THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 61, No. 6, July–August 1982
Printed in U.S.A.

Automated Repair Service Bureau:

Mechanized Loop Testing Design

By O. B. DALE, T. W. ROBINSON, and E. J. THERIOT

(Manuscript received December 2, 1980)

The Mechanized Loop Testing (MLT) system is a functional unit of the Automated Repair Service Bureau (ARSB) which tests and analyzes the condition of customer loops. The test results are used to verify trouble conditions, assist telephone company personnel in providing repair-commitment information to the customer, dispatch the appropriate repair craft, and reduce manual testing requirements. The MLT system design is distributed over three processing levels of the ARSB tree structure (host, front end, controller) to provide the necessary record utilization, data processing, loop analysis, and control of test equipment. The automatic access, monitoring, and testing of loops is performed by specially designed test equipment under the direct control of the controller. The MLT system provides a set of test series, each designed for specific applications. The test series are composed in real time as a function of the equipment thought to be present on the loop and the results of tests already performed to that point in the test sequence. This adaptive testing process has been implemented in hardware and software to provide an effective loop-testing system for most applications in the Bell System.

I. INTRODUCTION

The Mechanized Loop Testing (MLT) system is a major functional unit of the Automated Repair Service Bureau (ARSB) which automatically tests and analyzes the condition of customer loops. The tests are run at the time the customer reports a trouble, or at any time it is necessary to check the condition of the loop. Results of the tests and a detailed loop analysis are generally available within 30 seconds from the time the request is initiated. The results are used to provide the customer with an accurate assessment of the trouble and to assist in establishing an appropriate repair commitment time. In addition, the results can be used to efficiently dispatch repair craft and reduce the manual testing requirements of the repair operation.[1,2]

The MLT system had its origins in a testing system developed in the early 1970s, known as the Line Status Verifier (LSV).[3] This system was a threshold-based testing system used by repair-center personnel to test a customer line in a rapid, automatic fashion with a simple console input request. Tests initiated at the time of customer contact reduced substantially subsequent testing by skilled repair personnel. The LSV was later integrated with the Loop Maintenance Operations System (LMOS) so that a test could be initiated from an LMOS computer terminal and the result made part of the LMOS trouble report record. While this generation of the ARSB provided significant economies, it was clear that LMOS, by virtue of its comprehensive data base and computing power, provided the potential for much more sophisticated loop testing. For example, the electrical characteristics of the customer's station, loop, and central office equipment can be derived from the LMOS data and used to direct tests in an adaptive fashion and provide a comprehensive analysis in real time.[4] It was also clear that the LSV was not an appropriate testing vehicle since its hard-wired logic and threshold testing capabilities were not well-matched to the adaptive testing techniques and sophisticated analyses envisioned. Hence, a new testing system was necessary to take full advantage of the LMOS potential and provide additional benefits.

The development of these expanded testing capabilities occurred in two logical post-LSV phases. The first phase, representing the second generation of automated testing and known as the Automatic Line Verification system, developed the data-communication structure, modular physical design, loop access and automatic monitoring capabilities, computer-driven self-maintenance features, and the basic LMOS record-utilization techniques. The third-generation testing system, known as MLT, added a more comprehensive testing package and associated control software which used records more extensively. In this paper, we shall discuss only the resulting MLT system.

The MLT system was intended to significantly reduce but not eliminate the need for manual testing facilities such as local test desks.[1] Since manual testing facilities were already present in repair bureaus, they could be used as backup testing devices when the MLT system experienced temporary outages. In addition, some testing needs, such as interactive testing between field and inside repair craft, coin station testing, and four- and eight-party line testing seemed to be best left to the manual testing facilities because they were not needed frequently and, therefore, mechanizing them did not appear to be cost effective.

The purpose of this article is to describe the MLT system design. Most of the article is devoted to the hardware/software design, but it should be noted that an equally important piece of the design, namely, the personnel subsystem, is covered, at least sparingly, elsewhere in this issue.[5,6] The application of human factors engineering for the MLT project by a talented group of psychologists was a significant and highly valued contribution to the design and introduction of the system. Similarly, we felt it somehow unfair to pay only brief attention to other areas of the project including the elaborate self-maintenance design of the MLT system, but, again, brevity won out.

II. BACKGROUND

A number of requirements and assumptions influenced the design of the MLT system and led to an architecture that distributes the MLT software functions over three processors. Two of the processors have as their primary function the implementation of the LMOS system with which MLT must interact; the third processor is dedicated to the MLT hardware control task.

2.1 Operational users

The MLT system provides operational data to two types of users: Repair Service Attendants (RSA), who are in contact with the customer, and Repair Service Bureau (RSB) personnel, who analyze the trouble and dispatch repair craft. The needs of these two users are similar, but not identical.

The RSA needs a test summary that provides insight to the reported problem in a global way. Is a trouble confirmed? Is it a central office trouble, a loop trouble, a station trouble? The test has to be performed automatically when the trouble report is taken and the results are needed promptly so that an appropriate repair commitment can be given to the customer.

The RSB needs complete test results, preferably included with the trouble ticket that is automatically produced when the RSA takes the trouble report. The RSB also needs the ability to perform tests upon demand, sometimes while the repair craft is at the location of the trouble. Thus, the RSB needs a list of the available tests, some designed to duplicate the comprehensive tests performed when a trouble is taken, some tailored to providing data on only a subset of all possible problems but at smaller costs of system resources and with shorter run times. A total of 11 different series of tests were determined to be required. Examples of these limited test series include: ROTARYDIAL to test the speed and make-break ratio of a rotary dial, RINGER to simply count the number of ringers, LINECKT to test only the office line circuit, and LOOP to test only the loop and on-hook station.

For both users, it is necessary to interpret the test results in light of expected office, line, and station equipment (as gleaned from LMOS line records) and to be tolerant of incorrect or absent equipment records.[4] These requirements dictate a close tie to LMOS in the initiating of tests automatically, in getting test results onto the trouble ticket, and in the use of equipment data to define expected test results.

2.2 Support users

The MLT system has to provide for two additional types of users concerned with support functions. The MLT Administrator is concerned with MLT maintenance and the MLT Data Manager is concerned with MLT software data initialization and integrity.

The MLT Administrator requires tools to perform maintenance functions on the MLT system and to change various system parameters and thresholds so that the performance can be tailored to a particular test environment. Among the maintenance functions required is the ability to calibrate both MLT equipment and test trunks so that systematic errors can be subtracted from test results. Other functions provide the ability to perform tests designed to verify the general "sanity" of the system as a preventative maintenance tool or to perform detailed diagnostic testing of circuitry when specific hardware faults are known or suspected. Control functions include the ability to change the test-result decision thresholds of acceptable levels of foreign voltage and loop unbalance and the ability to take specific equipment out of service for maintenance. Eleven different transactions or commands were identified for maintenance and control functions.

The MLT Data Manager requires tools to create and maintain the various data bases related to MLT. In particular, a data base is used to define the equipment configuration of each set of MLT test hardware. This data base includes not only the specific quantities of optional testing hardware and test trunks that are present, but it also provides information concerning which test trunks are used to test which lines, the type of switching machine involved (step-by-step, crossbar, ESS), and the calibration constants for test trunks and MLT testing hardware. Some of these data are considered static, that is, seldom changed and then only by an appropriate user. The configuration and status data are of this type. Other data are considered dynamic, that is, changed by software; calibration data are of this type.

The requirements for the two support users are not served by a single software structure. The needs of the Administrator must be met by real-time software that can interact with the testing process when necessary.

Since the MLT files overlap the LMOS files, and since similar file creation and maintenance problems are solved by LMOS via off-line processes, the same approach was used for MLT. We will not discuss these off-line processes further except to note that one of the considerations was to design procedures for the creation and maintenance of the MLT files to be similar to the procedures used for LMOS files.

2.3 Hardware imposed requirements

Since the test hardware contained no processing capability, the software had to control the hardware on a step-by-step basis. The software had to identify the proper hardware resources for a particular use, allocate the resources, and cause the equipment to perform a particular sequence of steps on a time-critical schedule. This suggested that the hardware controller tasks should be on a dedicated machine to meet the real-time constraints.

III. ARCHITECTURE AND FUNCTIONAL DISTRIBUTION

The MLT system can be thought of as having five fundamental tasks: accessing the loop, monitoring the loop to ensure that testing can proceed, testing the loop, analyzing the test results, and presenting the results in an easily understood fashion. Loops are accessed via "no-test" trunks which connect the test system to the switching machine. Test trunks are switched onto the desired loop by commands sent from the test system to the switching machine. The test system then automatically monitors the loop for hazardous and/or busy conditions to determine if the testing process can proceed. The access and monitoring processes are performed by the Loop Testing Frame (LTF) under the direction of the MLT Controller (see Fig. 1). Assuming that testing can proceed, the Controller directs the LTF to connect an MLT Measurement Module (MMM) to the test trunk (and hence the loop) and then commands the MMM to perform a series of tests on the loop based on the central office, loop, and station equipment indicated by the LMOS data base. Tests proceed in an adaptive fashion, taking advantage of the increased knowledge of the loop as each test is run.[4] The Controller analyzes the test results and decides when to terminate the testing process and communicate the results to the front-end processor, where they are presented to the user. Typical results might look like those shown in Fig. 2.

The provision of these five basic tasks is accomplished by functions distributed throughout the ARSB system.

3.1 ARSB host processor

The host processor is used by the ARSB to maintain historical data on closed trouble reports and to maintain extensive line record infor-

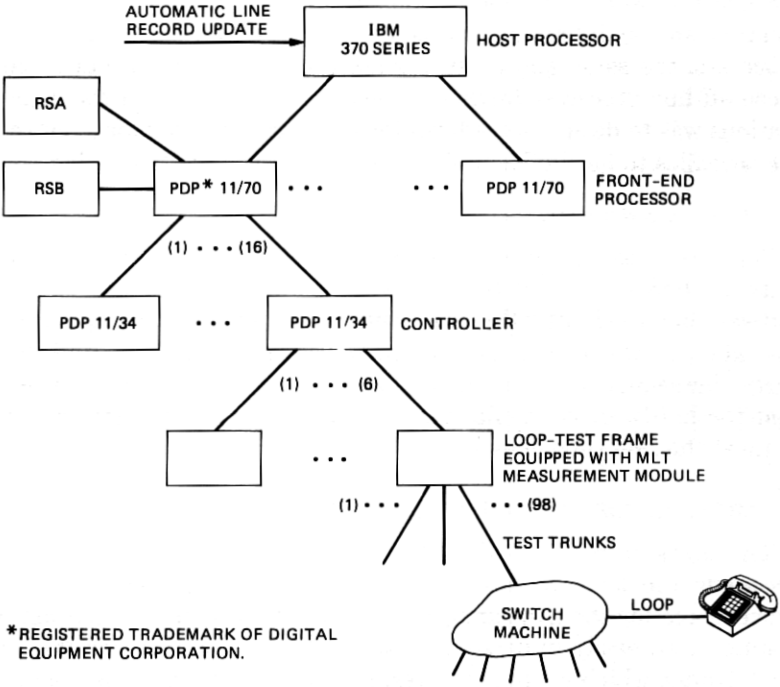

Fig. 1—Automated Repair Service Bureau architecture.

mation on each line. A subset of this line record (called a mini-line record) is duplicated in a front-end (FE) processor for fast access by RSB craft and to provide the ability to support basic operations when the link to the host fails. For MLT, the host is used to scan line records, at the time they are established, for the presence of installed equipment that could influence testability or test results. These data are formatted into a six-byte field for the mini-line record. Also, the host has to expand coded test results in real time when they are received from the FE into an english narrative in accordance with a predefined algorithm.

3.2 ARBS front-end processor

The FE is used by the ARSB to process trouble reports and to interface with RSA and RSB craft in real time.[8] The LMOS software on the FE maintains the mini-line record information. It also supports the numerous CRT terminals for trouble entry and printers to provide trouble tickets—called Basic Output Reports (BOR)—to the RSB craft.

Certain files maintained by the LMOS software had to be expanded to provide MLT information. For example, the file which provided data on the NPA-NNX (area code-exchange code) combinations adminis-

```
TEST TN 212 3867155                      EC 123 PRTR W020 REQ BY CFE

  TYPE FULL     MDF DED NBR                     OVERRIDE LINE RECORD? N

  CO:                         OSP:                         TERM: 608 PBX

  MLT RESULTS:   VER 21   GROUND

    GROUND, FAULT T-G
    3 TERMINAL DC RESISTANCE
     >     3500.00 KOHMS T-R
     =       33.62 KOHMS T-G
     >     3500.00 KOHMS R-G
    3 TERMINAL DC VOLTAGE
     =        0.00 VOLTS  T-G
     =        0.00 VOLTS  R-G
    AC SIGNATURE NOT EXPECTED
    3 TERMINAL AC RESISTANCE
     =       64.47 KOHMS T-R
     =       32.42 KOHMS T-G
     =      624.88 KOHMS R-G
    LONGITUDINAL BALANCE MARGINAL
     =       46     DB
    VALID THERMISTOR CONFIGURATION
      T-R THERMISTOR FOUND
    CROSSBAR, NO LINE CKT TEST

PROCESSING DATE AND TIME 10-09-78 0355A
*** END OF DATA ***
```

MLT Summary - 1 line narrative that provides a summary of the test results

3 Terminal DC Resistance Measurement - The first statement identifies the type, severity and location of the trouble (GROUND FAULT, T-G). The remaining information provides modifying analog values.

3 Terminal DC Voltage Values - This section provides values that define the true amount of voltage present on the line. In this case, no voltage was present.

3 Terminal AC Signature Information - The first statement indicates that MLT did not expect an AC signature for the type of termination (608 PBX). The modifying values are descriptive of the termination that was found.

Longitudinal Balance - This gives an indication of the lines susceptability to noise

Thermistor Information - This section provides information about thermistors detected on the line.

Central Office Information - This section provides information about the Central office line circuit and the system's ability to draw and break dial tone. In this case, MLT tests run on the crossbar switch cannot be interrogated from a test trunk.

Fig. 2—Typical MLT test results.

tered by the FE had to be enhanced to include an indication of MLT testability for the NPA-NNX and which MLT equipment could be used. The LMOS software also had to be expanded to include a provision to schedule the MLT process each time a trouble was taken on a line that was testable by MLT. The scheduling procedure included the passing of the identification of the terminal on which the trouble was taken, pertinent information from the mini-line record (such as the six bytes of test-affecting data), and information from the NPA-NNX file.

The MLT software on the FE provides several functions:

(*i*) It provides an interface to the LMOS software so that testing can be scheduled when a trouble report is being taken. It also translates the MLT bytes into attributes that describe the electrical characteristics of the loop, termination, and central office equipment for use by the Controller.

(*ii*) It provides an interface to RSB craft so that subsequent or alternate testing of a line can be performed when requested.

(*iii*) It provides an interface to the MLT administrator so that MLT maintenance requests can be serviced.

(*iv*) It manages the testing process by controlling the throughput to the various MLT Controllers and by preventing lost or delayed requests. In doing so, it resolves conflicts between the automatic tests requested as a result of trouble entry and tests requested by the RSB or MLT Administrator.

(*v*) It returns data to the correct requester in a format appropriate to the request.

(*vi*) It maintains data files unique to MLT needs. For example, one file contains the particular LTF configurations served by the MLT Controller as well as related calibration data.

(*vii*) It stores the Controller software and associated tables, and loads the Controllers when the system is initialized or when certain trouble conditions arise at the Controllers.

3.3 MLT Controller

The MLT Controller is given as large a part of the MLT software task as possible. This is done partly to minimize the load on the FE and partly to ensure that the LMOS software is isolated from MLT software changes. The Controller software is responsible for allocating MLT hardware resources, controlling the hardware to perform a specific test, interpreting test results in light of expected results, adjusting the sequence of tests in accordance with expected test results, and determining the format of the report.

Because it was originally expected that the Controller would be placed in non-EDP environments, the Controller hardware was kept simple. No colocated off-line storage devices or peripherals are used and all software is kept in 256 Kbytes of main memory. The Controller software and associated tables are stored in the FE and down-loaded over a data link to the Controller at the time of system initialization. Although subsequent deployment strategies have tended to centralize Controllers in controlled environments, the simplicity of the Controller hardware configuration has provided very high reliability and allowed low-cost backup schemes.

The operating system used in the Controller is the same Bell Operating System (BOS) used in the FE, although only a subset of the BOS features are provided in the Controller.[7] In particular, BOS provides a software driver for interfacing to multiple loop testing frames. The driver links the testing hardware and application software in real time so that data can be sent to and from an LTF in a serial format.

Similarly, communication between the FE and Controller is managed by the Communication Control Manager (CCM)[8] software on the FE and a subset of CCM, known as MLTCCM, on the Controller. The communication protocol used was chosen to be a subset of the IBM bisync protocol used between the host and FE processors to avoid the creation of a new protocol.

3.4 Loop Testing Frame

The Loop Testing Frame (LTF), equipped with MLT measurement modules, carries out the access, monitoring, and test functions under the command of the Controller (Figs. 3 and 4). A data link between the Controller and the Communication Control Circuit (CCC) of the LTF provides the data communication facility. No-test trunks between the Trunk Access Switch (TAS) of the LTF and the switching machines provide the metallic test path to the loop. Commands from the Controller are decoded by the CCC, and control information is passed to the appropriate LTF subsystem. Similarly, data from each subsystem are passed to the CCC which appends the output address and transmits the data to the Controller for interpretation.

The LTF was envisioned as an area-based test unit, i.e., a single system deployed on a wide-area basis and serving many central office switching machines, to share the common testing facilities over as many lines as possible. Since interactive tests were to be relegated to existing manual testing facilities, the most lengthy test series run by MLT on a loop was estimated to take less than 15 seconds of actual testing time. Hence, an area-based, high-usage, low-holding-time system was conceived. The LTF was designed to serve up to a maximum of 98 no-test trunks with nine test ports (or nine simultaneous access/monitor/test operations) for a concentration ratio of about 11:1. Traffic estimates suggested that this was adequate to cover anticipated testing

Fig. 3—Loop Testing Frame equipped with four MLT measurement modules.

traffic for more than 250,000 lines, which was expected to at least equal the capacity of the Controller.

The LTF serving area was determined not only by the demographics but also by the physical limitations of metallic testing. Beyond about 3,000 ohms of loop resistance or 100,000 feet if 19-gauge cable is used, it becomes increasingly difficult to differentiate between open loops and loops terminated with station set ringers or the equivalent. Since differentiation between open and terminated loops is essential, 3,000 ohms essentially defined the LTF serving area. This was a reasonable

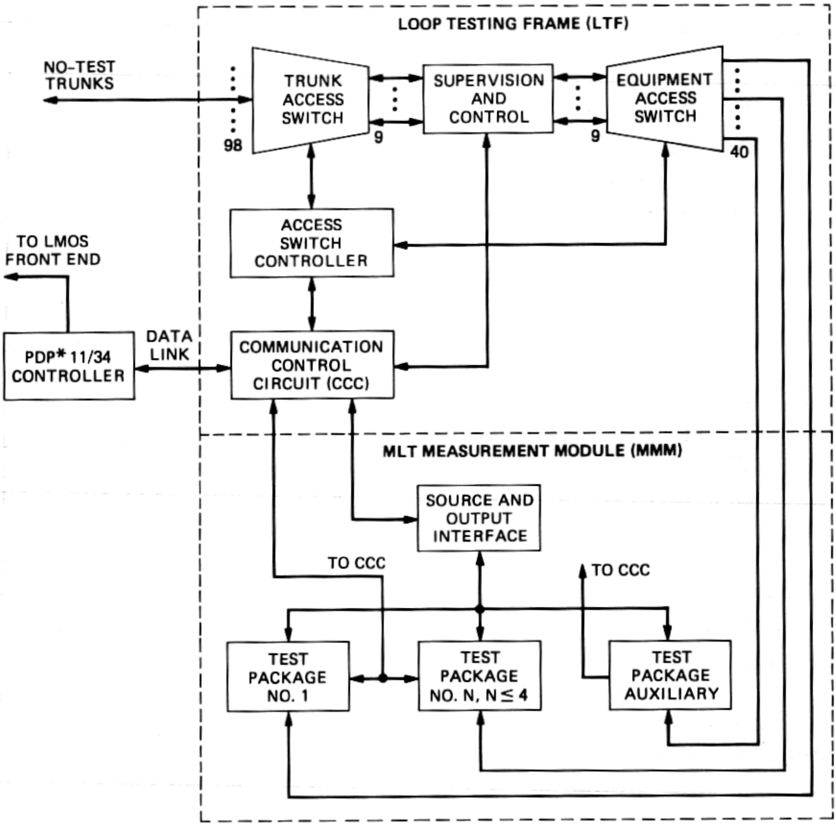

Fig. 4—Schematic of loop testing frame equipped with an MLT measurement module.

limitation since the signaling range for most no-test trunks is in the neighborhood of 1,500 ohms, and most loops are also less than 1,500 ohms. Hence, the LTF serving area accommodated most loop lengths with maximum-length no-test trunks, thereby supporting rather neatly the area-based testing concept.

3.5 Configuration

Each FE can serve up to 16 Controllers. Each Controller can serve up to six LTFs with no more than 40 NNXs or 120 no-test trunks. The Controller limits are primarily based on table limitations, but a ratio of about three no-test trunks per NNX has proven to be the proper average for MLT testing traffic. The LTFs are connected by data links to the Controller and therefore can be located in central offices in a pattern that ensures optimum coverage relative to the 3,000-ohm LTF serving area.

IV. SOFTWARE DESIGN

In this section the design of the software that executes on the Controller is described in some detail. First, we define three terms:

(*i*) LTF Communication—The protocol used to communicate between the Controller and the LTF was referred to briefly in an earlier section. In particular, a given transmission "packet" can be a variable number of words in length. Each word consists of two parts, an address and data. In the direction from the Controller to the LTF, the protocol includes a bit to indicate if the word is the last one of a "packet." One or more packets of data interchange may be necessary to set up or perform the most basic hardware action. In the direction to the Controller, the data is completely asynchronous. The Controller must be capable of accepting data at the peak rate determined by the serial line speed and must buffer these data until they can be processed.

(*ii*) Test Sequence—A set of hardware/software interactions that accomplish a single characterization, such as a count of ringers, a dc or ac Thevenin equivalent circuit, etc. Twenty-three separate test sequences are provided by MLT.[4]

(*iii*) Test Series—A set of test sequences that provide a complete characterization of a line. Obviously, the requested type of series influences the specific sequences that are used; for example, the RINGER series (designed to be a high-speed, low-cost test to count ringers) does not include sequences that make measurements on the switching office line circuits. But, the particular sequences that are used—even the order of the sequences—can be influenced both by the results of previous test sequences and the expected results from examination of the records.[4]

The operating system and MLT software processes are organized as depicted in Fig. 5. Their functions are described below.

The *Driver*, which is part of the operating system, interfaces with the Frame Communication Manager (FCM) and to the hardware circuitry that implements the serial interface to the LTFs. The driver is provided data for transmission that has been formatted into the proper protocol. It collects data from the various hardware circuits on a character-interrupt basis and inserts the data into buffers that are emptied by the FCM process when cpu time permits. The driver serves the data distribution and collection function to the LTFs and permits the MLT software to be largely unaware of the existence of more than one LTF.

The *Frame Communication Manager* interfaces to the driver and to the TSP and SUP (supervisor) processes that contain application

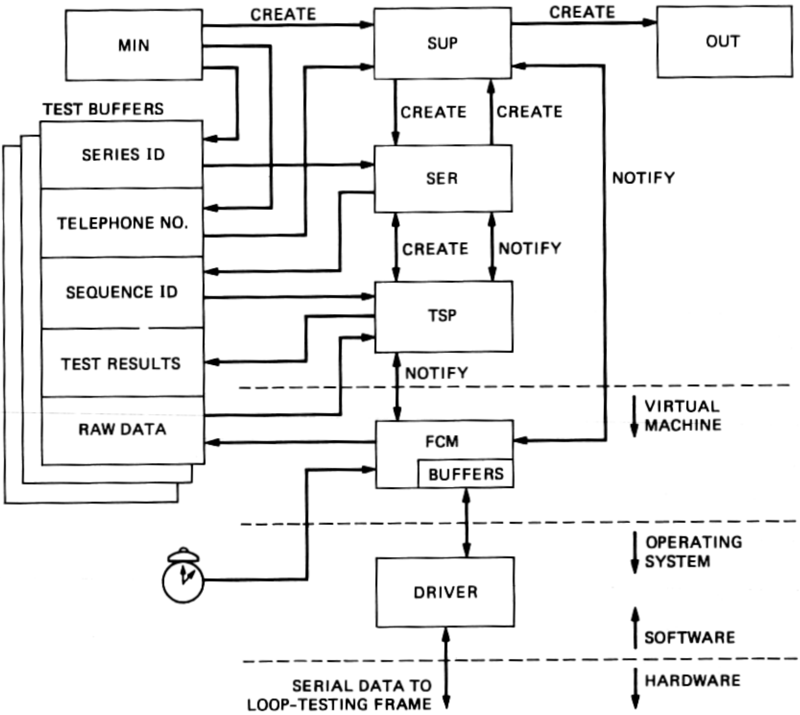

Fig. 5—MLT controller software structure.

software. The FCM process performs a service that is similar to the CCM process discussed earlier. The FCM can be thought of as an extension of the operating system in that it provides a virtual machine environment for the application software. With FCM, the processes that control a single test series have no knowledge of the existence of other processes executing other tests concurrently. The FCM accepts primitive commands to be sent to the LTFs, implements the protocol, collects data and sends it to the proper user process, and generally gives each user process the appearance of being the only process requesting data from the LTF.

The FCM process is created at initialization time, and it never terminates. It is awakened by either of two types of events. User processes, knowing that they have data to send to the LTF, can awaken the FCM process through BOS via a "notify" mechanism. FCM can also be awakened by an alarm mechanism, also managed by BOS. Each time that FCM is awakened, it checks whether data are available in the buffers that are filled by the driver. If so, it collects the data into complete messages and provides them to the user processes by writing the data into a previously agreed upon area of the test buffer. Normally, the user process is waiting for either these data or for a timeout time. The FCM, seeing that the data are available (or that the time-out occurred), awakens the user process that owns the test buffer (again via BOS). Also, each time that the FCM is awakened, it searches through the test buffers to see if any process has data ready to be sent to an LTF. Data to be sent are left in a previously agreed upon area in the test buffer and a ready flag is set in the buffer. Finally, FCM cancels any previous alarm and sets a new alarm call to be awakened within a short time.

The selection of the sleep time was an interesting problem. At one extreme, FCM could set the alarm for such a short time that little processing resources are available to other processes. At the other extreme, FCM could set the alarm for such a long time that the data collected by the driver would be utilized too slowly and would overflow the existing buffers. An aspect of this design is that the alarm feature is used less frequently when the Controller is heavily loaded with tests since the notify mechanism awakens FCM more frequently. The alarm feature is required to collect data when only one test is underway in the Controller. Under this condition, it is desirable not to wait too long before looking for returned data as this would unnecessarily delay the testing process. The optimum sleep time was determined experimentally to be in the range of 100 milliseconds.

The *Test Supervisor* (TSP) process is a child process to the Series Control (SER) and implements the test sequences. That is, SER, when it first determines that testing is required, creates a child TSP process (via a call to BOS) and passes to TSP the address of the test buffer that is owned by SER. At this point, the child TSP process inherits the test buffer. When a requested test sequence is completed, TSP notifies its parent process and asks to have its execution suspended. Each time that SER wants another test sequence executed, the name of the sequence is placed in an agreed-upon place in the test buffer and TSP is notified. Each time that TSP performs a sequence, it puts the accumulated data in the test buffer in two areas. Binary results (yes-no) are indicated by setting bits in a bit-field.* Analog results are placed in the test buffer in a format appropriate to the result. For example, voltages or currents are kept in floating-point format, whereas the number of ringers is stored as an integer. The TSP is finally terminated by SER when no additional testing is required.

The SER process interfaces to TSP and SUP, and implements the test series. SER is a child process to SUP, but SUP terminates after SER is created and gains control of the test buffer. (In principle, it would not

* Examples of binary results are: bit 66, if set, means that the dc signature looks like a PBX; bit 25, if set, means BUSY SPEECH; etc.

be necessary for SUP to terminate after creating process SER, but the arrangement described is used to minimize the number of processes that must be active simultaneously, since each active process entails additional overhead.)

The SER creates child process TSP; SER and TSP take turns being active as discussed above. When SER has implemented enough of a test series to determine that "early" results to the RSA are required, process SUP is scheduled to send the results to the FE. The SER waits for SUP to terminate and then continues to the point that testing is completed. Then SER terminates the child process TSP, schedules process SUP, waits for SUP to gain control of the buffer, and then terminates.

The SUP process interfaces to MLTCCM and to SER. SUP provides general control over the test function, gains access to the line under test, and formats reports to the FE.

V. THE TESTING HARDWARE DESIGN

As previously mentioned, the testing hardware (Figs. 3 and 4) can be divided into two major subsystems: the LTF and the MMM. The LTF controls access to customer loops via no-test trunks and determines the busy/idle status of the loop. The MMM performs tests on idle loops for fault characterization.

Two interfaces to the LTF exist. The CCC provides the interface to the Controller. The TAS provides the interface to customer loops via no-test trunks to the central office switch.

The CCC serves as a multiplexer and demultiplexer for data to and from the Controller. Transmission is via a four-wire circuit using full duplex asynchronous serial data rates of 1200 or 2400 baud. The 1200-baud rate uses 202-type data sets for installations where the Controller and LTF are separated by more than 1500 feet. Within 1500 ft the transmisssion can be set to 2400 baud using optically isolated current loops.

Data from the Controller consist of an address field and a control-signal field. The CCC decodes the address and passes the control signal to the appropriate subassembly of the LTF/MMM. Data from each subassembly are digitally encoded by the subassembly and passed to the CCC which adds the output address to the data and transmits them to the Controller for interpretation.

The Access Switch Controller (ASC) receives control signals from the Controller via the CCC and provides signals to the TAS and the Equipment Access Switch (EAS). These signals select, hold, and release the crosspoints of these two switching matrices.

The TAS serves as a concentrator and permits the connection of up to 98 no-test trunks to one of nine test ports. The test ports are routed through the Supervisor and Control Circuit (SCC) to the EAS. The EAS permits the connection of any test port to either a dialer, busy detector, or MMM.

The SCC provides the control and monitor functions required to access a customer loop via a no-test trunk. Loop access includes the functions of sharing trunks with the local test desk, sleeve-lead supervision, ring-tip supervision, dialing, and busy detection. In addition to access, the circuit provides for hazardous potential detection and interruption of the test path to prevent equipment damage. Dialing circuits are provided for both dial pulse (DP) or multifrequency (MF) signaling.

Once the loop has been dialed and the sleeve lead manipulated to effect a cut through to the loop, a busy test is made. The busy test provides outputs of idle, switching-machine overflow, dc busy (battery ring to tip), and speech busy. If the loop is idle, the test port, and therefore the loop under test, is transferred to the MMM for testing. If the loop is other than idle, access to the loop is dropped and the loop status is reported to the Controller.

5.1 MLT measurement module description and operation

The MMM provides the loop-testing capability for the MLT hardware and consists of three major sections: the Source and Output Interface (SOI), the Test Package (TP) and the Test Package Auxiliary (TPA).

The SOI performs a service function for the TPs and TPA by providing clock frequencies for the digital circuits, precision leveled frequencies for analog sources, and a precision dc reference for analog sources. The SOI also contains output encoding circuits for the analog measured voltages generated by TPs and the TPA during loop tests. The SOI pools the TPs and TPA for outputs and, if present, they are connected to the SOI encoding circuits. The encoding circuits include a 12-bit analog-to-digital (A/D) converter, an auto ranging amplifier required to present properly leveled signals to the A/D, and a polarity sensor. The digitally encoded output measurement is temporarily stored in a RAM buffer along with its unique output address. The address is determined by which TP or TPA circuits are being serviced by the A/D. Upon receipt of a poll, the output data is passed to the CCC for transmission to the Controller.

The SOI also contains sources and terminations which are used to isolate hardware failures to particular circuit packs. This capability is controlled by the Sanity and Diagnostic software modules in the Controller (Section 2.3).

The TP (Fig. 6) is a multipurpose test circuit which is configured for a particular test by commands from the Controller. The TP is the heart

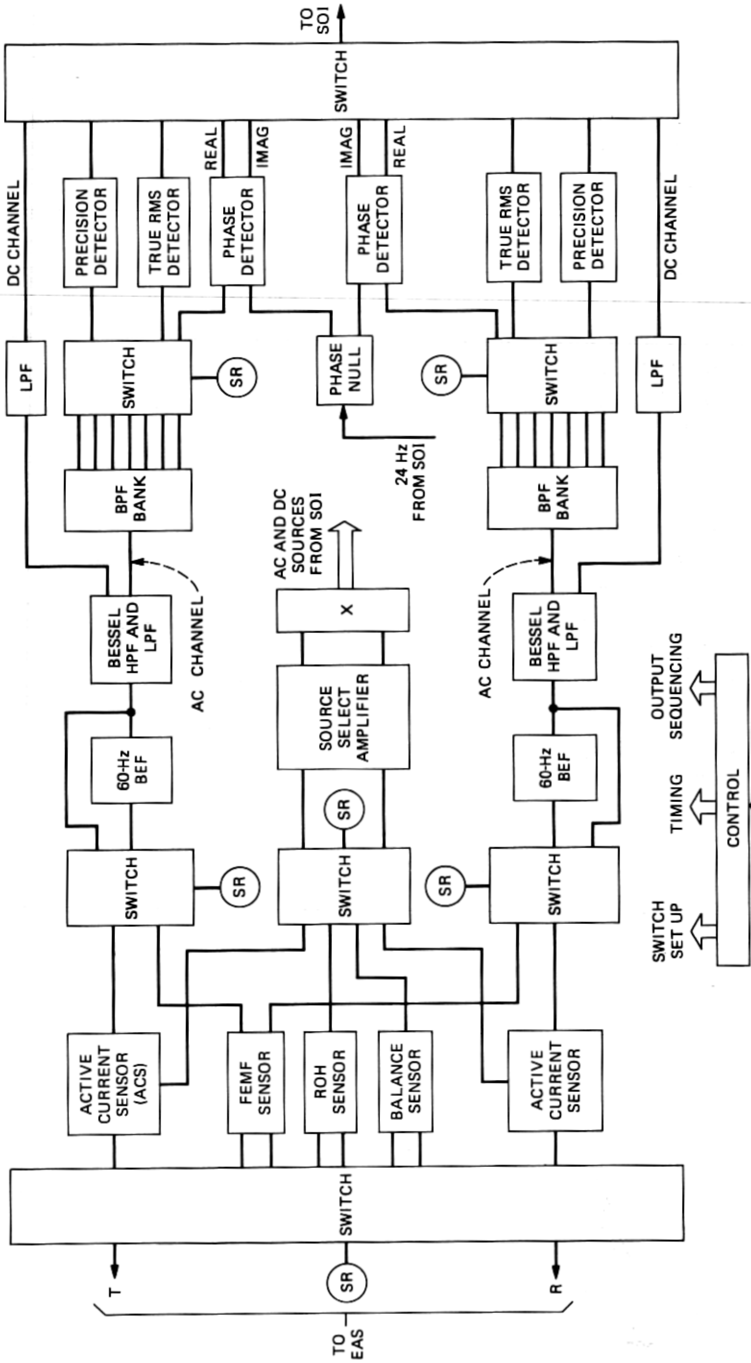

Fig. 6—Schematic of MLT measurement module.

of the MLT hardware. It can be divided into a test section, through which all analog test signals flow, and a control section.

The test section has as its input, the tip and ring of the loop under test. The output of the test section is a dc voltage proportional to the quantity being measured. The dc voltages are passed to the SOI for processing as previously described.

The test section has two essentially identical measurement channels which allow simultaneous testing of both tip and ring conductors. Each measurement channel consists of a number of analog signal-processing blocks which can be connected in various combinations, via switches, to set up the desired measuring circuit.

The analog signal processing blocks can be separated into four main categories: sensors, filters, detectors, and miscellaneous. The TP employs several types of sensors for the tests it performs.

The Active Current Sensor (ACS) senses the current flowing in the conductor under test. The dc and ac voltages ranging from 0 to 80 volts can be applied to tip and/or ring through the ACS, and the resultant current measured to give a measure of circuit impedance. Other sensors are provided for detecting foreign voltage, measuring the longitudinal balance of the loop, and identifying Receiver-Off-Hook (ROH) conditions.

The test package filters provide for rejection of 60-Hz noise by a 60-Hz band-elimination filter, separation of ac and dc signals by Bessel high-pass and low-pass filters, and selection of individual test frequencies by a bank of bandpass filters.

Three types of signal detectors are incorporated into the design. The phase detector produces a pair of voltages that are proportional to the real and imaginary components of the impedance at 24 Hz of the circuit under test. The RMS detector produces a dc voltage output equal to the true RMS of any ac signal or noise appearing at its input. The precision detector produces an output that is the average value of the ac signal appearing at its input.

Miscellaneous TP circuits are the Source Select Amplifer and the Phase Null circuit. The Source Select Amplifier selects the ac and/or dc voltage to be applied for a test and sets their levels dependent upon control signals received from the Controller. The phase null circuits automatically compensate for phase shift through the analog signal processing blocks and null the output of the phase detector during periods of idle time (no-test request).

Sanity Reference points (SR points in Fig. 6) are provided for signal injection or passive termination connection by the SOI during self-check (known as Sanity and Diagnostics) testing.

The control section of the TP receives instructions from the Controller via the CCC, operates the test package switches which control test configuration, times the set up and progress of all tests, and controls the sequencing of connection of the outputs to the SOI encoding circuits.

The third major section of the MMM is the TPA. Unlike the TP, the TPA is not a configurable test circuit. It is designed to perform specific tests which, due to their length, would be an inefficient use of the TP resources. Two test functions are provided: Rotary Dial Analyzer and Dial Tone Analyzer.

The Rotary Dial Analyzer contains a signal-conditioning circuit which performs dc restoration and signal clamping required to interface with logic measuring circuits. The dial pulses gate a clock signal to two digital counter circuits. The output registers of the two counters are used to determine the number of dial pulses received and the speed and percent break of the rotary dial being tested.

The Dial Tone Analyzer presents a constant-current sink for the switching-machine dial-tone generator. This sink provides a 20-mA load to the generator regardless of the length of the no-test trunk, thereby simulating worst-case loop conditions. The sink can be configured for loop-start, ground start, or ground start reverse line circuits.

The constant-current sink is followed by a band-pass filter to separate the dial-tone frequencies from noise signals. A precision detector converts the received dial-tone voltage to a dc level compatible with the encoding circuits of the SOI.

5.2 Test capabilities

A brief description of the test capabilities follows:

DC Thevenin—This test measures parameters required to form a three-terminal dc Thevenin equivalent circuit looking into tip, ring, and ground. It identifies resistive faults, dc foreign voltage, crosses to working pairs, ground start PBX signatures, central office line-circuit faults, and lines on intercept.

Short Circuit Current—This test measures the short circuit noise current tip to ground and ring to ground.

Three Terminal Admittance—This test uses the phase detector in the TP to determine the real (resistive) and imaginary (capacitive) components at 24 Hz of the loop between tip, ring, and ground. This measurement is used to determine the length of no-test trunks, the length of loops, to detect POTS ringers, to identify whether a pair is open on tip, ring, or both sides and whether the open is in or out of the central office, and to measure the distance to the open if it is out of the central office.

Open Circuit ac and dc Foreign Voltage—This test measures the open circuit dc foreign voltage and the ac voltage appearing tip to ground and ring to ground.

Receiver-Off-Hook (ROH)—This test uses the nonlinearity of the off-hook station set to discriminate between a tip-to-ring resistive short and an off-hook set.

Thermistor Heating—This test provides a means of detecting the presence of thermistors in the alerting circuits of PBXs and Key Telephone Sets by detecting the change in the real part of the admittance at 24 Hz as a voltage is applied by the TP.

Ringer Counting—This test provides a means of counting the number of ringers connected tip-to-ring, tip-to-ground, or ring-to-ground. Cable capacitance is differentiated from ringer capacitance by an algorithm that compares loop admittance at several frequencies.

Longitudinal Balance—This test provides measurements for determining loop balance to 65 dB at 200 and 800 Hz.

Soak—This test measures the time variance of a resistive fault with an applied dc voltage.

Dial Tone Analyzer—This test measures whether dial tone can be drawn and broken. It indicates whether the dial tone is drawn normally (within 3 seconds) or slowly (3 to 6 seconds).

Rotary Dial Analyzer—This test counts the number of pulses from a rotary dial, and measures the dial speed and percent break.

5.3 Calibration

To assist in improving measurement accuracy, the ability is provided to perform no-test trunk calibration. Dedicated telephone numbers, which are open circuited on the main distributing frame, are accessed and tested to determine whether trunk faults exist and to determine the length of the trunk. The length measurements are used in making decisions on the locations of open faults.

Component aging may cause a degradation of the measurement accuracy. To compensate for this effect, the TP is designed to permit measurement of critical ac and dc gains and offset voltages. These parameters are stored in the Controller and used in the associated software test algorithms. The calibration is initiated via a system request at the FE.

As previously mentioned, test nodes are provided in the TP and TPA where sources and/or passive terminations in the SOI can be applied, under software control, to isolate component failures to particular circuit packs. The SOI also cotains circuits that permit testing of the A/D encoding circuits and the RAM buffer.

5.4 Packaging

The LTF is mounted in a standard 7-ft UNIFRAME consisting of dual 38-in. bays (Fig. 3). The design is modular and connectorized, which permits growth from a minimum to a maximum system by the addition of circuit packs of plug-in subassemblies. The minimum LTF/MMM serves up to approximately 15,000 lines and the maximum LTF/MMM serves approximately 250,000 lines.

The minimum LTF/MMM consists of one TAS that can accommodate up to 18 no-test trunks, circuit packs to activate two test ports, either two dial pulse or two multifrequency dialers, one Test Package (TP), and one Dial Tone Analyzer. The maximum LTF/MMM consists of four TASs to accommodate 98 no-test trunks, circuit packs for nine test ports, two dial pulse and two multifrequency dialers, four TPs, two Dial Tone Analyzers, and one Rotary Dial Analyzer.

VI. CONCLUSION

The MLT system had its field trial in Nashville, Tennessee, beginning in late April 1978. By the end of 1978, four Bell Operating Companies had implemented standard MLT systems produced by Western Electric and, by the end of 1979 thirteen companies were using MLT. Conversion to MLT systems during 1979 and 1980 was running at almost 2 million lines per month.

The economic and operational success of the MLT system is documented elsewhere in this issue.[2] At this writing, Southern Bell Telephone and Telegraph Company has installed more MLT systems (197 LTFs and 59 Controllers by the end of 1980) than any other Bell Operating Company. In Southern Bell, the operational availability of the LTFs and associated MMMs is running at 99.7 percent, while the overall availability (Controllers, data links, LTFs, etc.) of the system is currently 99.3 percent.

VII. ACKNOWLEDGMENT

The MLT design, development, and introduction was a large team effort involving the contributions of many highly skilled systems engineers, electrical engineers, computer scientists, physical designers, and experimental psychologists. To single out individuals would be unfair to those not mentioned, so the authors wish to acknowledge that the ideas and results stated in this article belong to those many individuals who gave unselfishly and tirelessly to the creation and implementation of the MLT system.

REFERENCES

1. P. S. Boggs et al., "Automated Repair Service Bureau: Evolution," B.S.T.J., this issue.
2. E. A. Overstreet, "Automated Repair Service Bureau: Economic Evaluation," B.S.T.J., this issue.
3. O. B. Dale, "The Evolution of the Automated Repair Service Bureau With Respect to Loop Testing," International Symposium on Subscriber Loops and Service—Conference Record, March 20–24, 1978, IEEE Cat. No 78CH1279-9 COM.

4. F. J. Uhrhane, "Automated Repair Service Bureau: Loop Testing Strategies and Techniques," B.S.T.J., this issue.
5. G. H. Leonard and J. E. Zielinski, "Automated Repair Service Bureau: Human Performance Design Techniques," B.S.T.J., this issue.
6. R. F. Gauthier and W. A. Harris, "Automated Repair Service Bureau: Two Examples of Human Performance Analysis and Design in Planning the ARSB," B.S.T.J., this issue.
7. R. L. Martin, "Automated Repair Service Bureau: The System Architecture," B.S.T.J., this issue.
8. S.G. Chappell, F. H. Henig, and D. S. Watson, "Automated Repair Service Bureau: The Front-End System," B.S.T.J., this issue.