OPERATING MANUAL

gigaBERT-660 Tx
700 MHz BIT ERROR RATE TRANSMITTER

MICROWAVE LOGIC, INC.

20 Cummings Road
Tyngsboro, MA 01879

Rev 2.3 7/3/90

SERVICE INFORMATION

WARRANTY

All Microwave Logic products are warranted against any defects in material and workmanship for the period of one year from the data of delivery. Microwave Logic will repair or replace products which prove to be defective during the warranty period.

NO OTHER WARRANTY, EXPRESSED OR IMPLIED, INCLUDING FITNESS FOR PURPOSE, MERCHANTABILITY OR OTHERWISE IS GIVEN.

SERVICE ASSISTANCE

For repair information and service assistance please contact Microwave Logic Customer Service department at (508) 649-6099. The gigaBERT-660's high frequency design, and critically timed coaxial delay lines make field service difficult. The factory's experience and expertise will save valuable time in evaluating and correcting any unit malfunction.

REPAIR RETURNS

If the unit must be returned to the factory for repair please do the following:

1) Contact the factory for a Return Authorization Number (RAN), at (508) 649-6099. The use of a RAN will ensure prompt repair, and ease tracking of your unit.

2) Pack the unit in it's original packing materials or other suitable material, such as foam or bubble-wrap, to safely ship the unit. Pack unit in a double-walled carton, and seal the carton with suitable tape. Display the RAN number on the outside of the carton.

3) Ship unit to:

   Microwave Logic, Inc.
   20 Cummings Road
   Tyngsboro MA 01879

   Attn: Customer Service

   Phone (508) 649-6099

gigaBERT-660

TABLE OF CONTENTS

TABLE OF CONTENTS
(cont)

CAUTIONS AND WARNINGS

The following cautions and warnings should be read before operating the unit.

AC POWER SOURCE

The BERT-660 is grounded through the AC power cord. Attach only to a properly grounded outlet. Do not operate the unit without this ground, as all metal parts of the unit become a potential shock hazard.

STATIC SENSITIVE DEVICE NOTICE

The gigaBERT-660 Tx output amplifiers are of GaAs FET design, and are susceptible to damage from overdriven DC voltages, and from electrostatic damage. Never apply reverse voltage to the DATA or CLOCK outputs, or voltages outside the range specified in section 2 of this manual.

SMA CONNECTOR CARE

The SMA connectors used on the gigaBERT-660 units require care while attaching cables, to prevent damage to the connector or attaching cable. When attaching a cable turn the cable nut, not the cable. Tighten the nut to 10 lb-in maximum.

OUTPUT AMPLIFIER POWER CYCLE SEQUENCE

During unit power cycling the Tx output amplifiers may saturate to their specified positive or negative rail; (+2V, or -2V) for a minimum of 200 milli-seconds. If this condition is harmful to user circuitry the DATA and CLOCK signals should be disconnected from the circuitry before power cycling.

QUICK START

HOW TO RUN THE gigaBERT-660 Tx (with a gigaBERT-660 Rx or DRx data receiver), WITHOUT READING THIS MANUAL

REQUIRED EQUIPMENT

1 - gigaBERT-660 Rx, gigaBERT-660 DRx, or other Data Receiver unit

2 - pairs of equal-length 50 ohm coax cables, preferably SMA, otherwise you'll need a lot of adapters.

BEGINNERS BERT

1) Connect a 50 ohm coaxial cable from the Tx CLOCK output to the Rx CLOCK input, or
Connect two equal-length 50 ohm coaxial cables from the Tx CLOCK outputs to the DRx CLOCK inputs.

2) Connect a 50 ohm coaxial cable from the TX DATA output to the Rx DATA input, or
Connect two equal-length 50 ohm coaxial cables from the TX DATA outputs to the DRx DATA inputs.

3) Turn on Tx and Rx/DRx. Set Tx Output Amplitude to 2V, and baseline Offset to -1V. See below for Tx control information.

4) Select 'AUTO-SEARCH' mode on the receiver (AUTO-SEARCH LED should be on). Auto-search will select code, input level, polarity, and phase delay.

5) Select 'PRBS' mode on Tx (PRBS LED should be on).

6) Push Display reset and Error history reset keys on receiver.

7) You are now making BER, albeit on a piece on coax cable but BER just the same.

8) The unit's battery backed-up RAM will remember unit status, clock and WORD memory contents and where you left off when power failed.

9) Hook up a parallel printer to the printer port, GPIB port, or a serial printer to the serial port, push F4. An immediate unit status report will be printed for posterity, on one of these ports. To redefine which port prints see section 3 for PRINT PORT menu.

QUICK START (cont)

ADVANCED TX TOPICS

1) To modify frequency of test;

Press 'FREQ' on Tx (FREQ LED should be on).

Change the frequency on Tx (use up/down CLOCK keys).

2) To modify data pattern of test;

Change the code on Tx (use up/down PATTERN keys).

3) To modify Tx Output Data levels;

Press 'DATA' on Tx (DATA LED should be on).

Change the AMPlitude (use up/down keys)

Change the OFFset (use up/down keys)

4) To modify Tx Output Clock levels;

Press 'CLOCK' on Tx (CLOCK LED should be on).

Change the AMPlitude (use up/down keys)

Change the OFFset (use up/down keys)

5) To inject errors into data stream;

Press Error Inj 'Rate' to inject an Error Rate, press again to change rate,

Push single to shut off error rate injection (if selected), or to inject a single error.

6) To change GPIB address;

Press GPIB key, change address with INPUT up/down keys.

7) To change CLOCK memory location;

Press INPUT RECALL key, change memory location with INPUT up/down keys.

8) To change WORD memory location;

Press WORD key, press PATTERN RECALL key, change memory location with PATTERN up/down keys.

QUICK START (cont)

ADVANCED RECEIVER TOPICS

1) BER (Bit Error Rate) Display clear;

Press ERROR DETECT Clear key to clear BER display.

2) BER Display select;

Press ERROR DETECT DISPLAY key repetitively to select between Totalize, Window, and Test mode.

3) Input Phase delay adjust;

Press INPUT DELAY key, change delay with INPUT up/down keys.

4) Input Threshold adjust;

Press INPUT V-THRS key, change threshold with up/down keys.

5) To modify data pattern of test;

Press PRBS key, change pattern with PATTERN up/down keys.
NOTE: in 'AUTO SEARCH' mode the receiver will automatically select the correct code.

6) To change GPIB address;

Press GPIB key, change address with INPUT up/down keys.

7) To change AUTO SYNC mode;

Press AUTO SEARCH key, in AUTO mode (AUTO LED on) unit selects code, threshold, polarity, and delay, in MANUAL mode (LED off), user must select parameters. DISABLE key turns off synchronization circuitry.

8) To change WORD memory location;

Press WORD key, press PATTERN RECALL key, change memory location with PATTERN up/down keys.

9) To select Reference Data mode;

Press EXT REF DATA key, unit now compares data pattern at REF DATA input with pattern at DATA input, suitable for custom (non-standard) patterns.

10) To clear ERROR HISTORY LEDS;

Press ERROR HISTORY CLEAR key.

SECTION 1

GENERAL DESCRIPTION

1.1 INTRODUCTION

This manual provides the user with the information required to set up and operate the Microwave Logic gigaBERT-660 Tx (Transmitter) and the gigaBERT-660 Rx (Receiver). As a system, the pair can be used to perform Bit Error Rate Tests (BERT).

1.2 GENERAL DESCRIPTION gigaBERT-660 TRANSMITTER (Tx)

The Microwave Logic gigaBERT-660 Tx is a 700 MHz Pseudo-random and Programmable WORD Pattern Generator, designed for system and device evaluation in the frequency range of 1 MHz to 700 MHz, with 100 KHz resolution (1 KHz resolution optional). The integrated crystal locked clock source, 16 bit (32K bit optional) programmable WORD, 5 PRBS patterns (2^n-1, n=7, 15, 17, 20, and 23) and complimentary, programmable output Data and Clock amplifiers, enable a wide range of tests to be performed on high-speed components and systems.

A microprocessor is utilized to program the PLL clock source, WORD patterns, output level controls, and provide a user-friendly front panel and display. The gigaBERT-660 Tx supports both RS-232C and GPIB interfaces for remote control, allowing test repeatability and output of unit status and memory contents. The battery backed-up non-volatile RAM memory provides storage of 10 WORD patterns, 10 clock frequencies, and unit setup for future recall.

Front panel outputs provide CLOCK/4 and Pattern Sync for eye-diagram and waveform analysis on an oscilloscope.

Rear panel inputs for External Error inject, and Data Inhibit allow user defined variations to the generated data streams.

The gigaBERT-660 Tx is supplied in a 6x14x12 stand-alone cabinet, and is provided with an adjustable carrying handle for easy bench top angle adjustment. The supplied front panel cover protects the unit during transit.

SECTION 2

gigaBERT-660 Tx SPECIFICATIONS

INTERNAL CLOCK SOURCE

Frequency range:	1 MHz to 700 MHz
Step size range:	0.1, 1, 10, 100 MHz (.001, .01 optional)
Resolution:	100 KHz (1 KHz optional)
Stability:	30 ppm (10 ppm optional)
Frequency memory:	10 frequencies

EXTERNAL CLOCK INPUT

Frequency range:	1 MHz to 700 MHz
Input level:	500 mV to 1.5 Vp-p
Impedance:	AC coupled into 50 ohms
Unit prop delay:	15 nS, +/- 500 pS.
Connector:	SMA

DATA PATTERNS

Format:	NRZ-L, Normal and Complement
PRBS pattern:	2^n-1, n=7, 15, 17, 20, 23
WORD length:	16 bits (32K optional)
WORD Order:	Transmit LSBit or MSBit First
Pattern memory:	Ten 16 bit WORDs (Ten 32K WORDs optional)

DATA OUTPUTS

Amplitude:	Variable; 700 mV to 2.0 V, 50 mV steps
Baseline offset:	Variable; -2.0 V to +1.0 V, 50 mV steps
Pulse top limit:	+2 V into 50 ohms, +4 V open load
Rise/fall time:	200 ps max. (20-80%) at 1 V amplitude
Jitter:	100 pS p-p max.
Source impedance:	50 ohms
Output timing:	CLOCK/DATA edge aligned (+/-100 ps)
Connectors:	SMA
Data Inhibit:	Rear panel BNC, ECL (50 ohms to -2 V term).
Data Inhibit rate:	Asynchronous, 1 bit + 500pS minimum width.
Data Invert:	Front Panel select

CLOCK OUTPUTS

Format:	True and Complement
Amplitude:	Variable; 700 mV to 2.0 V, 50 mV steps
Baseline offset:	Variable; -2.0 V to +1.0 V, 50 mV steps
Pulse top limit:	+2 V into 50 ohms, +4 V open load
Rise/fall time:	200 ps max. (20-80%) at 1 V amplitude
Jitter:	100 pS p-p max.
Source impedance:	50 ohms
Connectors:	SMA

gigaBERT-660 Tx SPECIFICATIONS (cont)

ERROR INJECTION

Internal rates:	Single or 1E-n, n=3, 4, 5, 6, 7
External:	ECL, 50 ohms to -2V, 1 error/rising edge
Injection rate:	50 nS minimum pulse width.
Connector:	Rear panel BNC

AUXILIARY OUTPUTS; Clock/4, Pattern Sync

Level:	800 mV p-p, +/- 250 mV
Impedance:	50 ohms
Clock/4 output:	system clock/4
Pattern sync:	1 bit-wide pulse per frame
Connectors:	SMA

RS-232C and GPIB INTERFACES

Controlled functions:	Remote control of all front panel functions except POWER, and PANEL LOCK.
Read back functions:	Read back of 10 clock frequencies and 10 data patterns stored in non-volatile memory, unit operating frequency, clock source status, pattern select.
GPIB EOS character:	LF (line feed) (0A hex)
GPIB Address:	Front panel select 0-30, or OFF-BUS.

AC POWER REQUIREMENTS

Voltage range:	115 VAC, 230 VAC, (+/-10%) factory select
Frequency range:	47-63 Hz
Power:	100 VA max.
Fuse rating:	115 VAC; 2 Amp; 230 VAC 1 Amp.
Operating range:	0 to 50 degrees C

MECHANICAL

Weight:	10Kg (22 lbs)
Size:	152H x 366W x 340D mm (6" x 14.4" x 13.4")

OPTIONS

Op-CT1:	1KHz resolution clock source
Op-MT1:	32K bit programmable WORD
Op-MT2:	32K bit programmable WORD, with 32K factory programmed ROM pattern
Op-RM1:	19" Rack Mount adapter

SECTION 3

gigaBERT-660 TRANSMITTER OPERATING INSTRUCTIONS

3.1 AC POWER SET UP.

The AC power connector on the rear panel accepts nominal input voltage requirements for the original destination, (either 115 VAC, or 230 VAC, +/- 10%). This input voltage is set at the factory. Contact the factory for information on changing the operating voltage. To change the line fuse, slide the fuse cover upwards and remove the fuse. Install correct value fuse (2 AMP SLO-BLO for 115 VAC, 1 AMP SLO-BLO for 230 VAC operation) into holder. Close the fuse cover, plug in the line cord. The unit is now ready for power up.

3.2 UNIT MOUNTING

The gigaBERT-660 Tx is designed to be placed flat on a level surface, capable of supporting it's weight, or angled from the surface with the unit's feet or rotating carrying handle. To alter the handle's orientation, press both handle locking buttons, rotate the handle to the desired angle, release the buttons; the handle will click into a locked position. Assure the handle is locked before placing the unit on the work surface. A Rack Mount Option is available for installation of the unit into a 19" rack.

3.3 UNIT COOLING REQUIREMENTS

The rear panel fan openings and top mounted ventilation slots must be kept clear for proper cooling of the unit. Allow a minimum of 2 inches of rear panel clearance, and 1 inch of top clearance, while operating the unit.

3.4 FRONT PANEL

The gigaBERT-660 Transmitter front panel is divided into eight functional control areas, refer to Figure 3.1, gigaBERT-660 Tx Front Panel. The following sections define all controls and indicators, and describes their operation. The front panel controls allow modification to the unit's operating status, and provides recall and storage access to the 10 frequency and 10 programmable WORD memories.

The unit's operating status is shown on the 2 row by 24 column Liquid crystal display (LCD), and by various key mounted LED indicators. Refer to Figure 3.2 for the LCD display fields associated with each function.

3.4.1 POWER

This section controls unit power.

POWER — Unit power switch. This rocker switch turns on the unit. Upward is 'ON', downward is 'OFF'.

3.4.2 OPERATING MODE

VIEW ANGLE — View control key. Pressing this key adjusts the LCD view angle, to provide adjustment in extreme lighting and viewing conditions.

PANEL LOCK — Panel Lock key. Pressing this key LOCKS the front panel, disabling all other keys. This prevents accidental change to the front panel operating status for long-term tests. To return the unit to normal operating status, press PANEL LOCK a second time.

An illuminated PANEL LOCK LED indicates the front panel is locked out from operation.

GPIB ADDR — GPIB Address select key. Pressing this key selects GPIB address mode, and displays the message 'GPIB xx' on the left bottom row of the LCD display. The GPIB address is programmable over the range 0 to 30 or OFF-BUS, with the CLOCK up/down keys. When the desired GPIB address is selected press the key a second time.

An illuminated GPIB ADDR LED indicates the GPIB Address can be modified with the CLOCK up/down keys.

LOCAL — Pressing this key will 'return to local' the control of the unit, when selected by the GPIB remote controller. Only the GPIB controller can select 'REMOTE' mode. NOTE if the GPIB bus command 'local lock out' has been sent by the controller, only the GPIB controller can unlock the front panel, or cycle unit power, which defaults to 'LOCAL' condition.

An illuminated REMOTE LED indicates the unit is in remote mode, and will not respond to front panel keys, until the LOCAL key has been pushed.

3.4.3 CLOCK

This section controls the programmable clock source, selects between the internal and external clock input, and accesses the clock memory.

Selecting FREQ, RECALL or SAVE allows the CLOCK up/down keys to modify the existing operating status, holding the up/down key repeats the function 5 times a second. Each key is described below. The STEP key directly modifies the unit's step size.

FREQ

Pressing FREQ selects frequency mode. The internal clock frequency is modified by changing the desired digit with the CLOCK up/down keys. The current operating frequency is displayed on the upper left row of the LCD.

An illuminated FREQ LED indicates the unit's frequency is modified by the CLOCK up/down keys.

STEP

Pressing STEP selects the digit to modify with the up/down keys in FREQ mode. The digit is marked with an underscore on the LCD display. The underscore (and step size) moves one digit to the right each time STEP is pushed. The selected step size is varied with the CLOCK up/down keys.

RECALL

Pressing RECALL selects Clock memory mode. 10 programmed clock frequencies can be stored and recalled. The selected clock memory location is displayed on the lower left side of the LCD display as FREQ m. The memory location m (0 to 9) is selected with the CLOCK up/down keys. The frequency stored in memory is output when selected.

An illuminated RECALL LED indicates the clock memory location can be selected by the CLOCK up/down keys.

SAVE

Pressing SAVE selects clock save mode. The currently programmed clock frequency can be stored in the current memory location, by pressing SAVE a second time; or select a different memory location with the CLOCK up/down keys, followed by pressing SAVE a second time. Pressing any other key will escape from SAVE mode.

An illuminated SAVE LED indicates the clock SAVE mode is selected and the clock memory location can be selected by the up/down keys.

SAVE (cont) For example to modify the programmed clock frequency stored in memory location 6;

1) select memory location 6 (RECALL, up/down keys),
2) modify frequency (FREQ, STEP, up/down keys),
3) press SAVE twice.

Alternatively:

1) modify the frequency (FREQ, STEP, up/down keys),
2) press SAVE,
3) select memory location 6 (up/down keys),
4) press SAVE a second time.

The new clock frequency is now stored in memory location 6 for future recall.

EXT CLK Pressing EXT CLK selects between the internal clock, and external clock signal applied at the CLOCK 'INPUT' connector, and displays 'EXT CLK' on the upper left row of the LCD. While EXT CLK mode is selected the other CLOCK keys are locked out, press EXT CLK to return to internal clock mode.

An illuminated EXT CLK LED indicates the EXTERNAL CLOCK mode is selected.

INPUT (SMA) The clock INPUT SMA connector is used to operate the gigaBERT-660 Tx with an External clock source. The input signal must be within the specified frequency and level range as noted in Section 2, Specifications.

3.4.4 PATTERN

This section controls the internal pattern generator, selects between PRBS and programmable WORD pattern, programs pattern and word length, and accesses the WORD Memory.

Selecting PRBS, WORD, RECALL, SAVE, or WORD LENGTH, allows the PATTERN up/down keys to modify the existing operating status, holding the up/down key repeats the function 5 times a second. Each sub-menu is described below.

The programmable WORD is modified from the front panel in 8 bit bytes. The 8 'BIT' keys allow direct modification to the selected byte. The CLEAR and SET keys allow all 8 bits of the selected byte to be set to a logic '0' or '1' respectively. The WORD LENGTH key allows selecting the programmable word to lengths of 8 or 16 bits, (optional up to 32K length).

PRBS

Pressing PRBS selects Pseudo-Random Binary Sequence (PRBS) mode. 5 PRBS patterns (2^7-1, 2^15-1, 2^17-1, 2^20-1, and 2^23-1) can be selected with the PATTERN up/down keys. When selected the pattern is denoted on the LCD screen as PN[x] where x=7,15,17,20,or 23.

An illuminated PRN LED indicates that PRBS Pattern is currently generated and output, and the desired pattern can be selected with the PATTERN up/down keys.

WORD

Pressing WORD selects programmable WORD mode. The programmable WORD can be modified, recalled, or stored in memory. The programmable pattern's byte address can be selected with the PATTERN up/down keys. The byte to modify with the 'BIT' keys is denoted on the LCD screen by an arrow.

An illuminated WORD LED indicates the programmable WORD pattern is currently generated and output, and the byte to modify with the BIT keys can be selected with the PATTERN up/down keys.

The programmable WORD is modified through the front panel in 8 bit bytes. The number of available bytes is dependent on WORD length (16 bits standard, up to 32K bits optional). WORD length is selected with the WORD LENGTH key, and displayed on the LCD.

WORD (cont) Both 8 bit bytes of the 16 bit WORD are displayed in HEX in the LCD. The desired byte to modify (with the 'bit' keys) is selected with the PATTERN up/down keys, and marked by an arrow. The binary value of the byte is displayed on the row of 8 'BIT' keys. Each bit's value is displayed by the key's LED; an illuminated key signifies a '1', an unlit key a '0'. Pressing the 'BIT' key toggles it's logic value. For example to modify bit 13;

1) select WORD mode (press WORD key)
2) select byte to modify (pattern up/down keys)
3) toggle BIT 5 key (13th bit).

RECALL Pressing RECALL selects WORD memory mode. 10 programmed WORDs can be stored and recalled. The selected WORD memory location is displayed on the lower left side of the LCD display. The memory location (0 to 9) is selected with the PATTERN up/down keys. The data pattern stored in memory is output immediately upon selection.

An illuminated RECALL LED indicates the WORD memory location can be selected by the PATTERN up/down keys.

SAVE Pressing SAVE selects WORD save mode. The currently programmed data pattern memory location is displayed on the lower left side of the LCD as 'WORD m', which will blink in save mode. The data pattern can be stored in the current memory location, by pressing SAVE a second time. A different memory location can be first selected with the PATTERN up/down keys, followed by pressing SAVE a second time. The 'WORD m' will stop blinking when saved. Pressing any other key will escape from SAVE mode.

An illuminated SAVE LED indicates WORD SAVE mode is selected and the WORD memory location can be selected by the PATTERN up/down keys.

3.4.4 PATTERN (cont)

SAVE (cont)

For example to modify the programmed data pattern stored in memory location 6;

1) select memory location 6 (RECALL, up/down keys),
2) modify WORD pattern (WORD, up/down keys, BIT keys),
3) press SAVE twice.

Alternatively:

1) modify the WORD pattern (WORD, up/down, BIT keys),
2) press SAVE,
3) select memory location 6 (up/down keys),
4) press SAVE a second time.

BITs 1-8 — Pressing a BIT key, toggles the logic value of the bit, of the selected 8 bit byte between a 0 and 1.

An illuminated LED indicates that the bit is a logic '1', unlit indicates a logic '0'.

CLEAR — Pressing CLEAR resets all 8 bits of the selected byte to a logic '0'.

SET — Pressing SET sets all 8 bits of the selected byte to a logic '1'.

WORD LENGTH — This key selects WORD length select mode. The programmable WORD length is variable between 8 or 16 bits, with the PATTERN up/down keys, and is displayed on the LCD to the right of the the WORD byte values. The new length data pattern will be output, starting with byte 0 through the respective highest byte (bytes 0 and 1 for 16 bit WORD, up to 32K bits optional), dependent on WORD length.

3.4.4.1 FUNCTION KEYS

F1 — User-menu access key, see section 3.5.

F2-F4 — Undefined.

3.4.5 ERROR INJECT

RATE

Selects internal error injection rate, over the range; 1E-n where n= 7,6,5,4,3. Pushing the key selects between the ranges, from 7 to 3. Pushing the key a sixth time selects EXTERNAL error inject mode allowing errors to be injected via the rear panel mounted BNC connector. Pressing the key a seventh time shuts off INTERNAL error inject mode.

Pressing SINGLE disables internally generated errors, pressing RATE again, returns the unit to the previously selected error inject rate.

An illuminated RATE LED indicates that errors are being injected into the data stream from the internally generated error rate circuitry.

SINGLE

Pressing this key injects a single bit error into the output data stream.

The ERROR INJ LED flashes as the errors are injected into the data stream.

3.4.6 CLOCK AND DATA OUTPUT CONTROLS

This section contains the controls for Clock and Data output levels, and Data polarity.

CLOCK — This key selects Clock output level control. When selected the AMPL up/down keys, and OFFSET up/down keys, control the clock output.

An illuminated CLOCK LED indicates that the Clock Output levels can be modified with the up/down keys.

DATA — This key selects Data output level control. When selected the AMPL up/down keys, and OFFSET up/down keys, control the Data output.

An illuminated DATA LED indicates that the Data Output levels can be modified with the up/down keys.

AMPL UP/DOWN — These keys increment/decrement the Output clock and data amplitude by 50 mV. The output amplitude is displayed on the upper right side of the LCD. See section 3.4.6.1 for pulse top/bottom maximum range.

OFFS UP/DOWN — These keys increase/decrease the selected clock/data output baseline offset by 50 mV. The output Baseline offset is displayed on the lower right side of the LCD. See section 3.4.6.1 for pulse top/bottom maximum range.

DATA INV — This key inverts the Output Data signal.

3.4.6.1 OUTPUT AMPLIFIER APPLICATION NOTE:

1) 50 ohm LOAD;

The Output amplifiers will drive a 50 ohm load to 2 V peak-to-peak maximum. The maximum pulse top is +2 VDC, if the baseline is raised above 0V, maximum possible Amplitude is reduced respectively.

2) Unterminated, Rterm > 2Kohms;

The Output amplifiers will drive an unterminated load to 4 VDC peak-to-peak maximum. The maximum pulse top is +4 VDC, if the baseline is raised above 0V, maximum Amplitude is reduced respectively.

3) Output termination voltage;

The offset is displayed and calibrated as specified into a 50 ohm to GND load. Terminating the signal to other voltages will introduce a DC shift in offset, which must be accounted for.

3.4.7 CLOCK AND DATA OUTPUT SIGNALS

These six outputs provide differential Data and Clock outputs, and signals for oscilloscope trigger, external interface including mux/demux systems, and system clock distribution.

DATA — True NRZ-L data is output from this SMA connector. Amplitude and Offset are adjustable as previously described. Source impedance is 50 ohms.

$\overline{\text{DATA}}$ — Inverted NRZ-L data is output from this SMA connector. Amplitude and Offset are adjustable as previously described. Source impedance is 50 ohms.

CLOCK — True Clock is output from this SMA connector. Amplitude and Offset are adjustable as previously described. Source impedance is 50 ohms.

$\overline{\text{CLOCK}}$ — Inverted Clock data is output from this SMA connector. Amplitude and Offset are adjustable as previously described. Source impedance is 50 ohms.

PATTERN SYNC — This SMA connector outputs a signal suitable for oscilloscope trigger, occurring once per pattern frame.

CLOCK/4 — This SMA connector output provides a system clock/4 signal suitable for oscilloscope trigger for analysis of output data eye-diagrams, and clock/4 distribution requirements.

3.5 USER-MENUS (FUNCTION KEYS F1-F4)

Function key F1 provides access to the USER-MENUS. The Function keys F1-F4 are re-assigned in user-menu mode to allow convenient front-panel programming of a variety of features, and to provide control for unit options. Refer to Section 2 for a description of available options. Function keys F2-F4 are defined during user-menu mode as tabulated below. The menus are structured as follows;

3.5.1 USER-MENU STRUCTURE

ESCAPE KEY (F1)

Menu key F1 is the MENU ESCape key, functions as follows (sub-menu dependent);

1) Enter USER-MENU mode,
2) Provide MORE selections,
3) ESCAPE from an editing level, or
4) EXIT from menu mode.

F1 will either escape or access more menus, pushing repetitively will eventually EXIT menu mode, without changing previously set conditions.

EDITING KEYS (F2, F3, F4)

Menu keys F2, F3, and F4 are editing keys. Their functions are re-assigned for each sub-menu, as tabulated throughout this section.

3.5.2 USER-MENU SELECTION

The USER-MENU structure consists of two top-level menus entitled;

'MAIN', and 'MORE'.

The function key access sequence to each sub-menu, follows the Menu title in brackets []. Each top level menu is structured into 4 sub-menus.

Following the sub-menu names below are the sections in this manual where the detailed descriptions can be found.

3.5.3 USER-MENU REFERENCE TABLE

MAIN user-menu

Pressing F1 from normal operating mode, accesses 'USER-MENU mode'. Normal unit operation continues while in user-menu mode, and is displayed as;

F1	F2	F3	F4
MORE	RS232	GPIB	WORD

F1 -- accesses MORE menu, see below,

F2 -- selects RS-232C user-menu, section 3.5.4,

F3 -- selects GPIB user-menu, section 3.5.5,

F4 -- selects WORD user-menu, section 3.5.6,
NOTE: see OPTION OP-MT1 section if 32K WORD option installed.

Pressing 'F1' from above MENU accesses a sub-menu as;

F1	F2
EXIT	UTIL

F1 -- EXITs from user-menu mode.

F2 -- selects UTIL user-menu, section 3.5.7

A selected sub-menu will appear as follows;

F1:ESC F4:SET
VALUE: PPPPP

F1 Escapes from the sub-menu without change.

F4 Sets the selected VALUE as noted 'PPPPP'

The Value is selected with the PATTERN up/down keys.

3.5.4 RS232 user-menu

The RS232 menus provide user control of the following RS-232C Remote Interface parameters;

BAUD	--	Serial BAUD rate (300 to 9600)
PARITY	--	Serial byte PARITY (NONE, ODD, EVEN)
SIZE	--	Serial byte SIZE (7 or 8 bits)
EOL	--	End Of Line character (CR, LF, CR/LF, LF/CR)
XON/XOFF	--	XON/XOFF protocol (ON or OFF)
ECHO	--	ECHO of command to terminal (ON or OFF)

Pressing F2 from MAIN menu, selects the menu;

F1	F2	F3	F4
MORE	BAUD	PARITY	SIZE

Pressing F1 from the above menu selects the menu;

F1	F2	F3	F4
ESC	EOL	XON/XOFF	ECHO

BAUD allows selection of BAUD Rate among the values of (300, 600, 1200, 2400, 4800, 9600).

PARITY allows selection of PARITY (NONE, EVEN, ODD).

SIZE allows selection of RS-232 word SIZE among the values of (7, 8).

EOL allows selection of EOL character among the values of (LF, CR, CR/LF, LF/CR).

XON/XOFF allows activation (ON) or deactivation (OFF) of XON/XOFF protocol.

ECHO allows activation (ON) or deactivate (OFF) RS-232 Echo feature, which echoes back to the terminal all characters entered.

3.5.5 GPIB user-menu

The GPIB menu provide user control of the following GPIB (IEEE-488.2) Remote Interface parameters;

EOI -- Response Message Terminator sent by the gigaBERT-660 in TALK mode.

Pressing F3 from the MAIN menu:

EOI allows selection of the EOI Terminator among the values (EOI, EOI/LF). The EOI character is selected with the PATTERN up/down keys. F4 sets the value, F1 escapes without change.

3.5.6 WORD user-menu

The WORD menu provides control of the 16-bit WORD transmission order;

ORDER -- Transmit LSB or MSB of each byte first

Pressing F4 from the MAIN menu selects the menu;

F1	F2
ESC	ORDER

ORDER allows selection between transmitting the 16-bit WORD starting with the MSB of the first byte or the LSB of the first byte (MSB/LSB). After the MSB (or LSB), the bits would be transmitted continuing with the bits of the first byte until the LSB (or MSB) is reached, then proceeding with the next byte in the same order.

NOTE: The selected ORDER is effective only on the pattern bytes edited after the setting of the ORDER. If the ORDER is changed in the middle of programming pattern bytes, the bytes programmed before the ORDER change will be reversed.

For example, if the word ORDER is previous LSB at word location 6 and the data pattern is programmed as follows:

```
edit_begin  6
byte_length 4,0
byte_block  0,16,#HAA,#HAA
word_order  MSB
byte_block  2,16,#HAA,#HAA
edit_end    6
```

The pattern at location 6 would be:

```
byte_block  0,32,#H55,#H55,#HAA,HAA
```

3.5.7 UTILITY user-menu

The UTILITY menu provides convenient access to the following system information;

OPTION -- Installed unit OPTIONS

VERsion -- System software VERSION revision

Pressing F2 from the MAIN MORE menu selects the menu;

F1	F2	F3
ESC	OPTION	VER

OPTION displays installed gigaBERT-660 options, refer to Section 2, for available options.

VER displays software version.

3.6 REAR PANEL AUXILIARY OUTPUTS

Three rear panel mounted BNC connectors provide signals for Error Inject from an external source, and an asynchronous Data Inhibit to gate off the output Data stream. An auxiliary BNC connector provides user-specified custom factory installed options, consult factory for details. Refer to Section 2, Specifications for signal levels.

3.7 REMOTE INTERFACE CAPABILITIES

The gigaBERT-660 Tx supports remote control through the RS-232C and IEEE-488 GPIB remote interface bus connectors on the rear panel. The unit can be operated from the front panel and over the remote interface simultaneously. Any unit status changes made remotely will be displayed on the front panel. All front panel functions except Power and PANEL LOCK can be controlled remotely.

The remote commands sent to the gigaBERT-660 Tx differ from front panel control. The current operating mode is entered directly rather than through sub-menus. Operating frequency, data patterns, and memory contents are programmable directly over the interface.

Refer to Appendix A. for a complete description and programming guide to the RS-232C interface.

Refer to Appendix B. for a complete description and programming guide to the GPIB interface.

SECTION 4

gigaBERT-660 TRANSMITTER

PRINCIPLES OF OPERATION

4.1 DESIGN OVERVIEW

The gigaBERT-660 Tx is designed to generate a programmable WORD of 16 bits, and 5 Pseudo-random binary sequences (PRBS) of 2^n-1 (n= 7, 15, 17, 20, 23), at serial data rates of 1 to 700 MBit/s. The unit incorporates a programmable crystal-locked clock source that operates at this bandwidth, and 2 programmable pulse output amplifiers, for both Clock and Data Output.

Very high frequency GaAs, ECL and discrete circuitry is incorporated on multilayer controlled impedance printed circuit boards. RF shielding and critically timed coaxial cables provide wideband operation with sub-nanosecond timing. An embedded CPU controls the programmable clock source, high speed data generator hardware, programmable WORD loading, remote RS-232C and GPIB interfaces, and soft front panel control.

The gigaBERT-660 Tx consists of six functional block sub-assemblies, as shown in Figure 4.1 gigaBERT-660 Tx Block diagram.

4.2 PLL CLOCK SOURCE PCB

The PLL Clock Source PCB contains the circuitry to generate and distribute the internal clock signals. The clock source consists of a PLL (Phase Locked Loop) controlling a one-octave VCO (Voltage Controlled Oscillator). The microprocessor controls the VCO control voltage, and prescale divider ratios. An optional 1 KHz resolution PLL clock source is available for fine frequency programming.

4.3 DATA GENERATOR PCB

The Data Generator PCB contains the circuitry required to generate the PRBS pattern, programmable WORD, clock distribution, error inject circuitry, and pattern sync generator.

The PRBS data generator utilizes a pattern dependent, n-length shift register (where n is 2^n-1) with modulo-2 feedback, to generate the desired PRBS (pseudo random binary sequence) pattern.

The programmable WORD is level shifted from TTL to ECL and loaded into ECL registers, then multiplexed and clocked out in a serial stream at full-rate. The 16 bits are loaded at full rate, allowing immediate change to the data pattern.

4.3 DATA GENERATOR PCB (cont)

In PRBS mode, the Pattern sync circuit detects the start (n-1 zeros) of the pseudorandom pattern. This produces a single bit width pulse once per pattern frame. In WORD mode the shift register detects the programmable WORD load pulse, which occurs once per WORD frame.

The Error inject circuitry consists of a chain of decade counters, used to generate a pulse every 10E-n, (where n= 3-7) bits. This pulse generates a single bit error on the output data stream, providing a known error rate for back-to-back tests.

4.4 DATA OUTPUT AMPLIFIER PCB

The Data stream generated on the Data Gen PCB is sent to the Data Amp PCB, relatched into a flip flop, and differentially driven into a monolithic GaAs pulse amplifier which drives the front panel DATA and DATA invert outputs. The amplifier provides adjustment, controlled by front panel DACs, of the Data output Amplitude and baseline Offset. The amplifier will drive 2 Volts p-p into a 50 ohm load, unterminated 4 Volts p-p, suitable for Fast TTL and CMOS.

4.5 CLOCK OUTPUT AMPLIFIER PCB

The System Clock signal is sent to the Clock Amp PCB, and differentially driven into a monolithic GaAs pulse amplifier which drives the front panel CLOCK and CLOCK invert outputs. The amplifier provides continuous adjustment, controlled by front panel DACs, of the Clock output Amplitude and baseline Offset. The amplifier will drive 2 Volts p-p into a 50 ohm load, unterminated 4 Volts p-p, suitable for Fast TTL and CMOS.

4.6 CPU PCB

The CPU PCB contains the CPU, RAM, and software PROMs. The 80188 microprocessor handles all inter-board communication, storage and loading of the programmable 16 bit WORD, internal clock source PLL control and scale calculations, front panel interface and remote communication over the RS-232C and GPIB interfaces. Battery backup RAM provides storage of 10 programmed data patterns, 10 programmed clock frequencies, and unit operating status after power loss.

4.7 FRONT PANEL PCB

The front panel PCB provides user control of the unit, and contains the key decoders, LED drivers and 2 X 24 Liquid crystal display. The soft key front panel design and large display allows future enhancements to functionality, and additional custom factory installed options.

APPENDIX A

RS-232C INTERFACE APPLICATION INFORMATION

for
gigaBERT-660 Tx

TABLE OF CONTENTS

A.1 RS-232C REMOTE INTERFACE CAPABILITIES

The BERT-660 supports remote control through the RS-232C interface bus connector on the rear panel. The unit can be operated from the front panel and over the remote interface simultaneously, any unit status changes made remotely will be displayed on the front panel. All of the front panel functions can be controlled over the RS-232C interface, except 'POWER'.

The remote commands sent to the BERT-660 differ from front panel control; the current operating mode is entered directly rather than through sub-menus. Operating frequency, data patterns, and memory contents are programmed directly over the interface.

Commands are provided to read back stored Frequency and Data Memory contents. Memory contents can be read back and printed out for hardcopy archiving. A list of remote commands is given in Appendix C.

A.2 RS-232C INTERFACE DEVICE SETTINGS

The RS-232C interface device settings are programmable through the front panel MENU function (F1 key). The following RS-232C parameters are programmable, along with the possible selections and the default setting in parentheses:

PARAMETER	(DEFAULT)	VALUES
o BAUD RATE	: (9600)	4800, 2400, 1200, 600, 300
o PARITY	: (EVEN)	NONE, ODD
o DATA SIZE	: (8)	7
o EOL	: (CR/LF)	LF/CR, CR, LF
o XON/XOFF	: (ON)	OFF
o ECHO	: (ON)	OFF

To change an RS-232C setting through the front panel,

Step 1: Press F1 to select MENU mode,
Step 2: Select the RS-232C menu choice,
Step 3: Select the desired setting type,
Step 4: Select the desired setting,
Step 5: Press F4 to SET the selection,
Step 6: Press F1 to Exit.

A.3 RS-232C INTERFACE HARDWARE/HANDSHAKING CONSIDERATIONS

The remote interface consists of a 25 pin female D-type connector located on the rear panel. When using the RS-232C remote interface, connect the controller to the BERT-660 rear-panel connector with an appropriate 25 pin cable.

The BERT-660 is configured as an RS-232C Data Circuit terminating Equipment (DCE). For a local (direct) connection to a DTE device (most RS-232C controllers), connect the controller to the BERT-660 with a straight (non-null modem) cable.

To interface to another DCE device, a null modem is required to cross-connect signal pairs (2 & 3, 4 & 5, and 6 & 20). Refer to Table A.1 for RS-232C signal names, pinouts, and functional descriptions.

Table A.1 RS-232C Interface Connector Pin-out

PIN	NAME	FUNCTION
1	GND	Protective Ground.
2	RxD	Received Data Input. Data sent to the BERT-660 is received on this input.
3	TxD	Transmitted Data Output. Data sent from the BERT-660 to the terminal is transmitted on this pin.
4	CTS	Clear to Send. Sent to the BERT-660, a hi-level indicates that external device is ready to accept data from the unit. This pin must be high or open for the BERT to transmit Data. This pin is pulled high internally by 27K ohm to +12V.
5	RTS	Request to Send. Always High.
6	DTR	Data terminal Ready. A high indicates that the unit is ready to transmit data.
7	GND	Signal Ground.
20	DSR	Data Set Ready. Ignored
ALL OTHER PINS		Not used.

A.4 RS-232C INTERFACE TESTING

To test that the RS-232C interface is properly connected, attach a standard 25 pin D-type connector cable between the BERT-660 RS-232C rear panel connector and the controller, with the BERT-660 off. Turn on the BERT-660, the following message should appear on the RS-232C controller's screen, followed by the prompt "BERT660> ":

```
***** MICROWAVE LOGIC BERT-660 TX V2.30 6/90 *****
BERT660>
```

If the message does not appear, check the following;

1) The cable may be defective,
2) The controller may be configured as DCE equipment; a null modem may be required,
3) The controller signal format, or Baud rate may not match the BERT-660's settings; refer to section A.2 for interface setting information.

A.5 PROGRAMMING RS-232C REMOTE COMMANDS

There are two types of remote commands for the BERT-660;

1) set commands (or commands) and,
2) queries commands (or queries).

The set commands force the BERT-660 to take a specific action. The query commands direct the BERT-660 to return status information.

Commands are entered one line at a time. Errors may be corrected while entering a line, with the backspace key. A command string is terminated by a carriage return, which transmits the string to the BERT-660 and executes the command string. All valid commands are executed, incorrect or unsupported commands are responded to by an error message. Refer to section A.6 for RS-232C error messages.

Command lines may contain a single command or multiple commands. The command line may contain both queries and commands. Individual commands within the command line must be separated by semi-colons (;), parameters must be separated by commas (,). Non-decimal numeric parameters, Hexadecimal, Octal, and Binary, must be preceded by a '#H', '#Q', or '#B', respectively. Refer to section A.7 for command line examples.

The entire command name does not have to be entered for the command to be recognized as valid. There is a minimum valid length associated with each command, which is the length that makes it unique from all other commands listed below.

Refer to Appendix C for BERT-660 Tx commands.

A.6 RS-232C ERROR MESSAGES

All RS-232C remote commands received by the BERT-660 are checked for command validity, and appropriate parameters (parameters listed with commands within brackets []). All valid command strings are executed, incorrect command strings are responded to with error messages.

ERROR MESSAGE	ERROR
"Input Lost"	Input data lost over interface.
"Input Buffer Overflow"	Input buffer overflow, command line too long without terminator.
"Command Mnemonic Not Found"	Command not found.
"Invalid Command For Interface"	Command found, but not valid for this interface.
"Invalid Command Type"	Command mnemonic found, but command issued incorrectly: missing, or added, '?' on end of command.
"Too Few Parameters"	Missing parameter.
"Too Many Parameters"	Too many parameters or trailing garbage at end of command.
"Invalid Parameter"	Parameter invalid.
"Parameter Out of Range"	Parameter out of range.
"Parameter Not in Set"	Parameter not one of the values specified for the command.
"Invalid String Length"	Parameter string too long.
"Parameter Separator"	Parameter separator, ';', is missing or command line is terminated following separator.
"Invalid Hexadecimal Parameter"	Parameter not in hexadecimal format, or more than two hexadecimal characters.
"Command Execution Error"	Command not executed properly.

A.7 RS-232C COMMAND EXAMPLES

COMMAND	RESPONSE
clock_source?	Returns the the current clock operating mode, either 'ext',or 'int'.
clock_source ext	Sets unit to clock external mode
clock_source int	Sets unit to clock internal mode
clock_freq?	Returns the current operating frequency.
clock_freq 622.1E+6	Sets current operating frequency to 622.1 MHz.
clock_memory?	Returns all 10 frequencies stored in memory.
clock_memory? 3	Returns the frequency stored in frequency memory 3.
clock_memory 5 , 565300000	Stores frequency 565.3 MHz in frequency memory 5.
recall_freq 1	Sets unit operating frequency to value stored in frequency memory 1.
clock_step?	Returns the current step size.
clock_step 1E+8	Sets unit step size to 100.0 MHz.
clock_stp_up	Increases unit operating frequency by the programmed step size.
clock_stp_dn	Decreases unit operating frequency by the programmed step size.
clock_stp_up 93.4E6	Increases unit operating frequency by 93.4 MHz.
clock_stp_dn 3.1E6	Decreases unit operating frequency by 3.1 MHz.
word_memory?	Returns all 10 data patterns stored in memory.
word_memory? 7	Returns the data pattern stored in WORD memory 7.
word_memory 9 , 16 , #Haa , #H23	Programs the '16' bits of the data pattern stored in WORD memory '9' to the HEX data pattern 'AA23'

A.7 RS-232C COMMAND EXAMPLES (cont)

COMMAND	RESPONSE
word_bits 8 , #Q307	Sets the first '8' bits of the currently generated data to the octal data pattern '307', leaving upper bytes as they were.
recall_word 7	Sets the currently generated data pattern to that stored in WORD Memory 7.
word_mem_len 6 , 16	Sets the length of data pattern stored in WORD memory 6 to 16 bits.
view_angle?	Returns the current Front Panel view angle.
view_angle 3	Sets the front panel LCD view angle to angle '3'.
rs_prompt '660'	Sets the prompt to '660 '.
all_mem?	Returns the clock mode, all 10 frequency memory contents, and all 10 WORD Memory contents. NOTE: the receiver would only return the 10 WORD memories.

APPENDIX B

GPIB INTERFACE APPLICATION INFORMATION

for
gigaBERT-660 Tx

TABLE OF CONTENTS

B.1 GPIB REMOTE INTERFACE CAPABILITIES

The BERT-660 supports remote control through the GPIB interface bus connector on the rear panel. The unit can be operated from the front panel and over the remote interface simultaneously, any unit status changes made remotely will be displayed on the front panel. All of the front panel functions can be controlled over the GPIB interface bus, except 'POWER'.

The remote commands sent to the BERT-660 differ from front panel control; the current operating mode is entered directly rather than through sub-menus. Operating frequency, data patterns, and memory contents are programmed directly over the interface.

Commands are provided to read back stored Frequency and Data Memory contents. The memory contents can be read back and printed out for hardcopy archiving. A list of remote commands is given in Appendix C.

B.2 GPIB INTERFACE DEVICE SETTINGS

For proper GPIB Interface communication and handshaking, the GPIB controller (Computer or other controller), and the Device (BERT-660 Tx), must have their addresses and terminating characters set up prior to use. These 2 requirements are met as noted below.

B.2.1 GPIB INSTRUMENT ADDRESS

Each instrument on the GPIB interface bus must have a unique INSTRUMENT address. The INSTRUMENT address range for the BERT-660 (0 TO 30 decimal) is programmed on the front panel by the 'GPIB ADDR' key. The address cannot be set outside of the range 0 to 30 decimal, except to place the unit OFF-BUS. To select the BERT-660'S GPIB address:

Step 1: Press GPIB ADDR, 'GPIB xx' is displayed on the LCD,
Step 2: Press the left-most up/down keys (Clock Up and Down) to select the desired GPIB address 'xx',
Step 3: Press GPIB ADDR a second time.

The BERT-660 will then respond to commands sent to that INSTRUMENT address. This will be done without affecting the remote command processing.

B.2.2 GPIB MESSAGE TERMINATOR (EOI or EOI/LF)

The GPIB Message Terminator is set to either EOI or EOI/LF, using the Utility Menu (F1) key.

For EOI -- The EOI line will be asserted when the last byte of a message is transmitted.

For EOI/LF -- The last byte of the message will be the line feed character, and the EOI line will be asserted with it's transmission.

To select the GPIB Message Terminator:

Step 1: Press F1 (the MENU key),
Step 2: Select the GPIB menu choice,
Step 3: Select the TERMinator choice,
Step 4: Select between EOI and EOI/LF,
Step 5: Press F4 (the SET key), to enter the selection.

B.3 BERT-660 GPIB INTERFACE FUNCTIONS

The BERT-660 is configured as a talker/listener. No controller functions are implemented. As described in the IEEE-488 GPIB standards the BERT-660 supports the following implementation:

SH1	Complete source handshake.
AH1	Complete acceptor handshake.
T6	Basic talker, serial poll, no talk only, unaddressed if addressed to listen, no extended talker.
L4	Basic listener, no listen only, unaddressed if addressed to talk, no extended listener.
SR1	Complete service request.
RL1	Complete remote/local capability including local lockout.
PP0	No parallel poll capability.
DC1	Complete device clear capability.
DT0	No device trigger capability.
C0	No controller capability.
E2	Tri-state drivers used on DI0 lines for maximum data transfer rate.

B.4 GPIB CONNECTOR PIN-OUTS

The BERT-660 Tx uses the standard D-type 24 pin GPIB connector located on the rear panel. All signals and pins conform to standard GPIB pin out protocol.

B.5 PROGRAMMING GPIB REMOTE COMMANDS

There are two types of remote commands for the BERT-660;
1) set commands (or commands) and,
2) queries commands (or queries).

The set commands force the BERT-660 to take a specific action. The query commands direct the BERT-660 to return status information.

The controller sends commands to the BERT-660 as strings terminated by EOI or EOI/LF character. These command lines can contain a single command or multiple commands. The command line may contain both queries and commands. Each individual command within the command line must be separated by semi-colons (;), parameters must be separated by commas (,). HEXadecimal parameters must be proceeded by a '#H'. Refer to section B.7 for command line examples.

Each query command sent to the BERT-660 will return one response. The response may contain multiple response units (separated by semi-colons), however only one EOI or EOI/LF character is sent by the BERT-660 to the controller for each query command.

The responses for the BERT-660 commands will be either character mnemonics (Example; INT or EXT) or numerics (Example; 700.0). Appendix C lists all of the queries and specifies the query response format the command will return. If the query returns a mnemonic, the valid mnemonics will be listed.

B.5.1 GPIB NUMERIC RESPONSES

If the query responds with a numeric, it will be specified as one of the following types:

1) <NR1 Numeric> - decimal integer
2) <NR2 Numeric> - decimal real number without exponent
3) <NR3 Numeric> - decimal real number with exponent
4) <Non-decimal Numeric> - non-decimal number with leading '#H' (Hex), '#Q' (Octal), or '#B' (Binary) and always in the range 0 to 255 decimal.

Appendix C defines all of the GPIB commands, along with a brief description of each command, the valid parameters, and the basic response format. The parameters shown within brackets '[]' following the command name are required.

Refer to section B.8 for examples of using the GPIB commands with HP BASIC. These command examples shown are in Basic, other controller languages will differ. Refer to the GPIB controller manual for specifics on its implementation.

B.6 GPIB STATUS REPORTING

There is status reporting functionality provided for the GPIB interface, which is based on the service request (SRQ) and is fully defined in the ANSI/IEEE standard 488.2-1987. The implementation used by the BERT-660 for status reporting includes one additional register from what is specified within the IEEE 488.2 standard.

STATUS BYTE

There is a status byte which is used to determine the SRQ status. The individual bits within the status byte represent the different conditions which might cause the request for service, defined as follows:

```
Bits 1 to 3 : Unused
Bit 4       : (TSB)  Test Event Status Bit
Bit 5       : (MAV)  Message Available Bit
Bit 6       : (ESB)  Standard Event Status Bit
Bit 7       : (MSS)  Master Summary Status Bit
Bit 8       : Unused
```

Test Status Event Bit - This is the summary of the Test Status Event Status Byte. It will be set whenever an enabled Test event condition occurs. (Rx ONLY)

Message Available Bit - Set whenever there is output available for the controller.

Standard Event Status Bit - This is the summary of the Standard Event Status Byte. It will be set whenever an enabled standard event condition occurs.

Master Summary Status Bit - This is the Master Summary Status. It is a summary of the status byte, so that whenever one of the bits (TSB, MAV or ESB) is set and it is also enabled (by the Service Request Enable byte), the MSS bit will be set.

SERVICE REQUEST ENABLE

The different conditions for a service request can be individually enabled. The Service Request Enable byte contains the enabling bits for the status byte. For a service request to occur, either the TSB, MAV or ESB bit must be enabled. Each time the BERT-660 is powered on, this byte is reset so that no bits are enabled. The bit definition is the same as the status byte, except bit 7 is undefined.

SRQ

The status byte is used to create a service request. Whenever a condition occurs in the BERT-660 which requires service from the controller, the SRQ (Service Request) line will be set. The SRQ will be reset after the controller finishes a serial poll of the BERT-660 or when all of the service request conditions have stopped.

STANDARD EVENT STATUS REGISTER

The ESB bit is the summary of the Standard Event Status Register. This byte has an enabling byte which works in a similar manner to the above Status Byte. The individual bits within the Standard Event Status Register represent the different conditions which might cause a Standard Event. The bit definitions for the Standard Event Status Register are as follows:

Bit 1 : Operation Complete
Bit 2 : Request Control
Bit 3 : Query Error
Bit 4 : Device Dependent Error
Bit 5 : Execution Error
Bit 6 : Command Error
Bit 7 : User Request
Bit 8 : Power On

Operation Complete - Only set following an *OPC command.

Request Control - Not used.

Query Error - Set under the following conditions:

- o when output has been requested from the BERT 660 and none is available,
- o when a command is sent to the BERT-660 and the BERT-660 still has a message available,
- o when output has been requested from the BERT 660 and an unterminated command has been sent to the BERT-660.

Device Dependent Error - Not used.

Execution Error - Set under the following conditions:
- o when a command parameter is out of range,
- o when the command has too many or too few parameters,
- o when the command cannot be properly executed due to a device condition.

Command Error - Set whenever the BERT-660 receives an unrecognized command, or invalid GPIB command.

User Request - Not Used.

Power On - Set whenever the BERT-660 is powered on.

STANDARD EVENT STATUS ENABLE REGISTER

The different conditions within the Standard Event Status Register can be individually enabled and disabled. The Standard Event Status Enable Register contains enabling bits. Each time one of the event conditions or one of the enabling bits change, the status of the ESB bit is re-evaluated; If any status bit is set and its corresponding enable bit is set, the ESB bit will be set.

Each time the BERT-660 is powered on, this byte is reset so that no bits are enabled. The bit definition for the Standard Event Status Enable Register is the same as for the Standard Event Status Register.

TEST STATUS EVENT REGISTER (Rx ONLY)

The TSB bit is the summary of the Test Status Event Register. This byte has an enabling byte which works in a similar manner to the above Status Byte. The individual bits within the Test Status Event Register represent the different conditions which might cause a Test Event. The bit definitions for the Test Status Event Register are as follows:

Bit 1	: End-Of-Window Condition
Bit 2	: End-Of-Test Condition
Bit 3	: Threshold Error Condition
Bit 4	: Synchronization Loss Condition
Bit 5	: Phase Error Condition
Bits 6 to 8	: Unused

End-Of-Window - Set at the end of each window period.

End-Of-Test - Set at the end of each Test.

Threshold Error - Set whenever Test is running and Errored Second occurs, where error rate is above Test Error Rate Threshold.

Synchronization Loss - Set whenever SYNC LOSS occurs.

Phase Error - Set whenever a Phase Error occurs.

TEST STATUS EVENT ENABLE REGISTER (Rx ONLY)

The different conditions within the Test Status Event Register can be individually enabled and disabled. The Test Status Event Enable Register contains enabling bits. Each time one of the event conditions or one of the enabling bits change, the status of the TSB bit is re-evaluated; If any status bit is set and its corresponding enable bit is set, the TSB bit will be set.

Each time the BERT-660 is powered on, this byte is reset so that no bits are enabled. The bit definition for the Test Status Event Enable Register is the same as for the Test Status Event Register.

GPIB COMMON COMMANDS

The following commands are provided to use with the GPIB status reporting, as defined by IEEE 488.2 for service request:

***STB?, *SRE, *SRE?, *ESR?, *ESE, *ESE?, *CLS**

ADDITIONAL SRQ GPIB COMMANDS (Rx ONLY)

The following commands are provided to use with the Test Status SRQ feature, as explained above:

TSE, TSE?, TSR?

B.7 GPIB COMMAND EXAMPLES

The following command examples use HP BASIC, consult your controller/language manuals for it's specific command syntax. The program lines below are command specific lines extracted from a complete program, line numbers have been omitted for clarity.

A GPIB command consists of three parts, a Basic command ('OUTPUT'), followed by a space, then the address ('723'), another space, and finally the BERT-660 specific command ('clock_freq?'). As shown by the following example:

1) a command to the BERT-660 is entered, in a BASIC program as;

```
OUTPUT 723 clock_freq?
```

2) a command to the BERT-660 to retrieve the response from the BERT-660 is entered, in a BASIC program as;

```
ENTER 723; a$
```

which places the response string into variable a$.

COMMAND LINE(S)	RESPONSE
OUTPUT 723 clock_source? ENTER 723; A$	Returns to the controller the current clock operating mode, either 'ext',or 'int' and places the string (ext or int) into variable a$.
OUTPUT 723 clock_source ext	Sets unit to clock external mode
OUTPUT 723 clock_source int	Sets unit to clock internal mode
OUTPUT 723 clock_freq? ENTER 723; B$	Returns to the controller the current operating frequency and places the value in variable b$.
OUTPUT 723 clock_freq 654.3E+6	Sets current operating frequency to 654.3 MHz.
DIM C$[200] OUTPUT 723 clock_memory? ENTER 723; C$ PRINT C$	Returns to controller all 10 frequency memory contents and outputs them to a printer.
OUTPUT 723 clock_memory? 3 ENTER 723; D$	Returns to controller the frequency stored in frequency memory 3, and places it in d$.

COMMAND LINE(S)	RESPONSE
OUTPUT 723 clock_memory 5 , 565.3E+6	Stores frequency 565.3 MHz in memory 5.
OUTPUT 723 recall_freq 1	Sets unit operating frequency to value stored in frequency memory 1.
OUTPUT 723 clock_step? ENTER 723; F$	Returns to controller current step size and places it in f$.
OUTPUT 723 clock_step 100.0E+6	Sets unit step size to 100.0 MHz.
OUTPUT 723 clock_stp_up	Increases unit operating frequency by the programmed step size.
OUTPUT 723 clock_stp_dn	Decreases unit operating frequency by the programmed step size.
OUTPUT 723 clock_stp_up 93.4E+6	Increases unit operating frequency by 93.4 MHz.
OUTPUT 723 clock_stp_dn 3.1E+6	Decreases unit operating frequency by 3.1 MHz.
DIM I$[200] OUTPUT 723 word_memory? ENTER 723; I$ PRINT I$	Returns to controller all 10 data patterns stored in memory and outputs them to a printer.
DIM J$[80] OUTPUT 723 word_memory? 7 ENTER 723; J$	Returns to controller data pattern stored in memory 7 and places it in variable J$.
OUTPUT 723 word_memory 9 , 16 , #Haa , #H23	Programs the '16' bits of the data pattern stored in memory location '9' to to the hex data pattern 'AA23'. Note that all hex values have leading '#H', octal '#Q', and binary '#B'.
OUTPUT 723 word_bits 8 , #Hba	Sets the first '8' bits of the currently generated data to pattern to 'ba', leaving upper bytes as they were.

COMMAND LINE(S)	RESPONSE
OUTPUT 723 recall_word 7	Sets the currently generated data pattern to that stored in WORD Memory 7.
OUTPUT 723 word_mem_len 6 , 16	Sets the length of data pattern stored in WORD Memory 6 to 16 bits.
OUTPUT 723 view_angle? ENTER 723; M$	Returns to the controller the currently set view angle, and places it in variable M$.
OUTPUT 723 view_angle 3	Sets the front panel LCD view angle to angle '3'.
DIM P$[255] OUTPUT 723 all_mem? ENTER 723;P$ PRINT P$	Returns to the controller the clock mode, all 10 frequency memory contents, and all 10 WORD Memory contents, and outputs them to a printer, and stores them in an array P$. NOTE: the receiver would only have 10 lines returned.
DIM P$[80] OUTPUT 723 clock_term? ; data_term? ; clock_freq ENTER 723;P$ PRINT P$	Returns the corresponding data associated with the command queries.
OUTPUT 723 recall_word 7 ; word_mem_len 6 , 16	Sets the currently generated data pattern to that stored in WORD Memory 7 and sets the length of data pattern stored in data memory 6 to 16 bits.

B.8 IEEE-488.2 PROGRAMMING MANUAL REQUIREMENTS

Certain programming requirements are specified for GPIB interfaces by the American National Standard Institute (ANSI) document, ANSI/IEEE Std 488.2-1987, which are detailed in this section.

B.8.1 POWER ON SETTINGS

The BERT-660 will restore the device settings to their values from when it was last powered off. There are no remote commands which will affect this. The only exception to this is when the non-volatile RAM becomes corrupted, (which should never happen during normal unit operation). RAM Corruption, if it occurs will be displayed on the units LCD display. When this happens, the BERT-660 will have the settings specified in Appendix D.

B.8.2 MESSAGE EXCHANGE

The following message exchange options are as follows:

- The input buffer is command line oriented. There will be a new input buffer for each command line, or program message. The input buffer has a maximum length of 80 characters.

- The only remote commands which will return more than one response message unit (responses separated by semi-colons) are the following:

  sta? , rdm? , rfm? , all? , *lrn?

- All queries (commands) generate their response messages immediately when they are parsed. No queries are held until the responses are read for them to be generated.

- No commands are coupled.

B.8.3 FUNCTIONAL ELEMENTS

A list of the functional elements which are used by the BERT 660 is required by the IEEE 488.2 standard. These are the functional elements used in constructing the remote commands that control the BERT-660. For further details about this, see the IEEE 488.2 standard, sections 4.3, 7.1.1, and 7.3.3.

From Tables 4.2 and 4.3 of the IEEE 488.2 standard, the BERT-660 performs the following:

<PROGRAM MESSAGE>
<PROGRAM MESSAGE TERMINATOR>
<PROGRAM MESSAGE UNIT>
<PROGRAM MESSAGE UNIT SEPARATOR>
<COMMAND MESSAGE UNIT>
<QUERY MESSAGE UNIT>
<COMMAND PROGRAM HEADER> *
<QUERY PROGRAM HEADER> *
<PROGRAM HEADER SEPARATOR>
<PROGRAM DATA SEPARATOR>
<PROGRAM DATA>
<DECIMAL NUMERIC PROGRAM DATA>
<CHARACTER PROGRAM DATA>
<NON-DECIMAL NUMERIC PROGRAM DATA>

* <compound command program header> and <compound query program header> are not handled.

B.8.4 SPECIFIC COMMAND IMPLEMENTATIONS

Reset Command

The reset command, "*rst", performs a device reset. As defined in the IEEE 488.2 standard it will:

- Resets the device settings to default settings, with the exception of stored memory locations and any remote interface settings (see Appendix D).
- Macros are not implemented in the BERT-660, macros are ignored.
- Forces the BERT-660 into Operation Complete Command Idle State (OCIS) and Operation Complete Query Idle State (OQIS).

Self Test Query

The scope of the self test function is limited; it tests the basic BERT-660 functionality.

Overlapped vs Sequential Commands

All commands are sequential commands.

Operation Complete Message

All commands actions are immediate (no overlapped commands), such that operation complete is immediate.

OP-CT1.4 BERT-660 1KHz CLOCK RESOLUTION EDIT SESSION

Example #1: To program a frequency of 352.644 MHz:
"clock_freq 352.644 E+6"

Example #2: To change the frequency step size to 1 KHz and step up the frequency:
"clock_step 1E+3 ; clock_stp_up "

ADDENDUM for V2.31

gigaBERT-660 Manual

A.3 RS-232C INTERFACE HARDWARE/HANDSHAKING CONSIDERATIONS

The functionality of the Data Terminal Ready line has been changed. Replace the Table A.1 of Appendix A with the following table.

RS-232C Interface Connector Pin-out

PIN	NAME	FUNCTION
1	GND	Protective Ground.
2	RxD	Received Data Input. Data sent to the BERT-660 is received on this input.
3	TxD	Transmitted Data Output. Data sent from the BERT-660 to the terminal is transmitted on this pin.
4	CTS	Clear To Send. Sent to the Bert-660, a hi-level indicates that external device is ready to accept data from the unit. This pin must be high or open for the BERT to transmit Data. This pin is pulled high internally by 27K ohm to +12V.
5	RTS	Request To Send. Always High.
6	DTR	Data Terminal Ready. Always High.
7	GND	Signal Ground.
20	DSR	Data Set Ready. Ignored.
ALL OTHER PINS		Not Used.

ADDENDUM for V2.31

gigaBERT-660 Manual

A.3 RS-232C INTERFACE HARDWARE/HANDSHAKING CONSIDERATIONS

The functionality of the Data Terminal Ready line has been changed. Replace the Table A.1 of Appendix A with the following table.

RS-232C Interface Connector Pin-out

PIN	NAME	FUNCTION
1	GND	Protective Ground.
2	RxD	Received Data Input. Data sent to the BERT-660 is received on this input.
3	TxD	Transmitted Data Output. Data sent from the BERT-660 to the terminal is transmitted on this pin.
4	CTS	Clear To Send. Sent to the Bert-660, a hi-level indicates that external device is ready to accept data from the unit. This pin must be high or open for the BERT to transmit Data. This pin is pulled high internally by 27K ohm to +12V.
5	RTS	Request To Send. Always High.
6	DTR	Data Terminal Ready. Always High.
7	GND	Signal Ground.
20	DSR	Data Set Ready. Ignored.
ALL OTHER PINS		Not Used.

gigaBERT-660 Printer Interface

To connect a parallel printer to the gigaBERT-660 you can either purchase a standard IBM compatible Centronics parallel printer cable at your local computer store or make your own according to the following table.

Printer PIN #	DESCRIPTION	gB-660 PIN#
1	-STROBE	1
2	+DATA BIT 0	2
3	+DATA BIR 1	3
4	+DATA BIT 2	4
5	+DATA BIT 3	5
6	+DATA BIT 4	6
7	+DATA BIT 5	7
8	+DATA BIT 6	8
9	+DATA BIT 7	9
10	-ACK (NOT USED)	10
11	+BUSY	11
12	+P.END (NOT USED)	12
13	+SELECT	13
14	-AUTO FEED (NOT USED)	14
31	-INIT PRINTER	16
32	-ERROR	15
36	-SELECT INPUT	17
33	SG	18
19	GND	19
21	GND	20
23	GND	21
25	GND	22
27	GND	23
29	GND	24
30	GND	25

APPENDIX C

gigaBERT-660 Tx REMOTE COMMANDS

C.1 gigaBERT-660 Tx REMOTE COMMANDS

This is the command set for the BERT-660 Tx (Transmitter). All remote commands are listed here. A brief description of each command, valid parameters, and response format are included. The parameters shown within brackets '[]' following the command name are required. The commands are separated into five functional groups, listed in the following order.

	GROUP	FUNCTION
1	Frequency	Frequency Synthesizer
2	Data Pattern	Data Pattern Programming
3	Clock/Data Output	Clock and Data output adjustments of amplitude and offset
4	Miscellaneous	Miscellaneous commands
5	Common Commands	IEEE-488 defined Common Commands

C.1.1 BERT-660 Tx FREQUENCY SYNTHESIZER COMMANDS

COMMAND	ACTION
clock_source [ext,int]	Sets the clock source mode: ext - external clock mode int - internal clock mode
clock_source?	Returns the current clock source. Response: EXT or INT
clock_freq [v]	Sets the current operating frequency to the value 'v', in Hz: 1 to 705 MHz in steps of 0.1 MHz, (1 KHz steps with 1KHz option) Example: To program a frequency of 450.4 MHz: "clock_freq 450.4E+6"
clock_freq?	Returns the currently programmed operating frequency. Response: <NR3 Numeric>
clock_memory [m],[v]	Sets the frequency at memory location 'm' to the value 'v', in Hz, as follows: 'm' : 0 to 9 'v' : 1 to 705 MHz in 0.1 MHz steps, (1 KHz steps with 1KHz option) Example: To set memory location 4 to 700.0 MHz: "clock_memory 4 , 700E+6".
clock_memory? [m]	Returns the frequency value stored at memory location 'm': 0 to 9 The response will contain the memory location and the frequency. Response: <NR1 Numeric> , <NR3 Numeric>
clock_memory?	Returns the contents of all 10 frequency memory locations. The response will be 10 message units separated by semi-colons ';', and will contain the memory location and frequency. Response: <NR1 Numeric> , <NR3 Numeric>
recall_freq [m]	Sets the operating frequency to that stored at memory location 'm': 0 to 9

C.1.1 BERT-660 Tx FREQUENCY SYNTHESIZER COMMANDS (cont)

COMMAND	ACTION
save_freq [m]	Stores the current operating frequency into memory location 'm': 0 to 9
clock_step [v]	Sets the frequency Step size to the value 'v', in Hz: 0.1E+6, 1.0E+6, 10.0E+6, or 100.0E+6 (0.01E+6, 0.001E+6 with 1KHz option)
clock_step?	Returns the current frequency step size. Response: <NR3 Numeric>
clock_stp_up [v]	Step frequency Up, increment current frequency by value 'v', in Hz: 0.1 to 704.0 MHz, in 0.1 MHz steps (1 KHz steps with 1KHz option)
clock_stp_up	Increment frequency by the current frequency step size.
clock_stp_dn [v]	Step frequency Down, decrements the current frequency by value 'v', in Hz: 0.1 to 704.0 MHz, in 0.1 MHz steps (1 KHz steps with 1KHz option)
clock_stp_dn	Decrement frequency by the current frequency step size.

C.1.2 BERT-660 Tx DATA PATTERN PROGRAMMING COMMANDS

COMMAND	ACTION
prbs_length [v]	Sets the current data pattern to PRBS pattern PN[v], where 'v'; 7,15,17,20,23
prbs_length?	Returns the current PRBS pattern. Response: <NR1 Numeric>
word_bits [l],[b1,b2]	Sets the current data pattern to the 16 bit programmable Data pattern with the data being specified as the data length 'l' and the data as the bytes 'b1' and 'b2', with the following: 'l' : 8 or 16 'b1','b2': #H00 to #HFF, (Hex) or #Q000 to #Q377, (Octal) or #B0 to #B11111111 (Binary) NOTE: 'b2' allowed only with l = 16 Example: To program an 8 bit word of E4H: "word_bits 8 , #HE4"
word_bits?	Returns the programmable WORD pattern. The response will contain the pattern length, and data bytes. Response: <NR1 Numeric> , <Non-decimal Numeric(s)>
word_length [l]	Sets the current pattern to the programmable WORD pattern with a length of 'l': 8 or 16
word_memory [m],[l],[b1,b2]	Sets the programmable WORD Memory at memory location 'm', with WORD length 'l' and data bytes 'b1' and 'b2', with the following: 'm' : 0 to 9 'l' : 8 or 16 'b1','b2': #H00 to #HFF, (Hex) or #Q000 to #Q377, (Octal) or #B0 to #B11111111 (Binary) NOTE: 'b2' allowed only with l = 16 Example: To program memory location 3 with a 16 bit word of C4F0H: "word_memory 3,16,#HC4,#HF0"

C.1.2 BERT-660 Tx DATA PATTERN PROGRAMMING COMMANDS (cont)

COMMAND	ACTION
word_memory? [m]	Returns the contents of the 16 bit programmable WORD Memory at memory location 'm': 0 to 9 The response will contain the memory location, length, and data bytes. Response: <NR1 Numeric> , <NR1 Numeric> , <Non-decimal Numeric(s)>
word_memory?	Returns the contents of all 10 programmable WORD Memories. The response will be 10 message units separated by semi-colons ';', and will contain the memory location, length, and data bytes of each memory. Response: <NR1 Numeric> , <NR1 Numeric> , <Non-decimal Numeric(s)>
word_mem_len [m],[l]	Sets the pattern length of the programmable WORD Memory at memory location 'm' to length 'l', with the following: 'm' : 0 to 9 'l' : 8 or 16
recall_word [m]	Sets the current data pattern to the pattern stored in the WORD Memory location 'm': 0 to 9
save_word [m]	Stores the current WORD pattern into WORD Memory location 'm': 0 to 9
data_pattern [prbs,word]	Sets the current pattern, prbs - switches to PRBS mode, word - switches to WORD mode.
data_pattern?	Returns the current operating pattern mode; Response: PRBS or WORD

C.1.2 BERT-660 Tx DATA PATTERN PROGRAMMING COMMANDS (cont)

COMMAND	ACTION
data_invert [on,off]	Sets the Output Data polarity: on - Outputs Inverted Data, off - Outputs True Data.
data_invert?	Returns current status of Output Data Polarity. Response: ON or OFF
word_order [msb,lsb]	Sets the WORD Transmit Order to MSB or LSB. The unit will start the transmission with the MSB of the first byte or the LSB of the first byte (MSB/LSB). After the MSB (or LSB), the bits would be transmitted continuing with the bits of the first byte until the LSB (or MSB) is reached, then proceeding with the next byte in the same order. msb - Transmit MSBit first. lsb - Transmit LSBit first.
word_order?	Returns WORD transmission order; Response: MSB or LSB

NOTE: The selected ORDER is effective only on the pattern bytes edited after the setting of the ORDER. If the ORDER is changed in the middle of programming pattern bytes, the bytes programmed before the ORDER change will be reversed.

For example, if the word ORDER is previous LSB at word location 6 and the data pattern is programmed as follows:

```
edit_begin  6
byte_length 4,0
byte_block  0,16,#HAA,#HAA
word_order  MSB
byte_block  2,16,#HAA,#HAA
edit_end    6
```

The pattern at location 6 would be:

```
byte_block  0,32,#H55,#H55,#HAA,HAA
```

C.1.3 BERT-660 Tx CLOCK/DATA OUTPUT COMMANDS

COMMAND	ACTION
clock_ampl [v]	Sets the Output Clock Amplitude to the specified voltage 'v': 0.70 to 2.00 in 0.05 V steps
clock_ampl?	Returns the Output Clock Amplitude. Response: <NR2 Numeric>
clock_amp_up [v]	Clock Amplitude Up, increments the clock amplitude by 'v': 0.05 to 1.30 in 0.05 V steps
clock_amp_up	Increment clock amplitude by 0.05 V.
clock_amp_dn [v]	Clock Amplitude Down, decrements the clock amplitude by 'v': 0.05 to 1.30 in 0.05 V steps
clock_amp_dn	Decrement clock amplitude by 0.05 V.
clock_offset [v]	Sets the Clock Baseline Offset to the specified voltage 'v': -2.00 to 1.00 in 0.05 V steps
clock_offset?	Returns the Clock Baseline Offset. Response: <NR2 Numeric>
clock_off_up [v]	Clock Offset Up, increments the clock offset by 'v': 0.05 to 3.00 in 0.05 V steps.
clock_off_up	Increment clock offset by 0.05 V.
clock_off_dn [v]	Clock Offset Down, decrements the clock offset by 'v': 0.05 to 3.00 in 0.05 V steps.
clock_off_dn	Decrement clock offset by 0.05 V.
data_ampl [v]	Sets the Output Data Amplitude to the specified voltage 'v': 0.70 to 2.00 in 0.05 V steps.
data_ampl?	Returns the Output Data Amplitude. Response: <NR2 Numeric>
data_amp_up [v]	Data Amplitude Up, increments the data amplitude by 'v': 0.70 to 2.00 in 0.05 V steps.
data_amp_up	Increment data amplitude by 0.05 V.

C.1.3 BERT-660 Tx CLOCK/DATA OUTPUT COMMANDS (cont)

COMMAND	ACTION
data_amp_dn [v]	Data Amplitude Down, decrements the data amplitude by 'v': 0.05 to 1.30 in 0.05 V steps.
data_amp_dn	Decrement data amplitude by 0.05 V.
data_offset [v]	Sets the Data Baseline Offset to the specified voltage 'v': -2.00 to 1.00 in 0.05 V steps.
data_offset?	Returns the Data Baseline Offset. Response: <NR2 Numeric>
data_off_up [v]	Data Offset Up, increments the data offset by 'v': 0.05 to 3.00 in 0.05 V steps.
data_off_up	Increment data offset by 0.05 V.
data_off_dn [v]	Data Offset Down, decrements the data offset by 'v': 0.05 to 3.00 in 0.05 V steps.
data_off_dn	Decrement Data offset by 0.05 V.
amplitude [v]	Sets both Data and Clock amplitudes to 'v': 0.70 to 2.00 in 0.05 V steps.
amplitude?	Returns both Data and Clock amplitudes, returns 2 values. Response: <NR2 Numerics>
offset [v]	Sets both Data and Clock baseline offsets to 'v': -2.00 to 1.00 in 0.05 V steps.
offset?	Returns both Data and Clock baseline offsets, returns 2 values. Response: <NR2 Numerics>
error_rate [off,ext,rate_3,rate_4,rate_5,rate_6,rate_7]	Sets the Error injection rate into the output data pattern, off - turns error inject off ext - enables External error injection mode rate_v - where v = 3, 4, 5, 6 or 7, and the error rates are 1E-'v'

C.1.3 BERT-660 Tx CLOCK/DATA OUTPUT COMMANDS (cont)

COMMAND	ACTION
error_rate?	Returns the Error Injection Rate Response: OFF, EXT, RATE_3, RATE_4, RATE_5, RATE_6 or RATE_7
error_single	Injects a single error into the output data pattern.

C.1.4 BERT-660 Tx MISCELLANEOUS COMMANDS

COMMAND	ACTION
gpib_address [v]	Sets the GPIB Address to 'v': 0 to 30 (allowed by RS-232 only)
gpib_address?	Returns the current GPIB Address. (allowed by RS-232 only) Response: <NR1 Numeric>
gpib_bus [talk_listen,off_bus]	Sets the BERT-660 GPIB Bus control. When set OFF-BUS, it will not communicate over the GPIB bus. talk_listen - Communication allowed off_bus - Off GPIB Bus (allowed by RS-232 only)
gpib_bus?	Returns the GPIB Bus status. (allowed by RS-232 only) Response: TALK_LISTEN or OFF_BUS
rs_prompt [s]	Sets the RS-232 prompt to the string 's'. Maximum length is 8 characters.
rs_xon_xoff [on,off]	Sets the RS-232 XON/XOFF protocol status. on - Enables XON/XOFF protocol. off - Disables XON/XOFF protocol.
rs_xon_xoff?	Returns the RS-232 XON/XOFF protocol status. Response: ON or OFF
rs_echo [on,off]	Sets the RS-232 ECHO mode status. on - Enables ECHO mode. off - Disables ECHO mode.
rs_echo?	Returns the RS-232 ECHO mode. Response: ON or OFF
rs_pmt_lf [on,off]	Sets the RS-232 Prompt Line-Feed (0AH) Control status. on - Line-Feed (0AH) sent after RS-232 prompt. off - Line-Feed (0AH) not sent after RS-232 prompt.
rs_pmt_lf?	Returns the RS-232 Prompt Line-Feed (0AH) Control Status. Response: ON or OFF

C.1.4 BERT-660 Tx MISCELLANEOUS COMMANDS (cont)

COMMAND	ACTION
header [on,off]	Sets the remote command response header status. on - Responses include command header (command name). off - Disables header in response.
header?	Returns the remote command response header status. Response: ON or OFF
view_angle [v]	Sets the front panel LCD View Angle to 'v': 0, 1, 2, or 3.
view_angle?	Returns the current LCD view angle. Response: <NR1 Numeric>
all_mem?	Returns 10 Frequency and 10 WORD Memory location contents, and input clock status. The response is 21 message units, and are separated by ';'. The message unit responses are specified under the commands 'clock_source?', 'clock_memory?', and 'data_memory?'.
options?	Returns the options included with BERT-660; Response: 32K, PROM and/or 1KHz, or NONE.
logo?	Returns the BERT-660 Logo. (allowed by RS-232 only)

C.1.5 BERT-660 Tx IEEE-488 COMMON COMMANDS

COMMAND	ACTION
*cls	Clear Status, clears the Standard Event Status Register.
*ese [v]	Sets the Standard Event Status Enable Register to 'v': 0 to 255
*ese?	Returns the current contents of the Standard Event Status Enable Register. Response: <NR1 Numeric>
*esr?	Returns the current contents of the Standard Event Status Register. Response: <NR1 Numeric>
*idn?	Returns the BERT-660 identification as follows: Response: Manufacturer,Model, 0,Firmware Level
*lrn?	Returns the current device setup, as a series of message units that can be later sent back to the BERT660 to restore the unit to the same setup. Response: sequence of response messages
*opc	Operation Complete Command, sets the operation complete message in the Standard Event Status Register immediately, because all device operations are completed immediately.
*opc?	Returns the value '1' immediately. Response: <NR1 Numeric>
*rst	Performs a device reset, resetting the BERT-660 to the factory supplied setup, see Appendix C.
*sre [v]	Sets the Service Request Enable Register to 'v': 0 to 255
*sre?	Returns the current contents of the Service Request Enable Register, where bit 6 is ignored. Response: <NR1 Numeric>

C.1.5 BERT-660 Tx IEEE-488 COMMON COMMANDS (cont)

COMMAND	ACTION
*stb?	Returns the current contents of the Status Byte, where bit 6 is the Master Summary Status bit. Response: <NR1 Numeric>
*tst?	Returns the result of the self-test. Successful completion of the test will return a value of '0'. The scope of the test is limited. Response: <NR1 Numeric>
*wai	Wait To Continue, forces the BERT-660 to stop processing remote commands, until all operations are completed, which is immediately.

NOTE: See section B.8, "GPIB Status Reporting", for information concerning the usage of the Standard Event Status Enable Register, the Standard Event Status Register, the Service Request Enable Register, and the Status Byte.

APPENDIX D

D.1 gigaBERT-660 Tx FACTORY DEFAULT SETTINGS

Clock

External Clock Source	-	INT
Frequency	-	705.0 MHz
Frequency Memory (All 10)	-	705.0 MHz
Frequency Step Size	-	100.0 MHz

Data Pattern

PRBS Pattern	-	PN7
WORD Memory (All 10)	-	AA 55
Programmable Data Pattern	-	AA 55
Current Pattern Setting	-	PRBS
Data Invert	-	OFF
Word Order	-	LSB

Clock / Data Outputs

Output Clock Amplitude	-	1.50 V
Clock Baseline Offset	-	-0.75 V
Output Data Amplitude	-	1.50 V
Data Baseline Offset	-	-0.75 V
Error Injection Rate	-	OFF

Miscellaneous

View Angle	-	0
Panel Lock	-	OFF
Response Header	-	ON

D.1 gigaBERT-660 Tx FACTORY DEFAULT SETTINGS (cont)

Remote Interfaces

Remote Mode	-	LOCAL
GPIB Address	-	15
GPIB Bus	-	TALK_LISTEN
GPIB Terminator	-	EOI/LF
RS-232C Prompt	-	BERT660>
RS-232C Baud Rate	-	9600
RS-232C Parity	-	EVEN
RS-232C Data Size	-	8 bits
RS-232C EOL	-	CR/LF
RS-232C Echo	-	ON
RS-232C Prompt Line-Feed	-	OFF

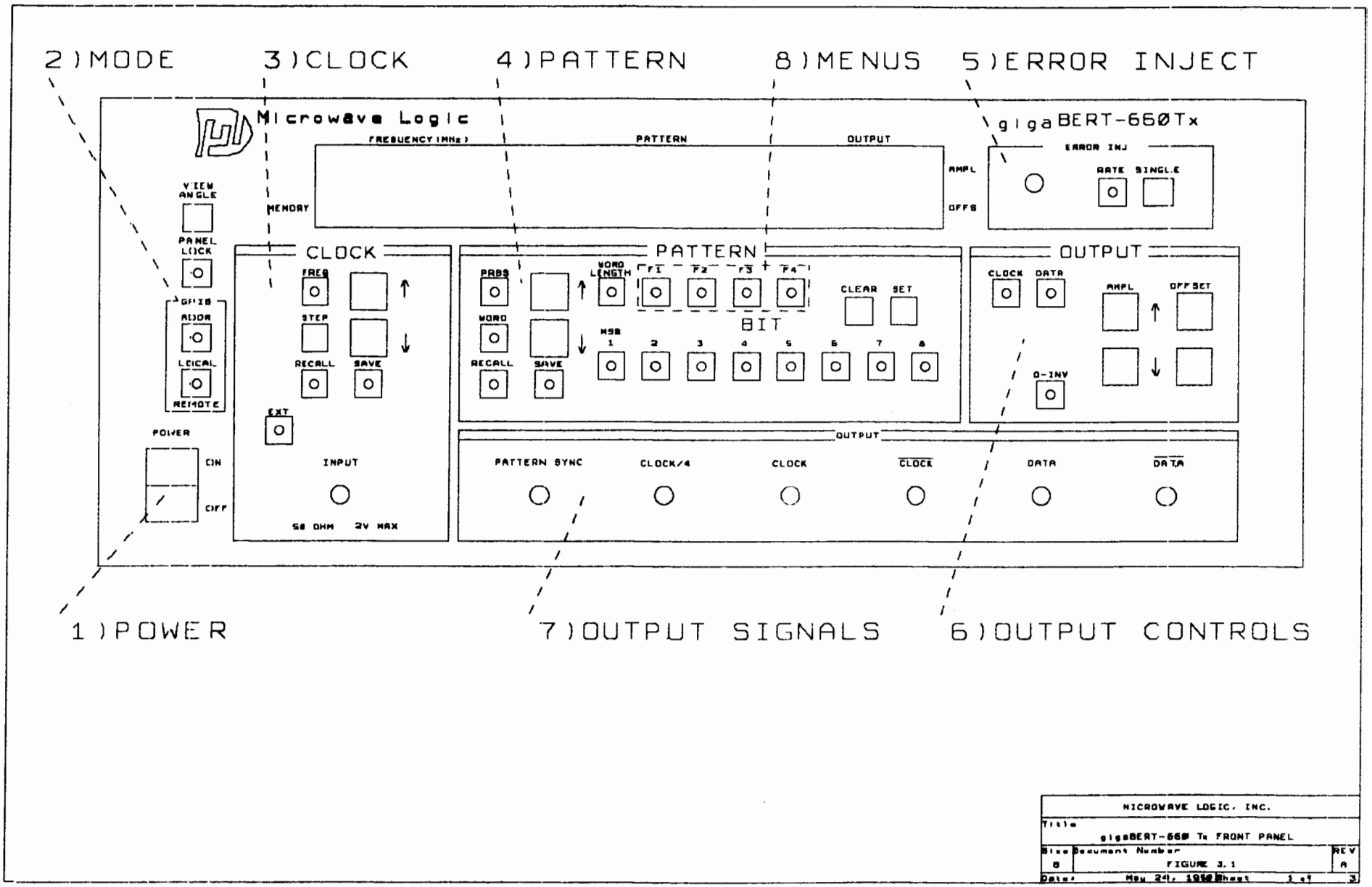
APPENDIX E

gigaBERT-660 FIGURES

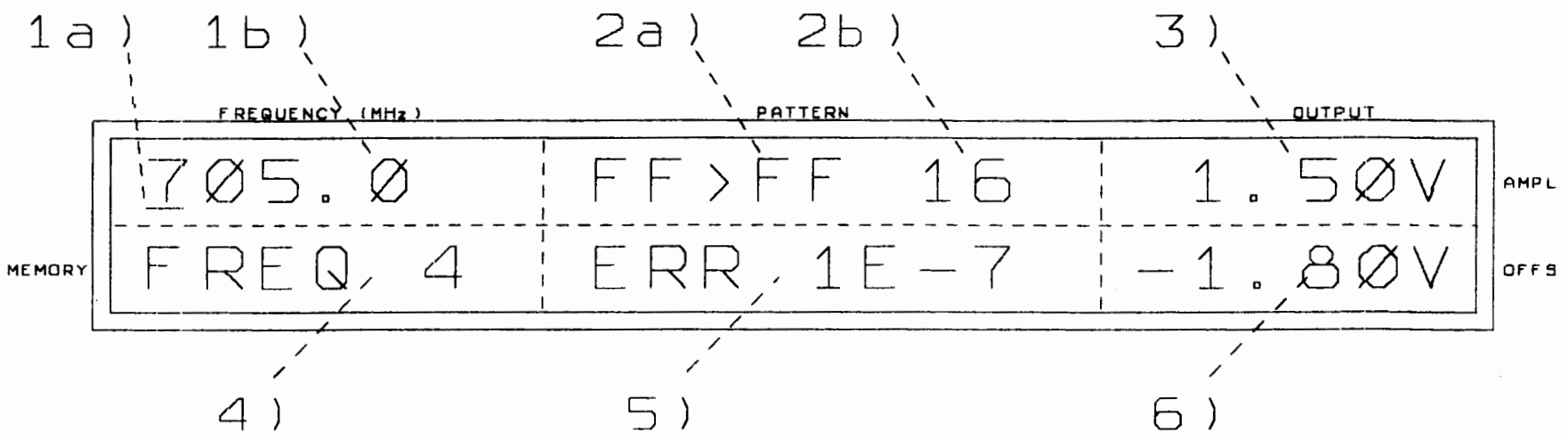
E.1 STATEMENT OF CONFIDENTIALITY

The following pages contain Figures referenced elsewhere in this manual. They are included for reference purposes only, and are included on a confidential basis only. They are Microwave Logic, Inc. proprietary materials and are not to be copied, distributed, or divulged to others without express written authorization of Microwave Logic, Inc.

E.2 LIST OF FIGURES



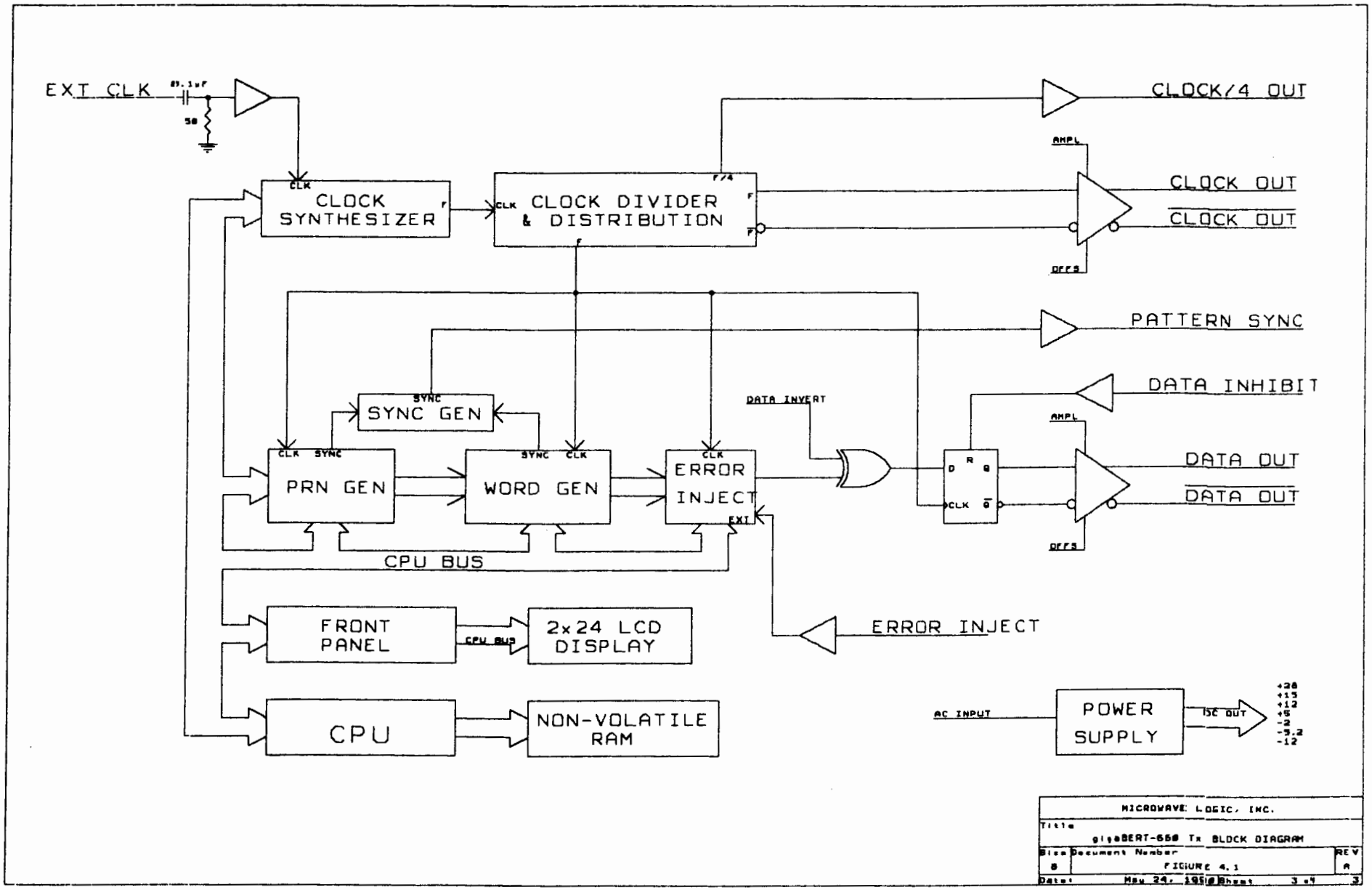
MICROWAVE LOGIC, INC.		
Title: gigaBERT-660 Tx FRONT PANEL		
Size: B	Document Number: FIGURE 3.1	REV: A
Date: May 24, 1990	Sheet 1 of 1	3



LCD FIELD DISCRIPTION

1) INTERNAL CLOCK FREQUENCY /EXT CLK INDICATOR
2a) WORD BYTE(S)/PRBS PATTERN
2b) WORD LENGTH/INVERT
3) AMPLITUDE
4) FREQ-WORD MEMORY/GPIB ADDR
5) ERROR INJECT RATE
6) BASELINE OFFSET

MICROWAVE LOGIC, INC.		
Title: gigaBERT-660 Tx LCD DISPLAY FIELDS		
Size: A	Document Number: FIGURE 3.2	REV: F
Date: May 24, 1990	Sheet 2 of 3	


EXT CLK
0.1uF
50
CLK
CLOCK SYNTHESIZER
F
CLK
CLOCK DIVIDER & DISTRIBUTION
F/4
F
F
F
CLOCK/4 OUT
AMPL
CLOCK OUT
CLOCK OUT
OFFS
PATTERN SYNC
DATA INHIBIT
SYNC
SYNC GEN
DATA INVERT
AMPL
CLK SYNC
PRN GEN
SYNC CLK
WORD GEN
CLK
ERROR INJECT
EXI
D R Q
CLK Q
DATA OUT
DATA OUT
OFFS
CPU BUS
FRONT PANEL
CPU BUS
2x24 LCD DISPLAY
ERROR INJECT
CPU
NON-VOLATILE RAM
AC INPUT
POWER SUPPLY
ISC OUT
+28
+15
+12
+5
-2
-5.2
-12
MICROWAVE LOGIC, INC.
Title
gigaBERT-650 Tx BLOCK DIAGRAM
Size
B
Document Number
FIGURE 4.1
REV
A
Date: May 24, 1995
Sheet 3 of 4
3

OPTION OP-MT1

32K PROGRAMMABLE WORD OPTION

OP-MT1.1 INTRODUCTION

This section describes the 32K WORD option, and provides operating information for programming the WORD, through front panel control and over the remote interfaces.

The following section replaces the WORD, and WORD LENGTH control information in section 3.4.4 Tx Pattern, all other Pattern controls function as described in section 3.4.4 remain unchanged.

OP-MT1.2 WORD CONTROL

WORD

Pressing WORD selects programmable WORD mode. The programmable WORD can be modified, recalled, or stored in memory. The programmable pattern's byte address can be selected with the PATTERN up/down keys, and is displayed on the upper row of the LCD;

'AAAA HH'

1) AAAA is the byte address (0 to 4095),
2) HH is the byte HEX value.

An illuminated WORD LED indicates programmable WORD pattern is currently generated and output.

The programmable WORD is modified through the front panel in 8 bit bytes. The number of available bytes is dependent on WORD length. WORD length is selected with the WORD LENGTH key, and displayed on the LCD.

The desired byte to modify (with the 'bit' keys) is selected with the PATTERN up/down keys, and displayed on the LCD in HEX, while the binary value of the byte is displayed by the row of 8 'BIT' keys. Each bit's value is displayed by the key's LED; an illuminated key signifies a '1', an unlit key a '0'. Pressing the 'BIT' key toggles it's logic value.

For example to modify bit 13;

1) select WORD mode (press WORD key)
2) select WORD address 1 (pattern up/down keys)
3) toggle BIT 5 key (13th bit).

OP-MT1.3 WORD LENGTH CONTROL

WORD LENGTH This key selects WORD length select mode, and displays the length on the upper row of the LCD. The WORD length increment size varies as;

WORD LENGTH	INCREMENT
1-256 bytes	1 bit
256-4096 bytes	1 byte

The WORD length is changed with the PATTERN up/down keys. The new length WORD pattern is immediately output, starting with byte 0 through the respective highest byte, dependent on WORD length. The original pattern is maintained in RAM, and is recalled (until power down) if the original length is returned.

OP-MT1.4 WORD UTILITY MENU

The WORD utility menu allows programming of WORD length, EDITing of WORD contents, FILLing of WORD contents over a specified range, and programming of WORD sync threshold in the Rx. See section 3.5 for utility menu details.

Function key F4 from the main menu accesses the WORD Utility menu. The WORD sub-menus are structured as follows for the Tx;

F1	F2	F3	F4
MORE	EDIT	LENGTH	FILL

F1	F2
ESC	ORDER

The WORD menu provides control of the 32K WORD as follows:

EDIT -- Change the pattern at an address.

LENGTH -- Change the WORD Pattern Length.

FILL -- Fill the WORD with an 8 bit pattern.

ORDER -- Transmit LSB/MSB of each byte first

EDIT allows the editing of the WORD at the address specified using the BIT/CLEAR/SET keys.

LENGTH allows the change of the WORD LENGTH in terms of Bytes and Bits. The WORD length can be set as follows:

0 BYTES + 1 BITS
512 BYTES + 0 BITS in 1 bit increments

512 BYTES + 0 BITS
4096 BYTES + 0 BITS in 1 byte increments

FILL allows the entire WORD Pattern to be filled with the specified 8-bit pattern.

ORDER allows selection between transmitting the WORD PATTERN starting with the MSB of the first byte or the LSB of the first byte (MSB/LSB). After the MSB (or LSB), the bits would be transmitted continuing with the bits of the first byte until the LSB (or MSB) is reached, then proceeding with the next byte in the same order.

NOTE: The selected ORDER is effective only on the pattern bytes edited after the setting of the ORDER. If the ORDER is changed in the middle of programming pattern bytes, the bytes programmed before the ORDER change will be reversed.

For example, if the word ORDER is previous LSB at word location 6 and the data pattern is programmed as follows:

```
edit_begin  6
byte_length 4,0
byte_block  0,16,#HAA,#HAA
word_order  MSB
byte_block  2,16,#HAA,#HAA
edit_end    6
```

The pattern at location 6 would be:

```
byte_block  0,32,#H55,#H55,#HAA,HAA
```

OP-MT1.5 BERT-660 32K PROGRAMMABLE WORD REMOTE COMMANDS

These are the additions to the command set of the BERT-660 for the 32K Programmable Word option. All of the remote commands used in controlling the BERT-660 Tx are listed here.

The following commands replace the standard BERT-660 remote commands found in APPENDIX C of this manual. The commands which are replaced or modified are:

MODIFIED:

all_mem? - will not return Data Memories

REPLACED:

word_bits [l],[b1,b2]
word_bits?
word_memory [m],[l],[b1,b2]
word_memory?
word_length [l]
word_mem_len [m],[l]

COMMAND	ACTION
edit_begin [n]	Starts the remote edit process of the 32K Programmable Word. The 32K word pattern to edit/download/upload is copied to a scratch pad location. The pattern location is specified by 'n'. 'n' : -1 to 9 (-1, current pattern) (0 to 9, memory locations) NOTE: This command MUST be done prior to changing any 32K pattern data.
edit_end [n]	Ends the remote edit process of the 32K Programmable Word. The 32K word pattern which has been edited/downloaded can be copied to memory or to the current pattern, or can be discarded. The pattern is specified by 'n', as follows: 'n' : -2 to 9 (-2, discards pattern) (-1, copies to current pattern) (0 to 9, copies to memory 'n') NOTE: This command MUST be done after edit session completed.

COMMAND	ACTION
edit_cntrl?	Returns which interface is currently editing any 32K word pattern, if any. If the front panel Menu is in a 32K edit screen, it will return LOCAL. If a remote interface is in the process of editing a 32K word pattern, it will return REMOTE. Otherwise, it will return NONE. Response: LOCAL, REMOTE, or NONE
byte_length [by],[bi]	Sets the current 32K Programmable Word to the length 'by' bytes plus 'bi' bits, with the following: 'by' : 0 to 4096 'bi' : 0 to 7 NOTE: If 'by' is greater than 255, 'bi' must be set to 0. Example: To set the 32K length to 19 bytes + 6 bits (158 bits): "byte_length 19 , 6 "
byte_length?	Returns the current 32K Programmable Word length. The response will contain the number of bytes and the number of bits, as follows: Response: <NR1 Numeric> , <NR1 Numeric>
byte_fill [l],[b1,...,bn]	Fills the entire 32K Programmable Word buffer with the pattern 'b1',... 'bn', which is of length 'l' bits, as follows: 'l' : 8 to 80, in 8 bit increments 'b1'...'bn': #H00 to #HFF, (Hex) or #Q000 to #Q377, (Octal) or #B0 to #B11111111 (Binary) NOTE: 'bn' only with l > 8 * (n-1) Example: To fill the 32K Word with a repeating 'A4C2F0' pattern: "byte_fill 24, #HA4,#HC2,#HF0"

COMMAND	ACTION
byte_block [a],[l],[b1,b2,...,bn]	In terms of blocks, edits the up to 10 bytes of the 32K Programmable Word at the address 'a', for a length 'l', replacing what is currently there with the pattern 'b1','b2',..., as follows: 'a' : 0 to 4095 'l' : 1 to 80 (10 bytes Maximum) 'b1'...'bn': #H00 to #HFF, (Hex) or #Q000 to #Q377, (Octal) or #B0 to #B11111111 (Binary) NOTE: 'bn' only with l > 8 * (n-1) Example: To edit addresses 340, 341 and 342 of the 32K Word and replace with the pattern 'C4FD56': "byte_block 340 , 24 ,#HC4,#HFD,#H56"
byte_block? [a]	Returns the contents of 32K Word for a length of 10 bytes starting at addr 'a': 'a' : 0 to 4095 The response will contain the address, the length, and the 10 byte pattern, as follows: Response: <NR1 Numeric> , <NR1 Numeric> , <Non-decimal Numeric(s)> NOTE: If 'a' is within 10 bytes of the end of the pattern, the number of bytes will be less than 10. Also, if the last byte of the pattern is included and it is not 8 bits in length, the unused bits will be '0'.
byte_edit [a],[b1]	Edits the 32K Programmable Word at the address 'a', replacing what is currently there with the pattern 'b1', as follows: 'a' : 0 to 4095 'b1' : #H00 to #HFF, (Hex) or #Q000 to #Q377, (Octal) or #B0 to #B11111111 (Binary) Example: To edit addresses 340 of the 32K Word and replace with 'C4': "byte_edit 340 , #HC4"

COMMAND	ACTION
byte_edit? [a]	Returns the contents of 32K Word at addr 'a': 'a' : 0 to 4095 The response will contain the address and the byte, as follows: Response: <NR1 Numeric> , <Hexadecimal Numeric(s)> NOTE: If the returned byte is the last byte of the pattern and it is not 8 bits in length, the unused bits will be '0'.
byte_insert [a],[l],[b1,b2,...]	Inserts into the 32K Programmable Word at the address 'a', for a length 'l', with the pattern 'b1','b2',..., where the address must be less than the word length, as follows: 'a' : 0 to 4095 'l' : 8 to 80, in 8 bit increments 'b1'...'bn': #H00 to #HFF, (Hex) or #Q000 to #Q377, (Octal) or #B0 to #B11111111 (Binary) NOTE: 'bn' only with l > 8 * (n-1) Example: To insert the pattern 'B44B' into the 32K Word at address 400: "byte_insert 400 , 16 ,#HB4,#H4B"
byte_delete [a],[l]	Deletes 'l' bits from the 32K Programmable Word starting at the address 'a', where the address must be less than the word length, as follows: 'a' : 0 to 4095 'l' : 8 to 80, in 8 bit increments Example: To delete 24 bits from the 32K Word starting at address 400: "byte_delete 400 , 24 "

COMMAND	ACTION
word_order [msb,lsb]	Sets the WORD Transmit Order to MSB or LSB. The unit will start the transmission with the MSB of the first byte or the LSB of the first byte (MSB/LSB). msb - Transmit MSBit first. lsb - Transmit LSBit first.
word_order?	Returns WORD transmission order; Response: MSB or LSB

NOTE: The selected ORDER is effective only on the pattern bytes edited after the setting of the ORDER. If the ORDER is changed in the middle of programming pattern bytes, the bytes programmed before the ORDER change will be reversed.

For example, if the word ORDER is previous LSB at word location 6 and the data pattern is programmed as follows:

```
edit_begin  6
byte_length 4,0
byte_block  0,16,#HAA,#HAA
word_order  MSB
byte_block  2,16,#HAA,#HAA
edit_end    6
```

The pattern at location 6 would be:

```
byte_block  0,32,#H55,#H55,#HAA,HAA
```

OP-MT1.6 BERT-660 32K SAMPLE REMOTE EDIT SESSION

Example #1: To download a 4096 byte pattern to memory location 6 without affecting the current 32K pattern.

```
edit_begin     6
word_order     msb
byte_length    4096,0
byte_block     0,80,#H00,#H01,#H02,#H03,#H04,#H05,#H06,
                                                #H07,#H08,#H09
byte_block    10,80,#H00,#H01,#H02,#H03,#H04,#H05,#H06,
                                                #H07,#H08,#H09
   .
   .
   .
byte_block 4080,80,#H00,#H01,#H02,#H03,#H04,#H05,#H06,
                                                #H07,#H08,#H09
byte_block 4090,48,#H00,#H01,#H02,#H03,#H04,#H05
edit_end       6
```

Example #2: To edit memory location 6 and save into memory location 8 without affecting the current 32K pattern.

```
edit_begin     6
word_order     msb
byte_block     12,16,#H24,#H23
edit_end       8
```

NOTE: If the data pattern being programmed is not of the same WORD ORDER as the WORD ORDER associated with the pattern location, the word order must be specified prior to editing (downloading) any data.

OPTION OP-MT2

32K WORD PROM OPTION

OP-MT2.1 INTRODUCTION

This section describes the 32K WORD PROM option, and provides operating information for controlling the 32K PROM, with both front panel control and remote control.

The following section replaces the PATTERN RECALL control information in sections 3.4.4 Pattern of this manual, all other Pattern controls function as described in section 3.4.4 remain unchanged.

OP-MT2.2 PATTERN RECALL CONTROL

RECALL — Pressing RECALL selects WORD memory mode. 10 programmed WORDs can be stored and recalled. There are an additional 10 programmed WORDs permanently stored in the optional PROM. The selected WORD memory location is displayed on the lower row of the LCD display. The memory location (0 to 9, or PROM WORDs P0 to P9) is selected with the PATTERN up/down keys. The data pattern stored in memory is output immediately upon selection.

An illuminated RECALL LED indicates the WORD memory location can be selected by the PATTERN up/down keys.

OP-MT2.3 BERT-660 32K WORD PROM REMOTE COMMANDS

These is one addition to the command set of the BERT-660 for the 32K WORD PROM option. There are no modifications. The addition is to allow the Recall of the PROM WORD pattern.

COMMAND	ACTION
recall_prom [m]	Sets the current data pattern to the pattern stored in the 32K WORD PROM Memory location 'm': 0 to 9

OPTION OP-CT1

1KHz RESOLUTION CLOCK SOURCE OPTION

OP-CT1.1 INTRODUCTION

This section describes the 1KHz RESOLUTION CLOCK SOURCE option, and provides operating information for controlling the clock source through front panel control and over the remote interfaces.

OP-CT1.2 FRONT PANEL CONTROL

The current operating frequency is displayed on the upper left row of the LCD, in units of 1 MHz. The 1 KHz option provides user control of operating frequency to a resolution of the 1 KHz digit. The programmed frequency has an accuracy of 10 ppm.

The frequency step programming size is selectable among 100 MHz, 10 MHz, 1 MHz, 100 KHz, 10 KHz, and 1 KHz.

See Section 3.4.3 for more information on the front panel control of the clock.

OP-CT1.3 BERT-660 1KHz CLOCK RESOLUTION REMOTE COMMANDS

There are no additions to the command set of the BERT-660 for the 32K Programmable Word option. The frequency, frequency memory and step programming are modified to allow greater precision.

COMMAND	ACTION
clock_freq [v]	Sets the current operating frequency to the value 'v', in Hz: 1 to 705 MHz in steps of 0.001 MHz, Example: To program a frequency of 450.432 MHz: "clock_freq 450.432E+6"
clock_freq?	Returns the currently programmed operating frequency. Response: <NR3 Numeric>
clock_memory [m],[v]	Sets the frequency at memory location 'm' to the value 'v', in Hz, as follows: 'm' : 0 to 9 'v' : 1 to 705 MHz in 0.001 MHz steps,
clock_memory? [m]	Returns the frequency value stored at memory location 'm': 0 to 9 The response will contain the memory location and the frequency. Response: <NR1 Numeric> , <NR3 Numeric>
clock_memory?	Returns the contents of all 10 frequency memory locations. The response will be 10 message units separated by semi-colons ';', and will contain the memory location and frequency. Response: <NR1 Numeric> , <NR3 Numeric>
clock_step [v]	Sets the frequency Step size to the value 'v', in Hz: 100.0E+6, 10.0E+6, 1.0E+6, 0.1E+6 0.01E+6, or 0.001E+6
clock_step?	Returns the current frequency step size. Response: <NR3 Numeric>