FIBER_FREQUENCIES instrument;
(d) In the "Counter Options" set the Counter Time Base to 67.108 MHz;
(e) Open the FO_IO_ROUTER instrument;
(f) Disable the THRU_FIBER (the indicator turns *red*);
(g) Record the listed values below

Value in COUNTER_CHANNEL_A: Nominal: 8.389 MHz

Displayed: _____

MONITOR indicator toggling rate: Nominal: 1 per second

Observed: _____

Notes:

Testing summary

For each test, indicate the results in the table below:

On-board voltages	PASS	FAIL	
VCXO circuitry	PASS	FAIL	
EEPROM & JTAG interfaces	PASS	FAIL	
Front-panel LED circuits	PASS	FAIL	
Serial interfaces	PASS	FAIL	
GPS antenna & receiver	PASS	FAIL	N/A
Fiber I/O channels	PASS	FAIL	
Fiber-delay calculator	PASS	FAIL	
OCXO interface	PASS	FAIL	N/A
BNC ports	PASS	FAIL	

Overall MFO board testing: **PASSED** **FAILED**

Test engineer: _____

Date: _____

Appendix

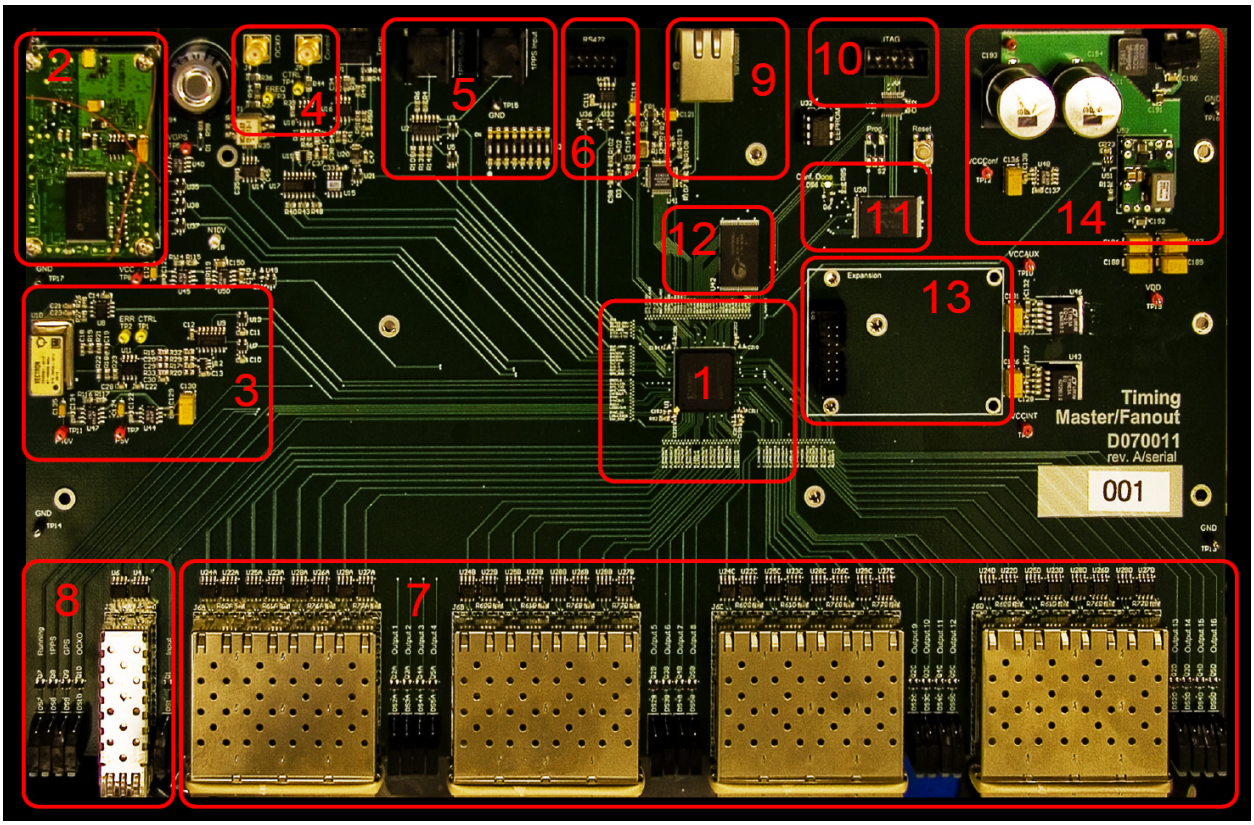

Figure 1. The Master FanOut board (MFO). (**1**) FPGA core chip. (**2**) On-board GPS-receiver. (**3**) **2^{26}Hz** voltage-controlled oscillator, used as internal clock. (**4**) The port for connecting external oven-stabilized **2^{26}Hz** oscillator. (**5**) BNC port for connecting to the external 1PPS source. (**6**) General purpose RS422 port. (**7**) 16 optical I/O FanOut channels. (**8**) Uplink optical I/O channel. (**9**) General purpose Ethernet port. (**10**) JTAG port for external programming of the FPGA. (**11**) Flash memory chip (used for programming FPGA). (**12**) RAM block. (**13**) Optional slot (currently used for mounting JTAG programming device). (**14**) The power supply unit.

The figure above shows the main blocks of the Master/FanOut board. The primary goal of the board is to acquire and distribute precise timing information with an accuracy of better than 1us, according to the requirements. To perform all the necessary calculations, we use a special logic element called Field-Programmable Gate Array, or FPGA (block #1 of the figure 1). The FPGA is in a sense a reduced version of a Central Processing Unit (CPU) alike those commonly found in computers. It uses the VCXO (block #3 on the figure) as its clocking device for the internal logic. While the VCXO is well suitable for driving the FPGA logic, its precision does not meet the 1us requirement. To overcome this, the VCXO is synchronized to a much more stable oven-controlled oscillator (OCXO) through the external port (block #4) which in its turn is synchronized to either the reference clocking signal received from the optical INPUT port (block #8), an alternative RF coaxial input (block #5) or the on-board GPS receiver (block #2). This given synchronization hierarchy is expected to hold the internal FPGA clock discrepancy within the 1us window.

The distinct feature of an FPGA is its ability to be programmed in a way similar to an ordinary CPU. The programming environment in our case is the PC-based Altium® Designer software package. The communication between the PC and the FPGA is done through the parallel port of the computer and the JTAG port of the Master/FanOut board (block #10). The programming code is stored in the flash-ROM module (block #11). All the communications with the PC, other than programming, are routed through either a serial port (block #6) or Ethernet interface (block #9, not tested here). The distribution of timing information to the devices other than PC goes through the 16 optical FanOut ports (block #7).

For more information, look the online documentation posted at
http://www.ligo.caltech.edu/docs/T/T070218-00/