

GBPPR 'Zine



Issue #57 / The Monthly Journal of the American Hacker / January 2009

"I love it. A 16 billion dollar deficit and '521 layoffs.' Wow, they are really attacking that problem. Don't worry, January 20th is coming and all things will be fixed. There is a wind at his back and he will solve this. The messiah is going to tax all the people who pay taxes so we can give it to those who don't pay taxes. Pretty soon, we will all be able to get a check from the 'one' because no one will be able to pay taxes. I also want my tax cut and free health care from Hussein. Oops, I think Gov. Patterson just took my Obama money."

--- Comment in a December 17, 2008 *New York Daily News* article about Gov. David Paterson's new "budget."

(www.nydailynews.com/ny_local/2008/12/16/2008-12-16_gov_david_paterson_unveils_dire_new_york.html)

Table of Contents

- ◆ **Page 2 / Software System Introduction – Software Description / #1A ESS (Part 4)**
 - ◆ Detailed analysis of the software "generic" program which a #1A ESS operates under.
- ◆ **Page 28 / Through-the-Wall Motion Detection Device**
 - ◆ Use parts from an automatic door opener to detect motion in hidden areas.
- ◆ **Page 37 / DMS-100 Test Cross Connections Table (TSTXCON)**
 - ◆ Remote headset to external trunk mapping under a DMS-100.
- ◆ **Page 39 / Two-Tone Pager Decoding Using Multimon**
 - ◆ Use the Linux program 'multimon' to decode a two-tone pager signal and send a text message.
- ◆ **Page 45 / Bonus**
 - ◆ TNB
- ◆ **Page 46 / The End**
 - ◆ Editorial and rants.

C. Call Event Codes

4.356 The call event code indicates how the call was disposed. The SMDR call event codes include:

- 0 - Completed directly
- 1 - Queued and completed
- 2 - Invalid NPA or NXX
- 3 - Invalid authorization code
- 4 - Insufficient FRL
- 5 - All facilities busy
- 6 - Abandoned on queue
- 7 - Timed out from queue
- 8 - Miscellaneous failure without queuing
- 9 - Miscellaneous failure after queuing.

Codes 0, 5, and 8 are also used for XMDR.

D. Service Feature Codes

4.357 The service feature code indicates that there were feature interactions on the call which affect the contents of the record.

- (1) Station billing on attendant handled call applies.
- (2) The record applies to the base to remote portion of a forwarded call.
- (3) The call was routed to the attendant, due to the toll diversion feature (XMDR only).

AUTOMATIC CALL DISTRIBUTION (ACD)-ESS SWITCH MANAGEMENT INFORMATION SYSTEM (AEMIS) DATA BASE (MSDU)

A. General

4.358 The ACD-ESS Switch Management Information System (AEMIS) (available with ACD2 only) is a minicomputer-controlled system designed to:

- Measure and analyze agent/traffic data and provide detailed agent/traffic information

- Performance calculations
- Summarize past history
- Short-term forecast to the ACD manager.

4.359 To perform all of the AEMIS functions, a data base of the necessary data (describing the ACD) has to be established for the AEMIS by the No. 1/1A ESS switch. This is accomplished by the management information data base update program (MSDU) via a centrex data link. MSDU uses the 1-second entries provided by the block data link loading function of the centrex data link (DLIO) feature to format and load the data link orders. System configuration and control requires the inquiry-response system (IRES) feature; therefore, the IRES feature must be loaded for the AEMIS feature. The necessary data for the AEMIS data base includes:

- (a) time of day
- (b) the AEMIS trunk groups and associated trunk network numbers
- (c) the facilities
- (d) the queue data
- (e) the agent to functional group assignments for each load compensating package (LCP) and the active LCP
- (f) the four 4-digit extension assigned to each agent terminal.

4.360 Time of Day: The time-of-day function gives the AEMIS a snapshot of the ESS switch real-time clock. The time sent to the AEMIS is the year, month, date, hours, minutes, and seconds. The AEMIS resets the PDP*-11 clock to equal this time.

4.361 Call Store Configuration: The call store configuration function provides the AEMIS with a snapshot of the ACD changeable data, namely, the active LCP, the functional group (FG) patterns of the active LCP, and the queue data. The active LCP is the current invoked LCP. The FG patterns are the FG patterns of the active LCP plus any changes made by the ACD customer. The queue data is the interflow threshold, primary outflow threshold, and the secondary outflow threshold; if the night directory number*Trademark

SECTION 231-045-005

ber (DN) is call forwarded, the forwarded DN is also sent; if not forwarded, all zeros are sent.

4.362 Initialization or Program Store Refresh: Both the initialization and the program store refresh functions send the same data to the AEMIS. The distinction is the rate at which the data link orders can be loaded into the data link output buffer. For the initialization request, the maximum rate is 20 data link orders per 1 second entry; whereas the maximum rate for the program refresh is 10 data link orders per 1-second entry. The data that is sent to the AEMIS for either function is:

- (a) All the trunk network numbers for each trunk group number associated with the AEMIS
- (b) All of the rows of data for each functional group for all of the LCPs in the data group associated with the ACD
- (c) The number of simulated facilities for each simulated facility group associated with the AEMIS
- (d) All of the agent terminals in the data group and their 4-digit extension number associated with the ACD
- (e) The inflow threshold, call waiting lamp threshold, primary outflow threshold A, primary threshold B, primary alternate server pool number, secondary alternate server pool number, queue size, number of queue registers, inflow queue indicator, functional group number associated with this queue, DN of this queue, base night DN of this queue, and the primary alternate server pool.

The AEMIS can also request a subset of the initialization or program store refresh data. That is, any of the individual blocks of initialization or program refresh data can be requested separately.

4.363 When interrogation requests are received by ESS switch, appropriate data is sent to the AEMIS to satisfy these requests. This data may include a copy of the current program store data and a call store configuration or some subset of this program store.

4.364 In addition to sending the AEMIS data to satisfy the interrogation requests, the ESS

switch sends a continuous stream of messages describing the call processing activity of the ACD customer. In order to report events to the AEMIS minicomputer, the ESS switch keeps a record of each incoming or outgoing call over customer trunking facilities and simulated facilities group. The ESS switch also keeps track of calls terminated to and originated from the agent consoles in order to maintain a record of the agent console state.

4.365 The AEMIS messages themselves may consist of one or two 24-bit words: 23 data bits, and one parity bit. The bits are numbered from right to left (0 through 23). Bit 23 is the parity bit. Bit 22 is a maintenance bit. When the maintenance bit is zero, this indicates an ESS switch maintenance request. When bit 22 is a one, the data link message contains AEMIS data. Bits 21 through 17 in single word messages contain the operation code (SOP). Bits 21 through 17 in the first word of a double word message are always set to "11101." The operation code (DOP) is contained in bits 16 through 13. Bits 17 through 21 of the second word of a double word message are always set to "11111" as an indicator that this is the last word of a double word message. The individual SOP and DOP code messages are listed in Fig. 23.

B. Call Processing

4.366 A series of call processing AEMIS messages are generated whenever an ACD simulated facility or a dedicated ACD-ESS switch trunk becomes involved in a call.

4.367 The sequence of facility messages that are sent to the AEMIS is essentially identical whether a simulated facility or a trunk is used. The messages sent to AEMIS are as follows:

- (a) Facility seizure message (SOP2 for trunks, DOP1 for simulated facilities)
- (b) Facility queued (DOP2)
- (c) Facility dequeued (SOP3)
- (d) Facility connected (DOP0)
- (e) Facility idle (SOP4).

4.368 As indicated in (a) above, the facility seizure messages are unique for trunks and simulated facilities as shown below.

- (a) Bit 16 of the SOP2 message is 0 for incoming trunks and 1 for outgoing trunks. When a trunk is seized and becomes traffic busy, the SOP2 message must be sent. The only exception to this is trunk seizures for a receiver attachment delay report (RADR) test. No message is sent on a RADR seizure.
- (b) Bit 16 of the second word of the DOP1 has the same function for simulated facilities.

4.369 In all facility messages a constant identifier, the facility number field (bits 14 through 0), is used throughout the call as a tag. When the facility is a trunk, the facility number field contains a trunk network number; bit 15 is 0 to indicate a trunk.

4.370 When a simulated facility is involved, the facility number field contains a simulated facility register address (bits 2 through 0 of the address is truncated in bits 14 through 0). Bit 15 is 1 to differentiate a simulated facility from a trunk. In addition, bit 14 of the facility number is always 1 to differentiate a simulated facility register from a queuing register.

4.371 In addition to the call processing facility messages, AEMIS messages are sent for various trunk maintenance states. These states may be initiated either via the TTY, the trunk and line test panel, or as a result of a hardware failure during call processing. The AEMIS maintenance messages are:

- (a) Trunk disabled (SOP5)
- (b) Trunk high and wet (SOP6)
- (c) Trunk locked out (SOP11)
- (d) Trunk active-in-service (SOP12)
- (e) Trunk make busy (TMB) or carrier group alarm (CGA)-(SOP7).

CENTREX STATION REARRANGEMENTS (CSR)

A. General

- 4.372** The CSR feature allows centrex customers to directly access the No. 1/1A ESS switch to:
- rearrange extensions
 - activate and deactivate extensions

- change certain features
- display information about extensions.

Before CSR, a centrex customer could make changes to their centrex group only through a service order, which could take several weeks. With CSR, the change takes place immediately.♦

5. MAINTENANCE SOFTWARE FUNCTIONS

5.01 In order to maintain a high degree of continuous and reliable service, the ESS switch provides duplicated equipment units, special maintenance circuits (which detect troubles and provide diagnostic access), and a comprehensive programmed system of maintenance programs. The ESS switch maintenance programs are used to detect and localize system troubles as well as control system configuration changes which may be required for maintenance purposes. Figure 35 provides an overview of ESS switch maintenance control structure.

5.02 The ESS switch maintenance software can be grouped into the following functional areas:

- Maintenance Control
- Fault Recognition (FOR)
- Diagnostics (DIAGs)
- Routine Exercises (REXs)
- Audits
- Diagnostic Results Processing.

5.03 The FOR programs attempt to recover the call processing ability of the system by establishing a working configuration of office equipment when a trouble is detected. These programs are non-deferrable and are of the highest priority in the maintenance program hierarchy. In general, FOR programs are initiated as a maintenance interrupt when a trouble is detected. The FOR runs to completion under interrupt control, after which call processing is allowed to continue undisturbed. The primary function of the FOR is to determine whether the faulty unit, as detected, has permanently affected system operations and whether substitution of a duplicate system unit is necessary to restore normal

SECTION 231-045-005

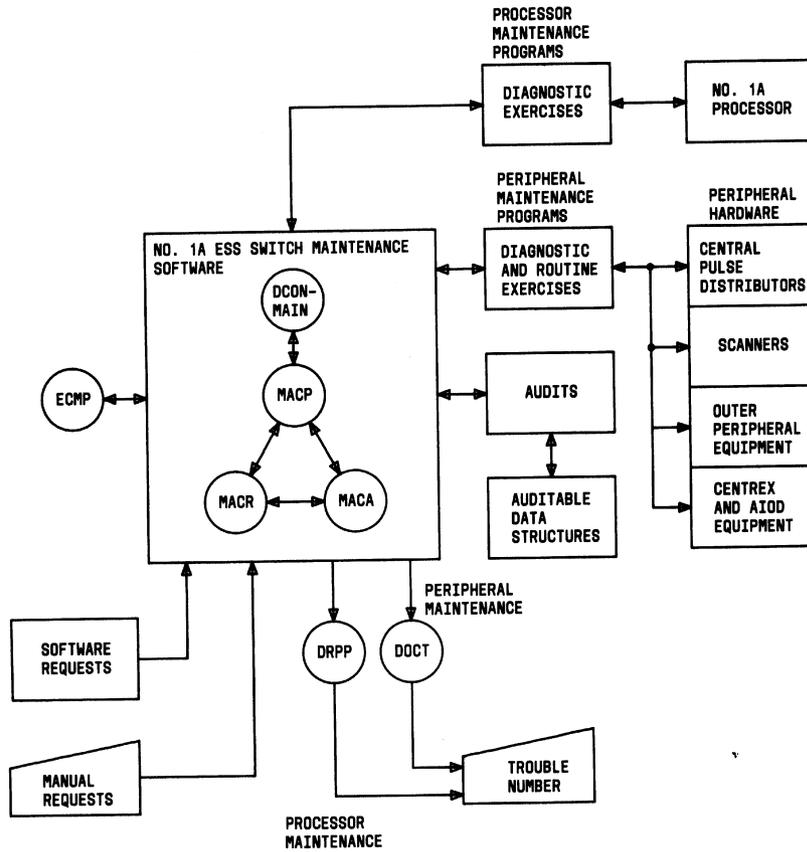


Fig. 35—No. 1A ESS Switch Maintenance Control Structure

operation. The maintenance control software has no part in the execution of FORs other than to recognize that an interrupt has occurred and to remove any related maintenance client currently in control of the maintenance scratch area. The FOR may request maintenance control to run a DIAG during normal base level maintenance.

5.04 The DIAGs are the second highest priority maintenance programs handled by maintenance control. DIAGs attempt to localize a trouble

within a faulty unit down to the smallest possible number of replaceable circuit packs. A DIAG consists of a series of sequential tests which are executed against the suspected faulty equipment type. Results from the tests are converted into a TTY output message by dictionary programs. The resulting output message provides a trouble number which is used as an index into a trouble locating manual (TLM). The TLM provides a list of circuit packs which, if faulty, could have produced the trouble number.

5.05 The REX programs consist of both scheduled automatic routine exercise programs (AEX) and demand request routine exercise programs. The REX programs are designed to check for faults that might otherwise go undetected, to search for uncorrected errors, to check the trouble detection circuits, and to exercise infrequently used hardware.

5.06 Audit programs comprise a system, complete with control structure, designed to maintain the validity of stored data structures in the ESS switch memory. Audits can detect, repair, and reinitialize data elements or structures as necessary.

5.07 Diagnostic Results processing is an automated means of assimilating the so-called raw data produced by the diagnostic programs. The resultant output from the diagnostic results processing programs may consist of a suspect list of faulty equipment or an index into such a list.

MAINTENANCE SOFTWARE CONTROL PROGRAMS

A. Maintenance Control Program (MACP)

5.08 In the No. 1A switch, MACP schedules and controls the execution of deferrable base level maintenance programs, eg, hardware diagnostic and routine exercise programs as well as other nonmaintenance programs. Specifically, all paged programs are executed under MACP control. The MACP interfaces with the paging program (PAGS) to accomplish the run time loading and execution of paged programs.

5.09 The MACP consists of a group of control routines which use application dependent job tables (MTBL, MJTB) to perform job scheduling for a particular application. These job tables specify the MACP resources and client jobs and, thus, define the MACP environment for the application system. A set of macros and subroutine (MAPL) is provided by MACP which can be used by client programs for delay timing, inhibiting maintenance interrupts, etc.

B. Base Level Scheduled Maintenance

5.10 All noninterrupt maintenance in the No. 1A ESS switch peripheral (ie, not processor) area is initiated by an entry to pident MACA. This entry is scheduled as one of the class E jobs in the ECMP base level task dispenser. Pident MACR (maintenance control peripheral program) is entered to set

up the correct return to the ECMP and to check for interfering MACR jobs in a multi-MAC environment. Pident MACA then checks a counter to see if trunk maintenance should be done. If not, a check is then made to see if audit work should be done. Scheduled routine audits are run in parallel with other No. 1A ESS switch maintenance. During the normal base level cycling, audits are run every third pass through this routine. If it is not time for audit work, control is passed to the job control routine in MACP. The MACP will then continue a client already in progress at the client's next segment address or, depending on the scheduling algorithm, look for other maintenance work. The MACR provides the basic control structure for peripheral maintenance as described below.

Diagnostic Requests

5.11 The MACP enters MACR when looking for diagnostic requests. Checks are made to see if MACR is currently busy in the other maintenance class. Pident MACR, as a client of MACP, runs out of the maintenance class, subclass 0 or subclass 1. Pident MACR can only run one peripheral maintenance job at a time; ie, if MACR is busy in subclass 0, a new job cannot be started in subclass 1. If upon entry, MACR finds itself busy in the "other class," then control is passed back to MACP to abort the job. If MACR is available to start the diagnostic, then MACP is given a common abort address. Should MACP abort the job, this common abort address would be given to the peripheral maintenance client in progress. Then MACR will hunt for diagnostic requests using a hunt routine. Refer to the MACR program listing for a detailed account of diagnostic request processing.

Routine Request Table Requests

5.12 The routine request tables store requests (from the maintenance personnel or other maintenance programs) to start a particular peripheral maintenance program. The routine request table (RRT) is divided into two levels: RRTA and RRTB. Manual requests are stored in RRTA; requests from other programs are stored in RRTB.

5.13 MACR provides entries for loading the RRTs. A fail return to the client is made if there are no available entries in the RRTs. If the proper entry to MACR is taken, an entry into the top slot in level B can be guaranteed; all previous entries are pushed down one entry. The lowest entry may be destroyed.

SECTION 231-045-005

Normally, each entry loaded into level A or B is loaded following any previous entry. Requests to run jobs are answered starting at the top of each RRT level, thus providing a first-in, first-out sequence. For each job, MACR will schedule a peripheral maintenance search of the RRT from MACP.

5.14 To process a job requested in RRTA, MACP enters MACR under control of the job scheduler routines in MACP. Again checks are made to see if MACR is available. Note that the same initial checks were made for the MACSDIAG entry. As entries are processed out of RRTA, the remaining entries are moved up one position. The MACP is entered to run the job.

5.15 For job requests in RRTB, MACP again enters MACR. The initialization checks are identical to those described earlier, down to the point where the RRTB is examined for requests.

AEX Job Requests

5.16 The AEX jobs are run on a fixed schedule, eg, per half-hour (on the half-hour and hour), per hour (on the hour or half-hour), every three hours, or daily at midnight. MACR controls two classes of AEXs, ie, class II routine exercises and class III routine exercises. The class II jobs are run out of the AEX2J1 and AEX2J2 job tables. Individual jobs are turned on by corresponding bits in flag words M4FLG1 and M4FLG2, respectively. These jobs are scheduled by an entry every half-hour from the ECMP.

5.17 The jobs in the AEX2J1 table are nonhardware testing programs and routines. Their basic functions are information reporting and auditing or verification. The jobs in the AEX2J2 table are solely concerned with hardware testing.

5.18 The class III routine exercise programs are the lowest priority programs in the ESS switch peripheral maintenance software structure. These jobs, run out of the MACR AEX3JB job table, are executed during spare time when no other AEX (class II) jobs are waiting. The AEX3JB jobs are turned on by a corresponding bit in M4AEX3; these bits are also set by ECMP visits to MACR.

5.19 To start an AEX job, MACP enters MACR. As with other peripheral maintenance requests, checks are made to see if MACR is busy. If busy, the

job is rescheduled for a later attempt. If MACR is available then an abort address is given to MACP.

C. Fill Maintenance

5.20 During periods of exceptionally light traffic, or when the call processing load is low, the ECMP base level task dispenser may run out of work to do. Rather than waste time looking for work that is not yet scheduled, the available time is used running audits as "filler." Audit control responds as for a scheduled entry.

AUDIT SOFTWARE

A. General

5.21 The audit program package provides an effective and rapid method of protecting the ESS switch from errors in temporary and permanently stored data structures. The audit system has access to system storage as shown in Fig. 36.

B. Audit Error Detection

5.22 The audit system is tasked with both detecting and correcting (if at all possible) software data structure errors which may occur from a great many sources. In general, an audit consists of an evaluation of a data item, ie, bit(s) word(s), etc, and based on the evaluation (actual value or data credibility), a pass/fail decision is made. If errors are detected, the audit in control or other audits may attempt to correct the faulty data. Data structures which can be audited exhibit one or more of the following properties:

- (a) Constant Data: The simplest auditable memory is that which is known to contain constant information. Errors can be detected by simply comparing the backup or redundant data with the actual data.
- (b) Timed Memory: Certain types of data structures are known to have short holding times relative to the length of a call. Other structures are not allowed to be in a transient state for a long period of time. Therefore, it is possible to monitor these structures and if a certain threshold time limit is exceeded, the memory is deemed to be in error. The outpulsing register is a good example of a timeable piece of software since no call should be in the outpulsing state for more than a few seconds.

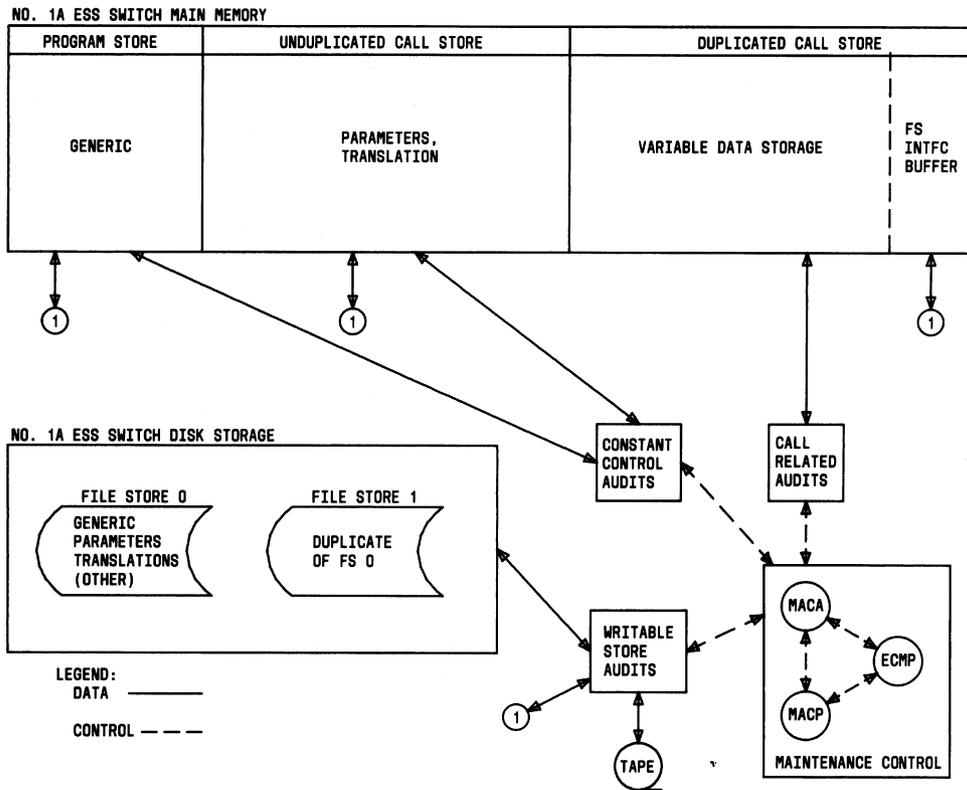


Fig. 36—No. 1A ESS Switch Audit System

(c) Redundancy: If a data structure contains redundancy, then all additional pieces of information may be compared by an audit to determine if they agree. If total agreement is not found, then the data structure is assumed to be in error. In many cases this redundancy is not essential to the normal processing of the data structure and must be added at some expense of real time or memory to provide auditability.

C. Correction Strategy

5.23 On detection of an error in the constant data, an audit will correct the erroneous informa-

tion from the backup data. An error in a more sophisticated data structure or a time-out will cause all identifiable memory associated with the structure to be idled and maintenance to be requested on all identifiable hardware. In general terms, then, no attempt is made to return the incorrect structure to its proper operational state. Any incorrect guess by the audit as to the proper busy state could be fatal to system sanity. This strategy was chosen as the safest way to guarantee a valid software state.

5.24 In some rare cases where there is an overwhelming N-to-1 vote for the state of a piece

SECTION 231-045-005

of software, the one dissenting piece of information will be changed to agree with the majority.

5.25 After the error is cleared, an audit message is printed on the TTY describing the error. Other audit messages will be printed for each additional piece of equipment restored or memory idled. These messages are printed to notify the craftsman that an error has been detected and are to be used in correcting the source of the error.

5.26 Finally, other audits are requested at high priority to examine all memory functionally related to the data structure in error.

D. Audit Subsystem Requirements

Audit Segments

5.27 An audit segment is defined to be that amount of base level time in which an audit runs without giving up control or taking a time break. The actual time allotted is a theoretical figure for maintenance programs determined by systems engineering. This figure is considered to be the amount of time in which base level call processing can be suspended without unduly influencing the system.

Short Segment

5.28 A short segment is an untimed segment that is known to run less than 2.4 ms under error-free conditions. A short segment audit usually has a very simple job whose execution time can be computed by counting cycles.

Timed Segment

5.29 Timed segment uses a timing method external to the audit to determine elapsed segment time. Once the proper segment time has passed, a flag is set by the external timing program. This flag is consulted by the audit at convenient points in the program where a real-time break would be allowed. If the flag is set, then a segment break is taken. Routines are provided to do this timing for the client audit. Timing is done using the maintenance timer. Since this timer measures elapsed real time and J level consumes 30 to 50 percent of that time, the timer must be set to a value equal to 2.4 ms plus the time consumed by the average J. By selecting an average J of 2 ms or 40 percent of all time, the audit will get less time at peak traffic and more time in an idle

system, thus making segment times somewhat dynamic in response to load. Timing subroutines can set the clock with the G-level interrupt inhibited so that time-out can be detected by simply reading the clock to determine if it is zero. It is not necessary to zero the clock at the end of an audit since time-out causes no system action.

Interject Long Segments

5.30 Audits that must run for long periods of time without a real-time break must run an interject long segment. This is necessary because certain call configurations cannot tolerate delays of this length without base level action. The interject segment warns the system not to start any delay sensitive jobs and then waits six peripheral order buffer cycles to allow any delay sensitive jobs that have been started to complete. The audit runs its long segment in interject and then takes an appropriate number of "do nothing" segments to allow the system to recover call processing ability. During the peripheral order buffer cycle wait period, the normal maintenance control program entry is shut off.

♦**Note:** Some audits do a fixed number of "do nothing" segments, others do a variable number based on the amount of work performed by the audits.♦

Base Level Scratch

5.31 The audit system is allocated a large block of scratch area in call store. *

Audit Holding Time

5.32 Audit holding time is that interval of real time in which an audit is holding the maintenance control (MAC) scratch pad. Audit cycle time is the sum of the holding times for all audits in the system. It is generally measured as the time between one pass of audit 30 and the next pass. This ensures that all audits have been run at least once. Determination of exact audit cycle and phase times is a very complex process depending on a large number of variables. Better figures are obtainable by manual timing in the field under actual conditions and even these results can vary widely from office to office, or within a single office at different times during the day. Keeping this in mind, the rest of this section attempts to formulate some guidelines for predicting audit times for a new office by comparing its parameters to base offices whose audit times are known.

Audit Cycle Time

5.33 Audits are run primarily as fill work when the main program has nothing else to do. The scheduled routine maintenance entry to audits is from main program job class E, a relatively infrequent entry as compared to the fill entry. Since most auditing is done in spare real time, and spare real time is a function of traffic in the system, then audit cycle time (ie, time required for one complete pass of all routinely scheduled audits) increases with traffic in the system.

Traffic

5.34 Individual audit holding times are also a function of traffic. Exact relationships are not clear since each audit responds differently to traffic stimulation. For instance, the map audit takes only a few cycles to process an idle path, but a few thousand to process a busy path. On the other hand a constant audit is unaffected by traffic, and an idle link list audit is inversely proportional to traffic since there are fewer idle structures to process during busy periods.

Number of Networks

5.35 Audit cycle time is almost directly proportional to the number of networks in an office. Increasing the number of networks also increases the number of trunks and lines to be audited, and it indirectly increases the amount of auxiliary memory to be audited, such as registers, peripheral order buffers, and number of call stores. By making audit cycle time a function of the size of an office, as expressed by the number of networks, it is obvious that a large office has more audit work than a small office.

Phase Times

5.36 Phase times are responsive to many of the same factors that govern audit cycle times but in different degrees. Obviously available real time is not a factor in phases because stitched audits do not run out of class E or as fill work and do not take real-time breaks for call processing. Traffic is not a variable in phases 1 or 6 but is important in a phase 4 where additional work must be done for each busy or transient connection. The number of networks is the primary factor governing the length of phase 1 with the phase 1 time being almost directly related to network size. ♦Phase 4 through 7 are comprised of signal

distributor audit actions. The signal distributor actions take 18.6 seconds and are invariant, so long as the audits 6 through 72 takes less than 18.6 seconds, office size is not a factor.♦

Audit Scheduling

5.37 Audits are run by maintenance control on a demand basis if software errors have been detected in the system or if a TTY audit request has been made. The audits are run on a routine basis (class E main program entry) if no higher priority maintenance work exists in the system. This routine mode serves as the systems chief protective defense against unknown software errors. Additionally, audits run as fill work when the main program job scheduler has available real time with no scheduled work to do. All other audit functions, such as phases, demand audits, and interrupt level audits, are after the fact and serve to clean up known errors.

Request Tables

5.38 The audits are requested out of five request tables (audit control blocks):

- Phase (runs audits in a phase)
- Demand high priority
- Demand low priority
- Routine high priority
- Routine low priority.

Each request table consists of three call store words. Each bit in the request table corresponds to a single one of all the possible audits. Requests are served by scanning the tables in order of priority (highest to lowest) from right to left until a 1 is detected. Once a 1 has been found, its corresponding bit position is zeroed in all tables to clear any other requests for that audit, and its bit position is used to index a vector table to run the correct audit. The priority structure ensures that certain critical audits will have execution priority over the least critical ones.

Routine Scheduling

5.39 Audits running as fill work get a smaller and smaller percentage of real time as traffic increases. This design provides a valuable dynamic

SECTION 231-045-005

audit response to load conditions. The system is simply trading some of its audit protection for increased call capacity as needed. This technique is useful but it has the disadvantage of providing the least protection at a time when the probability of errors and vulnerability of the system is greatest. Conversely, it provides the most protection in an idle system where the error rate is very low. In addition, relying solely on the class E visits to audits greatly increases the audit cycle time while the higher traffic causes a greatly increased error propagation rate. These two diverging trends cause a consequent decrease in the probability that the audit needed to detect a specific error will run within a short enough time span of the occurrences of the error to prevent further call degradation, interrupts, or phases. This routine protection function is also shared with the less detailed and faster checks of the data validation. The data validation, since it has a fixed entry rate, is not subject to traffic variations and is thus more likely to detect catastrophic errors during peak traffic situations.

E. Audit Activity During a Phase

5.40 During a system reinitialization or "phase," audits are executed in a "stitch" mode; ie, the audits comprising the phase are stitched together one after another with no real-time breaks taken. All other central control processing activities are suspended until the phase is completed.

Audit Output Messages

5.41 The audit system provides output messages to keep the office maintenance personnel aware of audit system activities.

DIAGNOSTIC AND EXERCISE SOFTWARE

A. 1A Processor Diagnostics

5.42 The purpose of 1A processor equipment diagnostic programs is to provide a programmed test capable of verifying that a unit is free of classical faults, and if a fault exists, to assist in locating the fault to a small number of replaceable circuit packs. The diagnostic programs are entered under the following conditions:

- (a) For fault location in the automatic fault recovery sequence following detection of a trouble
- (b) For verifying the integrity of a growth unit before it is made available to the system

- (c) For verifying the integrity of a unit before it is returned to service following a restoral request at the equipment frame

- (d) As an error analysis function for locating marginal circuit packs

- (e) As an automatic routine exercise function

- (f) As an on-line or off-line troubleshooting tool with special options via the TTY.

5.43 Diagnostic programs for processor units have many common structural features, in that:

- (a) All diagnostic programs are written in a high-level diagnostic programming language.

- (b) The compiled output of the diagnostic language is in a data table.

- (c) This data table is interpreted at execution time by a set of task routines linked together by a task dispenser.

Inasmuch as these and other common features exist, a common diagnostic control program, DCONMAIN in No. 1A ESS switch, oversees the individual diagnostic programs and performs many of the common functions needed by the individual unit diagnostic programs. DCONMAIN is the program store resident interface between these individual diagnostic programs and other ESS switch programs. Specifically, DCONMAIN has responsibility in the following areas:

- (a) Overall control of the diagnostic programs

- (b) Interfacing with MACP for execution of diagnostic requests and with PAGS for paging operations

- (c) Interfacing with fault recovery and error analysis programs for prediagnostic initialization and post-diagnostic final handling

- (d) Interfacing with the diagnostic results post-processing program (DRPP) for the implementation of trouble locating procedures to identify suspected faulty circuit packs.

5.44 The 1A processor diagnostic programs provide for computer-aided diagnosis and troubleshooting of the following equipment:

- (a) Central control
- (b) Memory units:
 - (1) Program store
 - (2) Call store
- (c) File store and the Attached Processor Interface
- (d) Auxiliary data system:
 - (1) Data unit selector
 - (2) Tape unit controller
- (e) Auxiliary unit bus
- (f) Input/output system
- (g) MCC and processor peripheral interface unit
- (h) Memory buses:
 - (1) Call store bus
 - (2) Program store bus.

Diagnostic Execution Control

5.45 All requests to run a 1A processor diagnostic enter a MACP which buffers the requests and initiates them when system time and resources become available. The diagnostic programs run as MACP clients on base level. These programs execute during the segment of time allotted to MACP to perform deferred maintenance.

5.46 The general flow which is typical for processing a diagnostic request is illustrated by the following sequence of events:

- (1) A request to run a diagnostic enters MACP from fault recovery programs following an interrupt, from manual control facilities (ie, TTY, MCC), or from automatic routine exercise programs.
- (2) The MACP serves the diagnostic request and transfers to DCONMAIN.
- (3) The DCONMAIN initializes itself, and based on the unit type under diagnosis transfers to

an appropriate prediagnostic initialization routine.

(4) The prediagnostic initialization routine determines if the diagnostic can still be validly executed, prepares the unit for a diagnosis (if necessary), and returns to DCONMAIN with a go or no-go decision.

(5) If the decision is no-go, DCONMAIN does not run the diagnostic but goes to step (9) below.

(6) If the decision is go, DCONMAIN requests the paging into program store of the first set of tests (data table), any associated paged task routines, and the task dispenser.

(7) The DCONMAIN controls the execution of the diagnostic and interfaces with other programs for input/output messages, storing of test results, etc.

(8) After the diagnostic has been run, DCONMAIN activates DRPP to process any failing data, prints the diagnostic test results, and sets flags indicating the test results.

(9) The DCONMAIN then transfers to the error analysis program which records the test results and returns to DCONMAIN.

(10) The DCONMAIN then transfers to a final handler program where the decision is made whether or not to restore the unit to service. The final handler will end the MACP job or return to DCONMAIN. If a return is made to DCONMAIN, DCONMAIN completes the execution of the job by returning to MACP.

(11) The DCONMAIN is reentered by MACP to set up and initiate summary processing by DRPP. The DRPP will generate and print the pack list output message and return to DCONMAIN. DCONMAIN then returns to MACP to terminate this job.

Diagnostic Program Structure

5.47 The 1A processor diagnostic programs are data-table-driven programs controlled by DCONMAIN. These programs are composed of three distinct program types: a task dispenser, diagnostic data table, and task routines. These are all paged pro-

SECTION 231-045-005

grams and are declared to the switching assembly program as follows:

- (a) CPSECT (Control Program Section): The task dispenser section which includes the vector table and high usage task routines (these are the XXDG00 pident [per unit control programs]).
- (b) PGSECT (Program Section): The data table, (these are the XXDG01, 02, etc, pidents).
- (c) SUBROUTINES: Task routines that are not a part of a CPSECT.

5.48 The task dispenser is a section of code which reads an index word in the data table and transfers to the appropriate task routine based on that index word. Entries to the task dispenser are via a transfer vector table at the start of the task dispenser. The 1A processor task dispensers must return to DCONMAIN to end each segment within 2.5 ms after being entered by DCONMAIN.

5.49 The data table is a series of blocks, each block consisting of an index word and zero or more data words. The index provides the task dispenser the means to locate the task routine which tests the unit. The data words are used by the task routines to assemble orders for the test.

5.50 The task routines are subroutines which are transferred to by the task dispenser based on the index contained in the data table. These routines apply the tests to applicable units under diagnosis according to the data from the data table. For test evaluation and raw data storage, the routines transfer to DCONMAIN.

5.51 The general structure of the diagnostic tests consists of three levels. The basic characteristics of the three levels of this modular test structure are described as follows:

- (a) Test: A test applies a combination of inputs to a small block of circuitry, compares resulting outputs with expected no-fault results, and records a pass/fail indication.
- (b) Test Segment: A test segment is a collection of tests to be run in sequence with no real-time break.
- (c) Test Phase: A test phase is a collection of test segments that tests portions of a processor

unit. If a failure is detected, the diagnostic may be terminated if the accumulated test results can pin down a fault in the area tested. The diagnostic may also terminate on a detected test failure if further testing requires that the circuits tested up to this point be working properly.

B. ESS Switch Peripheral Diagnostic and Exercise

5.52 The peripheral diagnostic and exercise program package is designed to localize peripheral equipment faults, reconfigure associated peripheral units, and provide status information concerning the peripheral system. Diagnostic and exercise programs are provided for the following peripheral equipment:

- Network, network controller, and signal distributor
- Scanner and scanner answer bus
- Central pulse distributor and buses
- Peripheral unit bus.

Peripheral Unit Diagnostic Programs

5.53 The peripheral unit diagnostic programs are used to localize faults within the peripheral unit to a small number of circuits. The initiation of these programs is normally done by way of the FOR programs or manually from the maintenance TTY. The diagnostic request is scheduled via a maintenance control block request administered by MACR or by setting the appropriate request bit in the status word of the unit.

5.54 The diagnostic programs under MACR control can be divided into control operations and functional testing. The control operations perform the following:

- (a) Determine whether or not the requested diagnostic can be performed
- (b) Establish the appropriate system configuration for diagnosis
- (c) Initialize the unit to be diagnosed
- (d) Process the test results in a form suitable for input to the diagnostic results processing program which generates a trouble number

- (e) Generate TTY messages indicating the diagnostic results
- (f) Establish a working system configuration depending on the diagnostic results.

5.55 The peripheral equipment testing performed by the diagnostic programs consists of a fixed sequence of tests. The test results are compared with expected results or by monitoring strategic points within the unit.

Centrex and AIOD Diagnostics

5.56 Centrex diagnostic programs are designed to locate problems with centrex data links, consoles, and the AIOD equipment. These programs can send data to and accept key data from a centrex console and test whether the data is correctly sent and received. Diagnostics are run every night if the console is on night service, and may also be executed by TTY request, or if centrex data errors are detected. The AIOD diagnostic program can test an AIOD receiver and associated automatic number identification (ANI) circuits. It is normally executed only upon TTY request, but may also be entered if an AIOD fault is recognized.

Peripheral Unit Exercise Programs

5.57 The peripheral unit exercise programs are also administered by MACR on a base level, low priority basis to:

- (a) Check for faults that might otherwise go undetected
- (b) Supplement trouble detection circuits
- (c) Check trouble detection circuits
- (d) Exercise infrequently used hardware.

5.58 Exercise programs are classified in the following categories:

- (a) Scheduled automatic routine exercises which are run at prescheduled intervals of time
- (b) Demand exercises which are run at the request of another program or by maintenance personnel via the TTY.

5.59 Diagnostic test results may be processed by the dictionary trouble number program

(DOCT). The resulting trouble number can be used to index a suspect list of faulty equipment in the trouble locating manual, as described below.

DIAGNOSTIC RESULTS POST-PROCESSING SOFTWARE

A. ESS Switch Peripheral

5.60 The maintenance personnel in an ESS switch office use TLMs to interpret diagnostic printouts from the TTY. These TLMs provide an ordered list of 12-digit decimal numbers with one or more circuit packs associated with each number. After the diagnostic programs have diagnosed a faulty unit, the DOCT program reduces the raw data of the diagnostic results to a 12-digit decimal number. This number, referred to as a trouble number, appears as part of the TTY output message identifying the system component and unit. The maintenance personnel can then select the proper TLM according to the system and unit specified by the printout, and look up the trouble number. The circuit packs associated with the given trouble number may then be determined.

5.61 An additional set of phase trouble numbers will also be printed out with the overall trouble number. These numbers provide a backup, although more difficult, means of fault resolution and will only be used if the procedure designed for the overall trouble number is unsuccessful.

5.62 Each diagnostic program independently translating its results into a trouble number would be redundant; therefore, the subroutine DOCT is provided. In addition to trouble number generation, the subroutine DOCT also allows its clients, the diagnostic programs, to request some special services (eg, raw data may be printed for examination by the maintenance personnel). Special patterns of apparent failures may be interpreted as All Tests Pass and special actions taken to compute trouble numbers in cases where a unit cannot be fully tested because an associated bus is out of service.

5.63 *Number Generation:* Raw data diagnostic results may be over 5000 bits long. The reduction of this long number to a 12-digit decimal number is accomplished in two stages. The first is called bin loading; it reduces the raw data to a number that can be stored in 20 or less call store locations. The second stage is a set of three scrambling routines that operate on the bin of 20 (or less) locations to produce three

SECTION 231-045-005

4-digit decimal numbers. The set of three 4-digit numbers is the trouble number.

5.64 Phase Number Generation: The diagnostic tests are made up of groups of tests called phases. The phase number generation is done in the same manner as the number generation (but only each failing phase of data is used), and a 12-digit trouble number is produced for the phase.

5.65 Bin Loading: The bin loading routine searches the raw data for failures (1s in a field of 0s). When it finds a failure, it computes its position in the overall string of raw data. It now has a number, the address of the failure. The address of the first test of the first phase of a diagnosis is 0. In the beginning, the answer bin (which will be the input to the scrambling routine) contains all 0s. The bin loading routine examines the address of the first failure. If bit 0 of the address is 1, the routine increments the first location of the bin. If not, it increments the eleventh. If bit 1 is a 1, it increments the second bin location; if not, it increments the twelfth. The routine continues in the above manner until all ten pairs of bin locations have been properly adjusted. It then finds the next 1 in the raw data and repeats the bin loading operation. In the calculation of phase trouble numbers, an additional bin is used (PHBN). At the end of each failing phase, the bin is scrambled and the resulting 12-digit number is printed under the appropriate phase heading.

5.66 Scrambling: At the end of each phase (of phase number generations) and at the end of the last phase (for overall number generation), the bin is delivered to the scrambling routine, which performs a series of additions of successive bin locations, with a rotate operation performed on the sum after each addition. The summing is done three times, each with a different rotation of the partial sums. The three results are the final trouble number.

5.67 These processing routines are simply a means of transforming large binary diagnostic results into small decimal numbers in such a way as to minimize the probability that two different diagnostic results will yield the same 12-digit decimal number.

5.68 The DOCT also performs the following special services for clients when so requested:

- (a) Prints raw data results
- (b) Generates special trouble numbers when communication buses are out of service
- (c) Permits diagnostic restarts
- (d) Checks for special all tests pass situations
- (e) Provides extra scratch memory
- (f) Generates 10-cell trouble numbers.

B. 1A Processor

5.69 The processing of 1A processor diagnostic results is performed by the DRPP. The purpose of DRPP is to provide an on-line facility which provides a human interface for each of several methods of automatic raw data analysis, called automatic trouble locating procedures. The main output of this process is an ordered list of suspected faulty equipment locations, more commonly called the "pack list." This pack list is the first line maintenance tool available to the craftsperson to aid in the repair of faulty frames.

5.70 The DRPP, more commonly called the trouble locating procedure (TLP) program, is the collection of diagnostic results post-processing programs including both the common DRPP programs and the frame dependent interface programs.

5.71 The common DRPP programs contain those routines which are applicable to both 1A processor and application system TLP processing. Some of these routines are required to be main memory resident; whereas, those which have a low occupancy and high main memory requirement reside in file store.

5.72 Frame dependent interface programs (FDIPs) are unit dependent; ie, one exists for each applicable 1A processor and application system unit type. These all reside in file store.

5.73 All application dependent portions of the TLP program are contained in the application dependent FDIP pidents. Linkage to the application dependent FDIP program sections (PGSECTs) are handled by an application pident which receives control from the diagnostic control program (DCON) to perform all linkage-required processing.

5.74 The TLP program implements three TLP methods, each of which may apply for a specific group of unit tapes.

- The behavioral TLP

- The pattern analysis TLP
- The connectivity TLP.

Each TLP method processes the raw data in a different manner and the data base data used by each TLP is of a different format. Although the methods employed by the various TLPs may differ, the common program flow which controls the gathering of the raw data and the generation of the pack list is identical for all TLPs.

5.75 For diagnostics executing under DCON control and interfacing with the TLP program using the normal DCON interfaces, the typical generation of a pack list takes place in five general steps:

- (1) Raw data collection (DCON interface function)
- (2) Raw data analysis
- (3) Locate and monitor
- (4) Index generation
- (5) Index translation.

The generation of the pack list proceeds as described below.

5.76 The diagnostic program applies tests to the suspected faulty frame and passes each diagnostic test result along with the expected test result to DCON. The DCON program compares the actual and expected test results to determine if the test has passed or failed. The DCON program passes any failing raw data to the TLP program and prints a message on the TTY indicating the tests that have failed.

5.77 The raw data passed by DCON is analyzed by the raw data processor portion of the TLP program, and the resultant summary data is placed in file store where it is held for later processing. This summary data is referred to as a TLPFILE and the collection of TLPFILES is called the TLPQUEUE. An output message is printed on the TTY indicating the summary data that has been generated, and its disposition (ie, whether it was placed in the TLPQUEUE).

5.78 The file store queue is necessary to temporarily hold the summary data while the TLP data

base tape is positioned to generate the pack list from the summary data. Depending on the speed of the 1A processor tape drives and on the relative position of the required data on the tape, the positioning process may take several minutes. Instead of holding up other diagnostics for this length of time, the TLPFILE is created and the diagnostic ends normally.

5.79 The data file used to generate the pack list from the summary data is on auxiliary data system tape referred to as the TLP data base tape. The data on the tape is organized in groups of blocks, each group containing all the data files required to process a pack list for a particular unit. Therefore the TLP data base tape must be positioned at the start of the group associated with the particular unit whose diagnostic produced the summary data.

5.80 This positioning process, called the locate step, is controlled by a portion of the TLP program which is resident in main memory. Initiation is by the raw data processor program requesting the initial 1-second MACP entry for this program prior to returning control to DCON for termination of diagnostic processing for the current diagnostic MACP client.

5.81 As the locate step is being stepped up, a TTY message is printed to indicate that the summary data is being processed. The locate steps, operating independent of any diagnostic MACP client, positions the TLP tape to the tape file required to process the summary data. When the tape has been positioned, the locate steps initiate a TLP MACP client to start the summary processor.

5.82 The summary processor uses the summary data in the TLPFILE and data from the TLP tape to generate and print the ordered pack list. All packs for a given unit on the TLP tape are referenced by a 12-bit index, so the generation of the pack list is a 2-step process. The first step, index generation, generates the list of indexes representing the faulty packs. The second step, index translation, then translates these indexes to produce the final pack list which is output on the TTY. Once the pack list has been successfully produced, the TLPFILE is removed from the TLPQUEUE and the next TLPFILE is processed.

5.83 The craftsman may intervene in the TLP processing using TLP utility functions.

SECTION 231-045-005

Through TTY input messages, access to the TLPQUEUE is possible to check status, clear one or all of the TLPFILES from the TLPQUEUE, inhibit or allow the placement of data into the TLPQUEUE, manually generate a TLPFILE, or completely initialize the TLP program. The TLP functions also permit the craftsperson to switch the active and inhibited states of the TLP program.

TRUNK AND SERVICE CIRCUIT MAINTENANCE SOFTWARE

A. General

5.84 The overall objectives of trunk and service circuit maintenance programs are to remove

faulty units from service, pinpoint detected troubles within a faulty unit, and alert maintenance personnel of trunk and service circuit status. The trunk and service circuit maintenance programs rely on several other subsystems to aid in performing the functions previously noted. (See Fig. 37.)

B. Trouble Detection

5.85 Trunk and service circuit maintenance action may be requested under the following circumstances:

- Any difficulty in establishing a customer's call

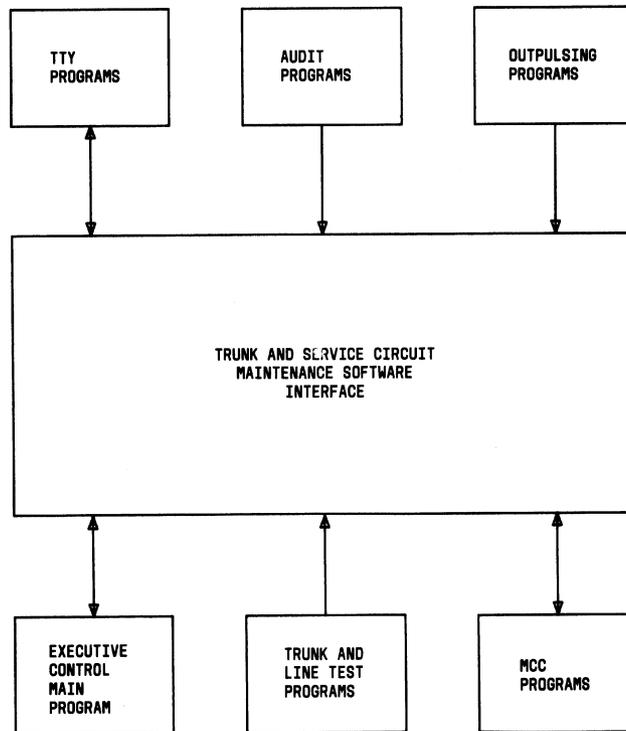


Fig. 37—Trunk and Service Circuit Maintenance Software Interface

- During automatic progression testing (APT)
- Manual request.

5.86 When problems arise during call processing activities, the trunks involved are placed on a maintenance list for testing. Some types of problems which may be detected with trunks and service circuit failures are:

- Incomplete outpulsing to a distant office
- Network continuity check failures
- Peripheral order buffer execution failures
- Transceiver time-outs
- Internal circuit check failures.

5.87 When many trunk and service circuit failures occur, the maintenance list may become full. In this case failing circuits are not tested at this time but are returned to service since a minimum number of trunks must be in service at all times.

5.88 Trunks on the maintenance list are tested using normal trunk diagnostics. If a trunk fails the test twice, a message is printed for the maintenance personnel and the trunk or service circuit is removed from service.

5.89 Besides testing these circuits when problems arise, trunk and service circuits are routinely tested on a flexible schedule during nonpeak hours. This automatic progression testing is done intermittently throughout the day, and a full cycle of testing is typically completed once a week. When faulty circuits are found, they are placed on the maintenance list and tested as previously described.

5.90 Also, when maintenance personnel deem it necessary, they may test individual circuits or groups of circuits. In this case, raw data may or may not be returned to the maintenance personnel for analysis.

C. Testing

5.91 Once a trunk or service circuit is placed on the maintenance list, it is tested via normal diagnostic procedures. The trunk and line test programs are used to request diagnosis of the circuits and place

selected circuits on various lists for further processing.

5.92 The major functions of the diagnostics are fault detection and the generation of failure data used to locate faults. Diagnostics are comprised of routines that are driven by data tables. The diagnostic programs are resident on disk in No. 1A ESS switch and are paged into main memory when executed. The diagnostics initialize the trunk and service circuits into their various states and check their reaction to certain situations.

5.93 Error analysis programs are consulted to compare the present faults with a past history of faults to obtain a data base for future analysis. Data collected by error analysis programs include the kinds of faults encountered and how often they occur.

D. Trouble Reporting

5.94 Once a trunk or service circuit has been detected as faulty and diagnostic testing has pinpointed the errors, the maintenance personnel must be informed. Teletypewriter messages are printed to allow personnel to locate detected troubles. This allows the maintenance personnel to correct the faulty circuit and return the circuit to service. In all, the trunk and service circuit maintenance programs enable the ESS switch to have a minimum number of circuits unavailable or unfit for service.

TRUNK AND LINE TEST SOFTWARE

A. General

5.95 The trunk and line test software is comprised of programs which automatically conduct a sequence of tests on trunks and lines. These tests may be activated manually from a TTY and/or automatically by the No. 1/1A processor during routine maintenance activities. Messages to maintenance personnel are printed to inform them of the tests progress and results. Besides interfacing with TTY software, the trunk and line test programs work with several software subsystems to provide reliable trunk and line maintenance. (See Fig. 38.)

5.96 The trunk and line test software include the automatic line insulation test program, incoming trunk test program, station ringer test program, line termination denied program, and through

SECTION 231-045-005

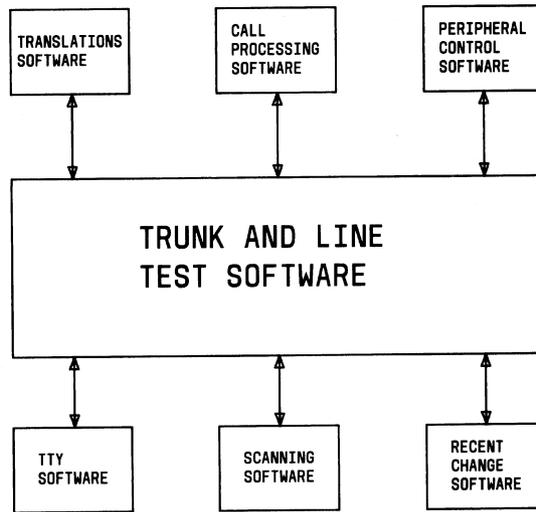


Fig. 38—Trunk and Line Test Software

balance test facility program. These programs and associated tests are discussed in the following paragraphs.

B. Automatic Line Insulation Test (ALIT)

5.97 The ALIT provides software to permit the automatic testing of line insulation values for all idle lines in an ESS switch office, except PBX lines.

5.98 Testing is initiated by requests from the MCC or may be started at a specific time and day by program control. ALIT includes instructions to enable various hardware tests to be performed with varying sensitivity. Sensitivity of the test is set to one of four ranges by adjusting the proper control digit. Other control digits are adjusted to select starting and stopping information and which tests to perform, if not all.

5.99 Normal progression testing begins with the lowest directory number in an office and proceeds through the office until the highest directory number is tested. An ALIT register is used to store the lowest directory number and the register is incremented as testing progresses.

5.100 When an idle line is found, a connection is made between the line and the ALIT test circuit. The tests are made and if successful the process is repeated on the next line. If a failure is encountered, TTY messages are printed to alert maintenance personnel and the testing is stopped.

C. Incoming Trunk Test Termination (ITTT)

5.101 Upon demand from a distant office, this program provides terminations for incoming trunk testing. An ITTT is responsible for all actions required by the far-end office to complete the test. These actions include making the appropriate connections through the office to the proper test circuit, placing the trunk under test into the proper state for testing, providing test signals over the trunk under test to the originating office, and initiating supervision for the length of the call by transferring control to proper supervisory routines.

5.102 An ITTT provides test terminations for eight incoming trunk tests:

Code test 100, 102, 103, 104, 105, 107, 108 charge test, and synchronous line test.

D. Station Ringer Test (SRTT)

5.103 The SRTT program provides the software necessary to allow maintenance personnel to perform three tests from the customer's premise. These are:

- TOUCH-TONE dialing test
- Party ground test
- Ringer test.

5.104 The SRTT program is responsible for four functions in relation to these tests. These are:

- (a) Connect the calling line to a station ringer test circuit
- (b) Test the TOUCH-TONE dialing and signal the result
- (c) Check and indicate ground identification
- (d) Connect the line to a ringing circuit to test the station.

5.105 The SRTT program is activated when the ESS switch receives a predetermined code. The SRTT connects the line to a TOUCH-TONE service test receiver via the station ringer test circuit and returns dial tone. Upon receipt of the second dial tone, the maintenance personnel dials a sequence of digits, used by SRTT to test the TOUCH-TONE service facility. The SRTT returns a double burst of high tone for a successful test and a single burst for a failure.

5.106 Any time after receipt of the second dial tone, maintenance personnel may initiate the party ground test by flashing the switchhook. The SRTT then tests the ground connections and returns coded signals to identify the type of ground connection found. After flashing, the ringer may be tested by going on-hook. SRTT will connect the line to a ringing circuit to apply ringing current. If a problem is present with the station on-hook, a coded signal is applied instead of ringing current.

E. Through Balance Test Facility (TBTF)

5.107 This test activity is initiated by a TTY request message. The TBTF checks to see if the

trunk specified by the message has been cleared for a through balance test. If it has, the state of the trunk is determined. If the trunk is busy the test is abandoned and an output message is printed. If the trunk is idle, a search is made for idle test facilities. When a test circuit becomes available, TBTF initiates a connection between the test circuitry and the trunk under test.

5.108 Once the connection has been successfully set up, the power must be removed to allow maintenance personnel to insert a test tool. If power is not removed, TBTF will not allow the test.

5.109 Power is restored when the test tool is in place and ready for testing. The TBTF places the trunk under test in the proper testing state and makes the final test connections.

5.110 After the tests are completed, maintenance personnel may request a test on the next consecutive trunk. The TBTF will take appropriate actions to initiate the test and the testing procedure will be repeated.

F. Line Termination Denied Program (PLUG)

5.111 When a subscriber's line has been denied termination, calls to the line are routed to a trouble intercept. The PLUG program processes all requests to initiate or cancel this condition.

5.112 Requests to deny line termination may only be initiated manually from the recent change TTY. A recent change message is used to change the line equipment number and directory number translation words to reflect the denied service condition and to reroute incoming calls.

5.113 When a fault is found in a subscriber line, it may be desirable to deny terminations to that line. When an idle line is found faulty, PLUG will place it on a line termination denied list. When a busy line is found faulty, PLUG will place the line on a high and wet list and indicate its condition.

5.114 Canceling the line termination denied condition can be done three ways. When a line that has been denied termination originates, PLUG takes action to remove the line from the list. (The ability to originate indicates that the fault has been cleared.) Also this condition can be cleared via a recent change message. Furthermore, when a line,

SECTION 231-045-005

which was previously placed on the high and wet list, is found to be in good condition, the permanent signal partial dial programs request PLUG to grant the line terminating ability.

DATA MAPPING SOFTWARE

A. General

5.115 This section describes the functional operation of the data mapping control and linking program (DMAPAPPL). This program is responsible for the movement and preservation of data structures containing variable data in the No. 1A ESS switch application of the 1A processor during a system update.

5.116 The major software subsystems with which the data mapping program interfaces (Fig. 39) are as follows:

- (a) File Store Administration Program: This program contains standard routines which are used by the data mapping program for all disk (file store) input/output requests.
- (b) Input/Output Control Program: This program contains standard routines which are used by data mapping to receive TTY input from maintenance personnel and to print status and error messages encountered during the update (mapping).
- (c) Maintenance Control Program: Since the data mapping program is a client of the peripheral maintenance control program, routines in maintenance control are used by data mapping for segmenting the data mapping operation into appropriate time slots.
- (d) Audit Programs: The translation system audit is used by data mapping to compress and map the temporary recent change area in core. The system audit program is used to reduce register-carried stable calls to a nonregister state.

B. Functional Description

5.117 The introduction of a new generic or new parameters into an existing ESS switch office requires, in most cases, a rearrangement of the data structures which are resident in the call store, program store, and file store. This rearrangement normally involves relocating existing structures

from their current addresses to new addresses in the same or in different stores.

5.118 The data structures which are resident in the program store, simplex call store, and file store normally contain the fixed data that is included as part of the update. These data structures are preserved directly by the system update program. No additional action is required by any other program (including the data mapping program) in the retention of these data structures during and after an update.

5.119 Call dependent data and other variable data structures contained in the duplicated call store which must also be preserved during an update include:

- Path memory information needed to maintain an existing call in the talking state
- Temporary recent change data
- Trunk maintenance lists.

5.120 The core copy of the temporary recent change area is mapped by a subroutine in the translation system audit program during core-to-core mapping. The disk backup copy is updated after the memory reinitialization phase is run following the update. Data structures to be mapped are:

- Temporary Recent Changes
- Path Memory for Lines
- Trunk-to-Trunk Memory
- Path Memory for Trunks
- Traffic Scratch
- General Purpose Traffic Accumulators
- Day, Month, and Year Counter
- Traffic Counters
- Plant Measurements Daily Counter
- Input/Output Status Tables
- Central Pulse Distributor Status Tables
- Peripheral Unit Bus Status Tables

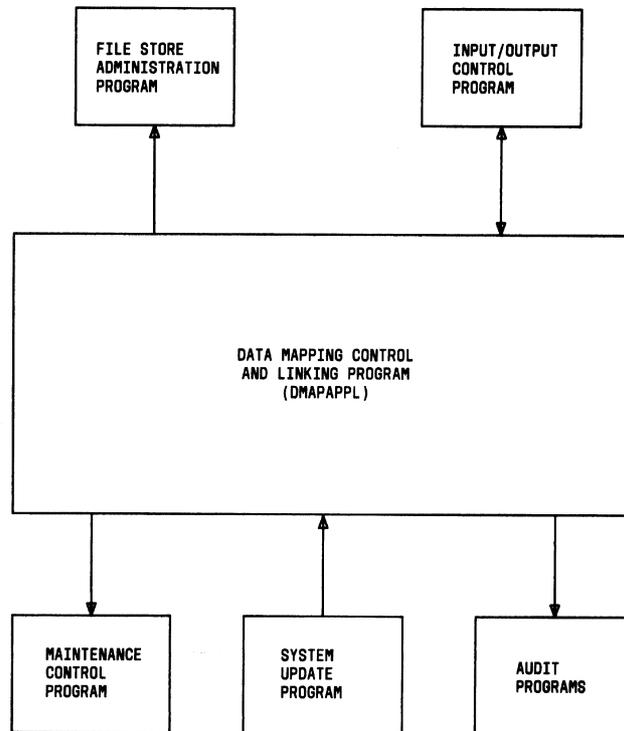


Fig. 39—Data Mapping Interfaces With Other Subsystems

- Scanner Answer Bus Status Tables
 - Central Pulse Distributor Bus Status Table
 - Pointers to the Recent Change Rollback Area.
- 5.121 These data cannot be saved by a common program, such as the system update program, since the identity of these data structures depends largely on the following:
- (a) The application of the 1A processor
 - (b) The nature of the generic programs involved in the update
 - (c) The nature of the memory recovery phase run after the update.
- 5.122 In the No. 1A ESS switch, scheduled changes to the generic program or the office parameters are introduced to an existing office by means of a full or a partial update. A full update implies that all data in either or both program stores and the parameter area of unduplicated call store are being replaced with new system data. A partial update will result in only selected data in the parameter or generic areas being replaced with new data.
- 5.123 The data mapping program is concerned with that segment of the system update pro-

SECTION 231-045-005

cess which involves the preservation of transient call store data.

Order of System Update

5.124 The steps followed by the system update and data mapping programs during a full update are as follows:

- (1) Copy the tape(s) containing the new generic and/or office parameter data to a file store community that was manually configured for update.
- (2) Verify that all call store K-codes to be duplicated by the new copy of the parameter data are in fact duplicated.
- (3) Map file store data with a new copy of the data mapping program.
- (4) Map transient call store data.
- (5) Run a phase 4 to initiate the periphery.
- (6) Begin call processing with the new generic and/or parameter data.

5.125 The above order of events involved in a system update minimizes the complexity of recovery from an interrupt during the update process. This procedure also has the advantage of having no loss in call processing if a mapping failure occurs prior to Step 4. A failure after Step 4 may lead to the loss of both transient and talking state calls.

Update Procedure

5.126 The process for a generic update (Fig. 40) begins with maintenance personnel securing an available tape unit controller for the update function. The tape unit controller should be set for read only by inputting the appropriate TTY message.

5.127 A rover program store is then seized and configured for use as a buffer area for data mapping and system update by the appropriate input message. (Since maintaining a dedicated area of core for infrequent use is uneconomical, a rover program store is utilized.)

5.128 A file store community is next selected for update by operating the appropriate file

store key on the MCC common panel. This action will prevent the use of that file store community by programs other than those involved in the update process.

5.129 The contents of the update tape is then copied into the selected file store community by a TTY input message. At this point, maintenance personnel can request, via the TTY, to bypass the data mapping segment of the update (system default is to perform data mapping).

5.130 Following the integration of the file store with the data from the update tape, the system update program will print a message indicating that the file store tape is completed. A new tape or the update of core should be started.

5.131 When all update tapes have been inputted to the file store community, the system update program prepares the following new system maps:

- (a) Core to disk map
- (b) Identification (id) tag to file store map.

5.132 The old and new system maps are checked for ID overlaps and for disk descriptor block consistency. A failure in any of these checks will result in a request to either provide additional tape input or to abort the update. A successful pass in each of the above checks and the absence of a TTY request to inhibit data mapping results in control being transferred to the disk mapping program to perform file store mapping and to build tables for core mapping.

5.133 The old control routine in the disk mapping program locates, from the system maps prepared by the system update program, the new copy of the disk mapping program and the new copy of the parameter data assembler (PDA) on the updated file store. The new disk mapping program and PDA are then pumped into the rover program store that has been configured for use as a scratch area.

5.134 After the pump of the new data mapping program and PDA into the rover program store, the old data mapping program then transfers program control to the new data mapping program to map the necessary file store structures. File store mapping consists of the copying of the required data from their locations on the nonupdated file store to

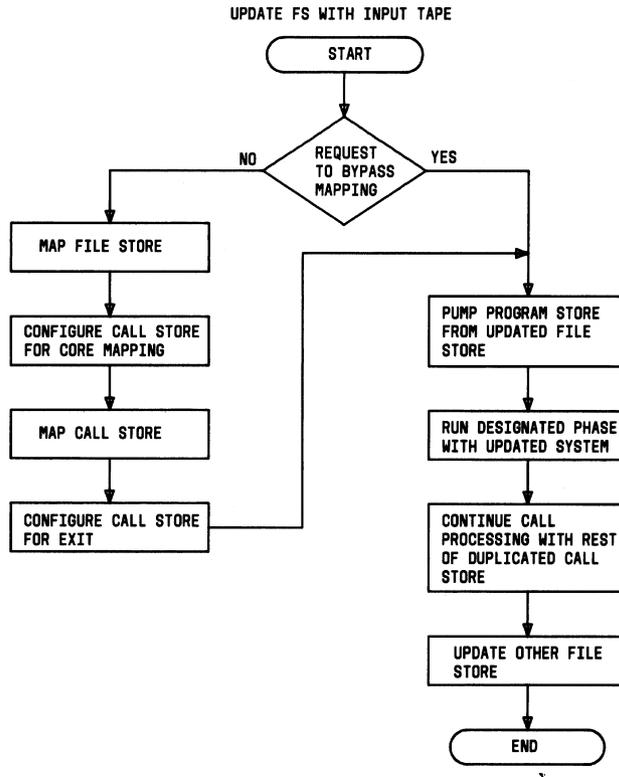


Fig. 40—Functional Flowchart of Update Procedure With Emphasis on Mapping

a buffer area in the rover program store. The data is then modified, as required, and copied to the updated file store.

5.135 The data mapping program returns program control to the system update program upon completing file store mapping which then sets up the system for core mapping by splitting the duplicated call stores into two halves.

Configuration of Duplicated Call Store

5.136 The duplicated call stores on the active bus are maintained in the normal mode while those on the standby bus are reconfigured to the ac-

tive bus, set in the maintenance mode, and initialized to zero.

5.137 The set of duplicated call stores that are now in the normal mode can be accessed by normal reads and writes while those in the maintenance mode can only be accessed by maintenance orders. This will enable the data mapping program to address its reads and writes to particular member numbers in the duplicated call store community.

5.138 H-level and J-level interrupts are now inhibited in order to prevent call processing programs from updating transient data structures concurrently with the mapping of the same struc-

SECTION 231-045-005

tures. G-level interrupts are also inhibited to prevent routine maintenance programs and the generic utility program from either updating transient data concurrent with mapping or the triggering of recovery actions which may also abort the update.

Actions by Old Data Mapping Program

5.139 The system update program will transfer program control to the old data mapping program upon the completion of system preparation. The old data mapping program when entered at this time will perform the following functions:

- (a) Unpest J-level interrupts and set the J-level modify instruction address so that a count of J-level interrupts may be made to ensure that the B-level timer may be reset before it expires. The data mapping program counts a predetermined number of J-level interrupts before it transfers control to the system update program to reset the long timer. If the data mapping program fails to transfer to the system update program in time to reset the timer, the timer will expire, triggering an automatic processor configuration which will abort the update.
- (b) Transfer program control to the system audit programs to reduce register-carried stable calls to a nonregister state in order that they may be saved by mapping path memory. These types of calls include hotel/motel, temporary transfers, and simulated facilities.

Transfer to New Data Mapping Program

5.140 Transfer of program control is next made to the new data mapping program in the rover program store to map transient data. The old and the new data mapping programs are then scanned to determine which structures to map and which structures to skip. If mapping is required, a transfer is made to the appropriate mapping algorithm for the actual mapping.

5.141 Program control is passed at the completion of core mapping from the new data mapping program to the system update program to pump the program stores, translation stores, and call store 0. The original standby stores in the maintenance mode which now contain the mapped data are made active and the originally active stores are taken out of service before the pump. The phase which is specified on

the tape header of the update tape is next run to complete the recovery. At some time after this phase is run, the call store fault recognition programs are called on a time-shared basis to reconfigure a duplicated call store community.

SYSTEM PERFORMANCE SOFTWARE

A. General

5.142 The system performance software consists of program operations designed to indicate the ability of the system to perform its call processing functions.

B. System Performance Software Interface

5.143 The interface between the system performance software and other ESS switch software systems is illustrated in Fig. 41.

5.144 The system operational control software provides scheduled entries to the system performance software. The system performance indicator program (SYPI) is entered every 10 seconds to collect data and perform tests on the displayed system performance indicator located on the No. 1A system status panel. The dial tone speed test program (DTST) is entered every 4 seconds only if the DTST feature has been activated either manually or automatically. An entry is made every 4 seconds to the RADR program to test the delay in receiver attachment.

5.145 All of the system performance programs provide interface with the TTY software for manual interfacing with the TTY.

C. System Performance Software Function

System Performance Indicator Program (SYPI)

5.146 The system performance indicators are located on the No. 1A system status panel. The purpose of these is to visually alert the craftsman that the system processing capabilities are decreasing. The information provided by SYPI allows the craftsman to take corrective measures before the system performance deteriorates to a lower level.

5.147 The status of the following resources are displayed by TEST STATUS lamps:

- Call Registers
- Hoppers

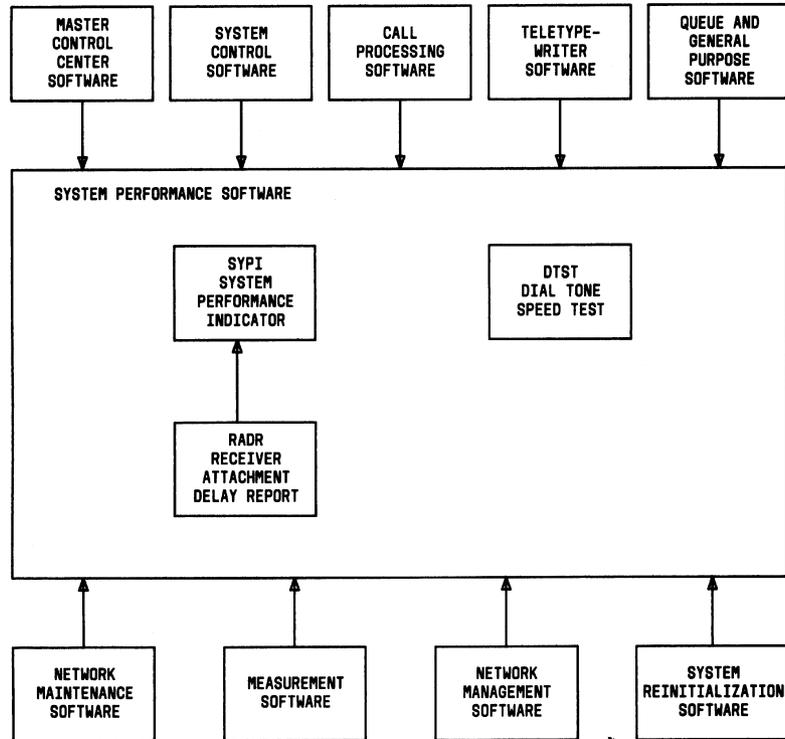


Fig. 41—System Performance Software Interface

- Queues
- Service Circuits
- Line and Network
- Data Transfer
- Abrupt Traffic Change.

5.148 The SYPI program applies a statistical test to call registers based on holding time. The average holding time is defined as the ratio of the sum of the active elements to the number of seizures. A moving average and a moving variance are com-

puted based on the average holding time. If the sample is very small, no statistical analysis is performed. A safety factor of 4 is normally applied if the sample is sufficiently large. If the value falls beyond the safety value, the test is considered a failure and the lamp is lighted amber. Also, whenever there are requests for the facility but no seizure is made, the lamp will be lighted amber.

5.149 The lines and networks are checked for excessive failures by SYPI. The check periodically computes the ratio of the weighted sum of the faults constituting the particular failure type to the total traffic per network observed during the interval. If this ratio exceeds a predefined threshold, then

SECTION 231-045-005

the TEST STATUS—LINE AND NETWORK lamp is lighted amber.

5.150 Statistical tests are applied periodically in order to alert the craftsperson via the panel status lamps whenever an excessive number of failures occur.

5.151 The statistical tests for line and network are done once every minute. Based upon this criteria, the TEST STATUS—LINE AND NETWORK lamp is lighted amber whenever an excessive number of failures, in comparison to the traffic, occurs in the system. The lamp is lighted green only if the statistical checks are successful.

5.152 The TEST STATUS—DATA TRANSFER lamp reflects results of the test on the disk and TTY equipment. The tests are executed once every 10 seconds by SYPI.

5.153 A statistical test is performed to verify the availability of the OMRs. A failure is recorded if there is either an excessive number of OMR seizure failures or a lack of idle OMRs. In either case, SYPI requests the appropriate TTY audit (no more than once every 5 minutes).

5.154 The TEST STATUS—DATA TRANSFER lamp is lighted amber if any of the preceding tests fail. The lamp is lighted green only if all tests pass continuously a fixed number of times.

5.155 Every 10 seconds SYPI applies a statistical test on certain traffic conditions. The test is similar to that applied to call registers and service circuits, differing in precision maintained in calculations. If the value falls beyond the given tolerance interval, the TEST STATUS—ABRUPT TRAFFIC CHANGE lamp is lighted amber.

5.156 Light-emitting diodes (LEDs) on the panel display system activity associated with:

- Originating calls
- Incoming calls
- Intraoffice calls
- Tandem calls
- Processor occupancy
- Manual selection (miscellaneous).

5.157 In general, the number of LEDs lighted green indicate the value of the item being displayed as a percent of the threshold (100 percent) value. If the value of the item falls below 10 percent of the threshold value, the column identification lamp at the bottom of the bar graph is lighted white to indicate a change in scale. The change implies that the percent reading for that bar graph must be multiplied by 0.1 to obtain the true reading. For these low values, each LED represents increments of 1 percent. If, on the other hand, the item being displayed exceeds the threshold value, the LED at the top of the bar graph is lighted red in addition to the 10 green lighted LEDs.

5.158 Bar graphs, ORIG. CALLS, INCOMING CALLS, INTRA OFF CALLS, and TANDEM CALLS are updated only once each minute. The PROCESSOR OCCUPANCY bar graph is updated once every 10 seconds.

5.159 The PROCESSOR OCCUPANCY bar graph value is calculated by system operational control software once every 10 seconds. This value is the percent of real time spent by the processor on nonrelinquishable and call processing functions. The bar graph reflects the time spent on accomplishing the basic objective of the system, which is to process calls.

5.160 The MANUAL SELECTION bar graph is normally not lighted.

Dial Tone Speed Test Program (DTST)

5.161 The purpose of the DTST program is to provide measurement of dial tone service. It applies a test to measure the interval of time separating customer off-hook from receipt of dial tone. If dial tone is furnished within 3 seconds, a success is recorded. If the 3-second test was a failure and at the end of a 11-second period dial tone has not been furnished, the test is scored as an extended failure.

Note: The dial tone speed test function is activated by a TTY input message.

Receiver Attachment Delay Report (RADR) Program

5.162 The RADR program generates, maintains, and administers incoming test calls designed to measure the amount of delay incoming calls are experiencing in getting a connection to a receiver.

Through-the-Wall Motion Detection Device

Overview

They have names like Crunch, Goldstein, Cheshire, Prophet, Newby, Rambam, and Jello. They meet once every two years in New York City under the guise of putting on a "hacker" conference. Their method of swindle is very simple. Perpetuate a continuous string of "I'm a victim!" and other "boogie man" stories in the media, which they can control, then sit back and watch the money flow in. Only these particular "hackers" want even more than just money. They are also after a steady supply of young boys to lust over.

And it will be your job to capture these predators! Thanks to budget cuts (illegals need health care too), your own safety is no longer a considered factor. So, how can *you* ensure your own safety when tracking or arresting sick criminals like these? Well, a useful device you can build is called a "through-the-wall motion detector." This is a device which can detect any human movements in a particular room or closet before you enter or search. And all it takes to build is a fairly well-stocked "junk" pile and a handful of parts from Radio Shack.

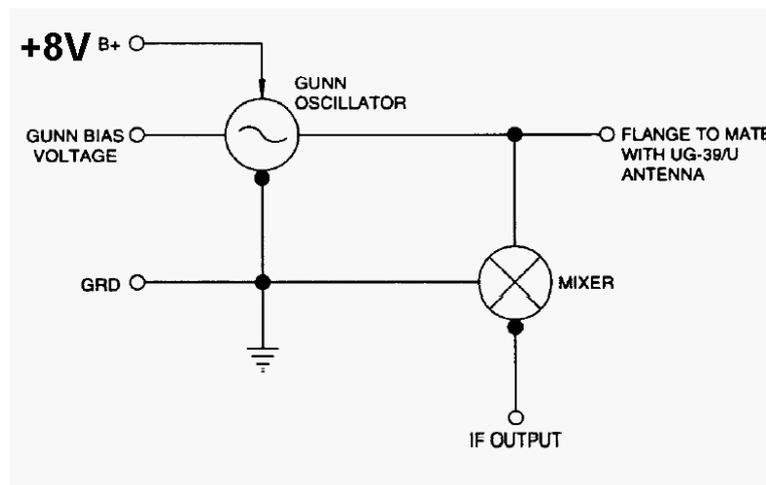
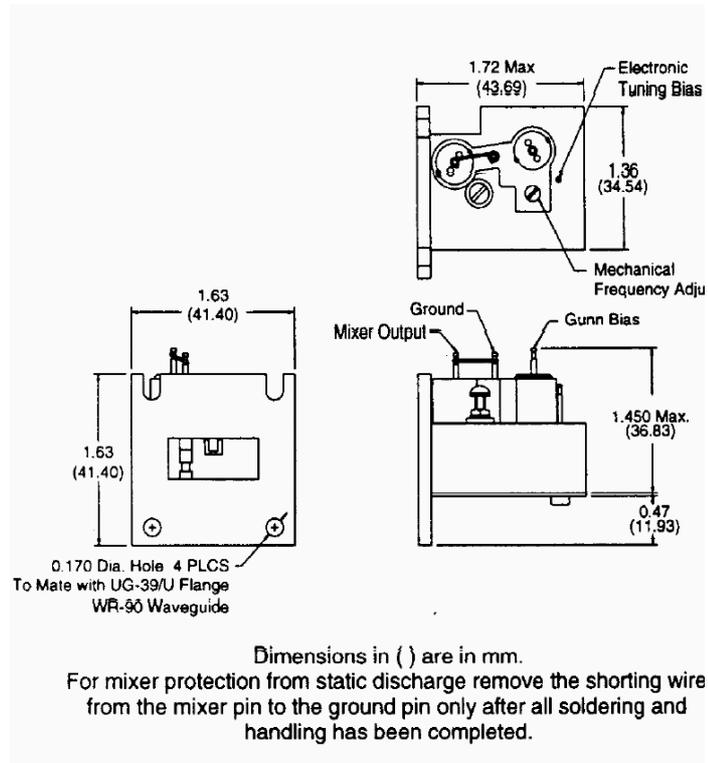
Construction Notes

The idea is to salvage a 10.5 GHz Gunn oscillator/transceiver (Gunnplexer) from an old automatic door opener or police radar gun. You'll then add a high-gain pre-amplifier and comparator circuit after the Gunnplexer's internal mixer output to detect any motion via the Doppler effect. The Gunnplexer is used as a RF "illumination" device and the mixer output is a low-frequency signal equal to the amount of Doppler shift in the target's motion. Since the Gunnplexer samples both the *transmitted* and *received* 10.5 GHz signal, the received (reflected) signal of *any* target in motion will impart a slight frequency shift via the Doppler effect. This output "beat" frequency can also determine the speed of the target. If you have ever gotten a speeding ticket, well, now you know how... This particular device is capable of detecting motion from anything like a person walking to, in theory, just a simple heartbeat.

The post-mixer pre-amplifier stage will be made around the commonly available LM358. After amplification, the signal will be sent to the comparator stage, made from the second op-amp in the LM358. This is also where you can set the threshold for the motion detection trigger. A sample of the pre-amplified signal will also be sent to a LM386 audio power amplifier. This will allow the use of headphones to help weed out any "false positives" from electrical interference. Induced electrical interference will sound like a loud 60 or 120 Hz buzzing tone, with fluorescent lights being the biggest problem. The "high" output from the comparator stage toggles a transistor, which, in turn, triggers a 555-timer wired for monostable output. The output pulse from the 555-timer then controls a LED or other alerting device. When any motion is detected which is greater than the comparator's threshold setting, the 555-timer will trigger, illuminating the LED for a second or so.

The heart of this device is the 10.5 GHz Gunnplexer. New Gunnplexers and their associated datasheets can be fairly difficult to track down. The following block diagrams should cover the pin-outs on most of the basic models from Alpha and M/A-Com. The Gunn bias voltage is usually around +8 VDC (200 mA) and the varactor frequency tuning voltage (if present) is usually between +1 and +20 volts. A new Gunnplexer might have a static shorting wire from the mixer output to ground. You'll need to replace this with a 1,000 or 2,200 ohm resistor. Mechanically tuning the Gunnplexer's cavity can usually swing the output frequency from 9.5 to 11 GHz. Varactor tuning can be used to electrically swing the output 100 MHz or so. This is useful for modulating the output RF signal or for adjusting the frequency to avoid interference from other devices in the same band.

M/A-Com Gunnplexer Block Diagrams & Notes



Doppler Frequency Shift Equation

$$F_d = (2 * F_o) / c$$

F_d = Doppler Shift (Hertz)

F_o = Original Transmit Frequency (Hertz)

c = Speed of Light (299,792,458 meters per second)

Using a 10.5 GHz Gunnplexer, the on-axis, two-way Doppler shift would be 70 Hz each meter per second (31.3 Hz each mile per hour) of the target's speed.

The Doppler shift frequency is same whether the target is moving *towards* or *away* from the Gunnplexer.

Pictures

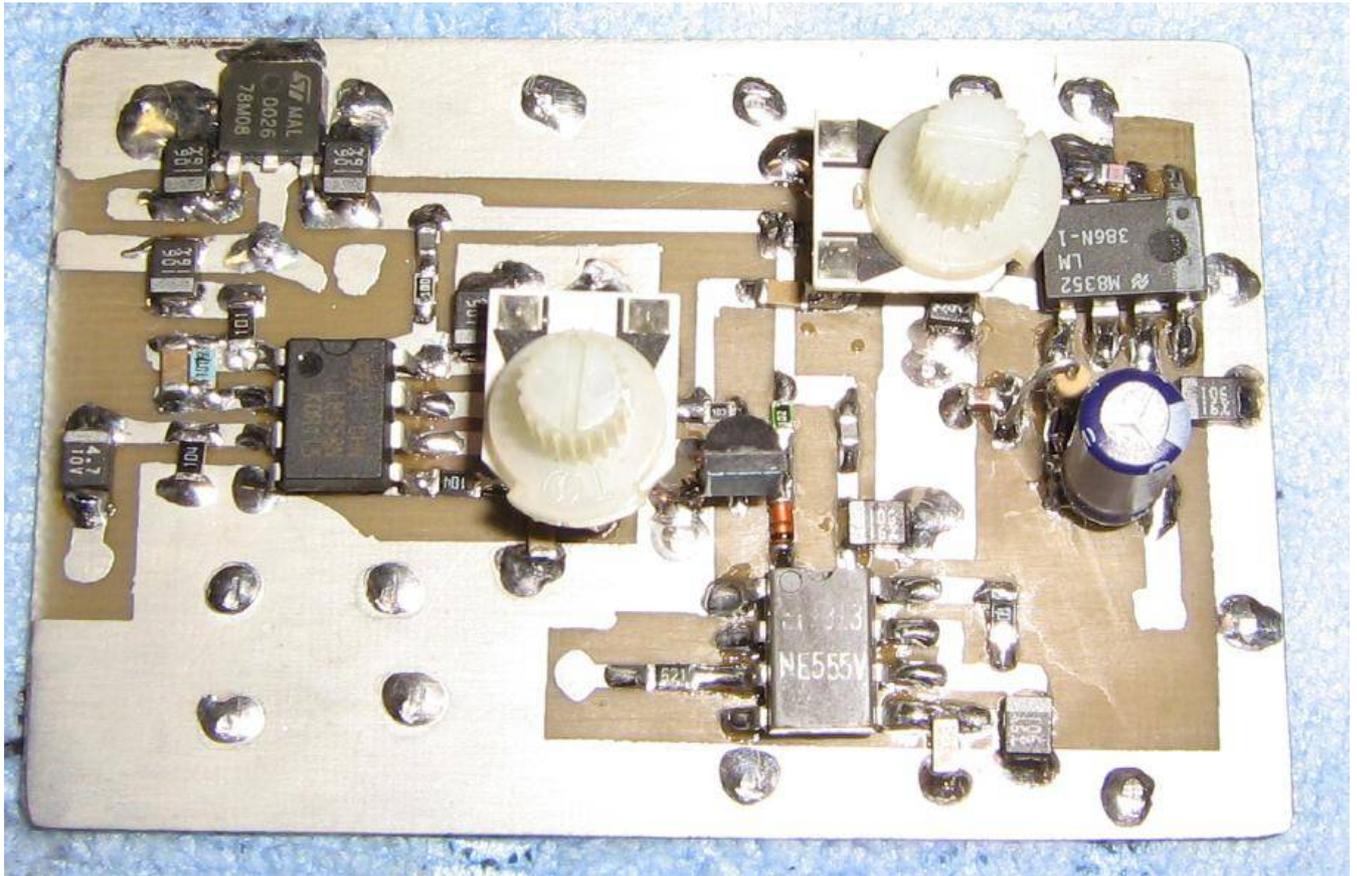


Overview of the major parts used.

An "Aluminum Handle Single Suction Cup" from Harbor Freight Tools (#92825) will be used to mount the Gunnplexer and its associated electronics. This will allow the motion detection device to be mounted to a smooth surface without the worry of any vibrations causing false triggers.

The Gunnplexer is a M/A-Com MA87728 and the horn is from an old automatic door opener.

The small piece of aluminum bar stock on the right side will attach to the suction cup's handle to hold the Gunnplexer.



Pre-amplifier and audio amplifier circuit board.

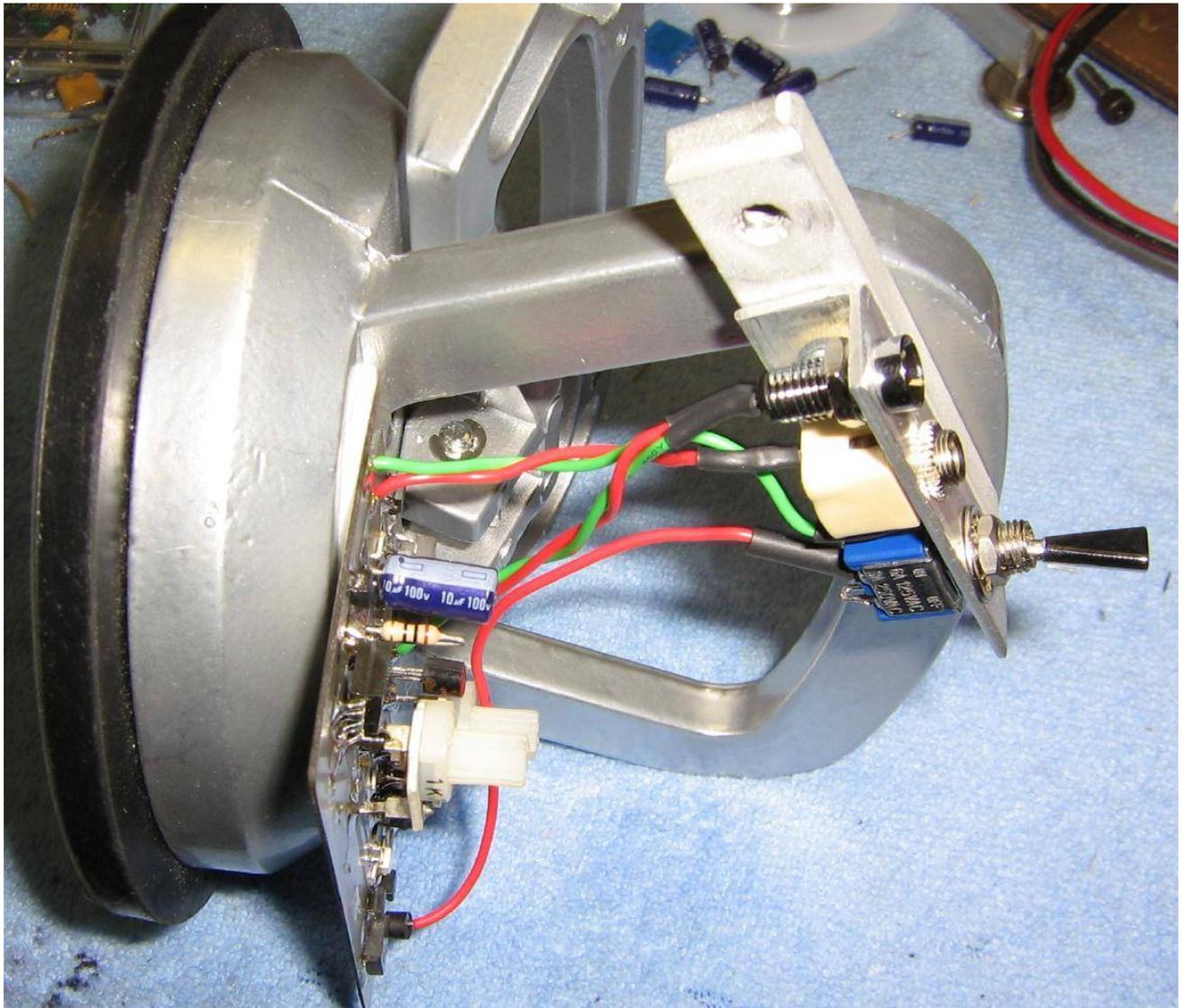
Construction of the board is not too critical, but you'll want to have a large ground plane and proper EMI filtering. Some components and traces in the above photo are different from the schematic due to various changes. The schematic will be accurate.

The LM358 is the 8-pin IC on the left, the comparator threshold setting potentiometer is in the middle. The 555-timer is the 8-pin IC on the bottom-right, and the LM386 is on the upper-right with its input volume potentiometer next to it. A 78M08 voltage regulator is in the upper-left. The voltage regulator is for both the pre-amplifier circuit board and the Gunnplexer bias.

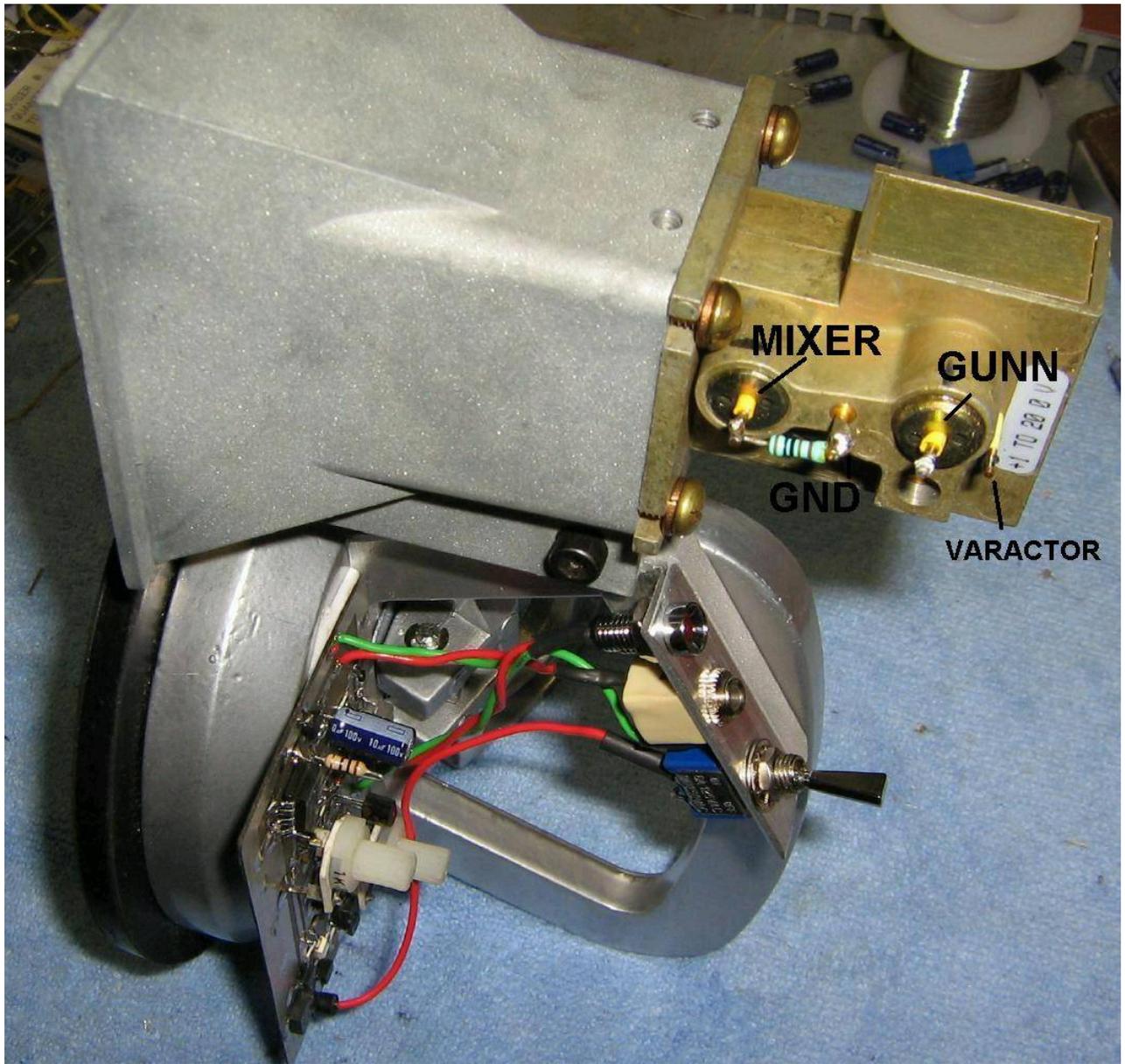
The LM358 pre-amplifier is set to only amplify below 1,000 Hz or so. You can adjust the op-amp's feedback and gain setting components to reduce false triggers or electrical interference. Additional circuit improvements would include using a higher quality low-noise op-amp instead of the LM358 and 60 or 120 Hz notch filters after the pre-amplifier.

It is even possible to get target range information if you modulate the Gunnplexer's varactor with a triangle wave, then compare the phase relationship with the received reflected signal.

And yes, modulating the varactor with a 1,720 Hz sine wave will make a X-band police radar gun read "55 MPH."

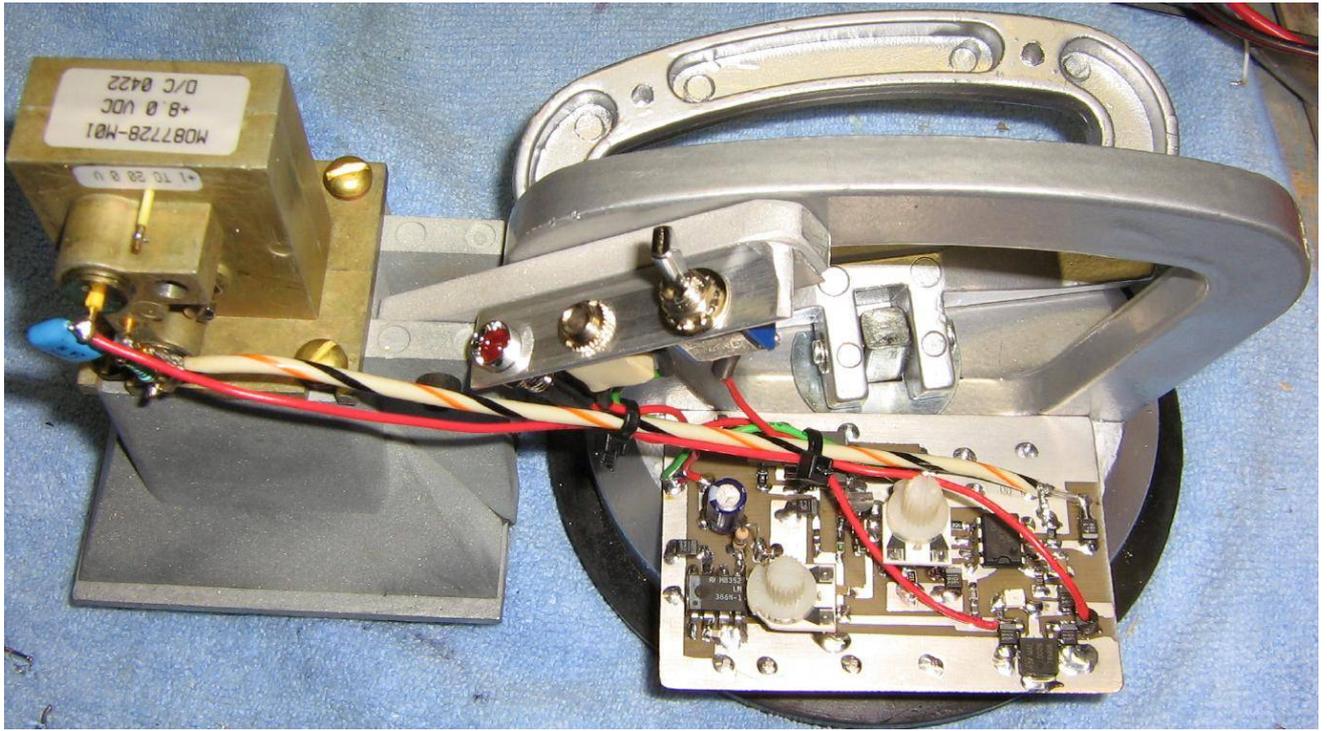


A little piece of aluminum angle stock is used to hold the motion indicator LED, a 1/8" jack for the headphones, and the power switch.



Attachment of the Gunnplexer and horn assembly to the suction cup handle is via a single #8 screw. The entire Gunnplexer assembly can be tilted up or down slightly by loosening the attachment screw.

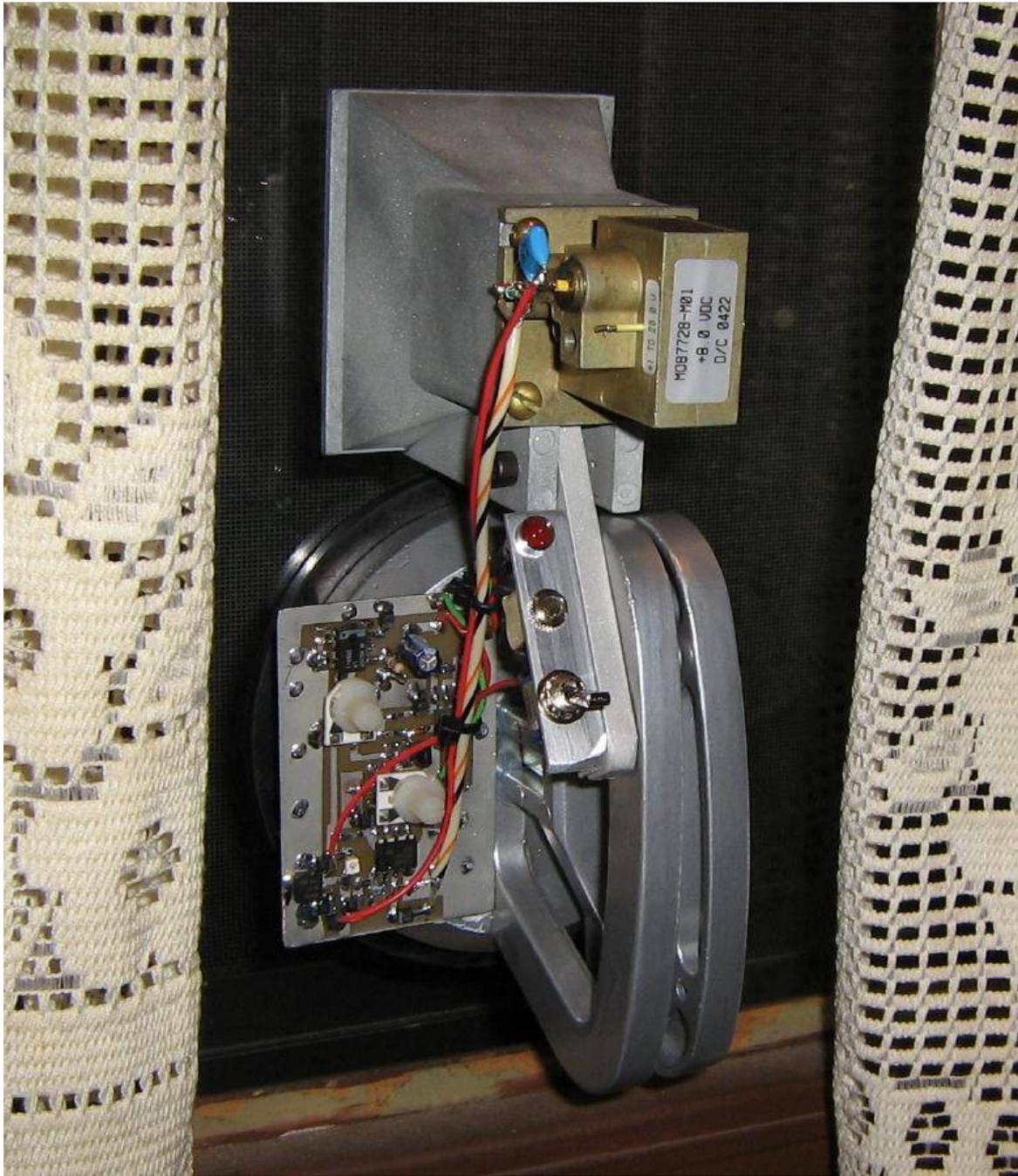
Note the 1,000 ohm resistor from the Gunnplexer's mixer output to ground. This provides a proper DC return for the mixer diode bias.



Completed through-the-wall motion detection device.

The Gunnplexer's mixer output is via shielded wire and the +8 VDC Gunn diode bias is straight off the 78M08 regulator's output. A 0.1 μF capacitor is across the Gunn diode bias pin to ground.

Since this was mostly a test setup, nothing was properly shielded and there is no room for batteries. The device was powered from an external +12 VDC battery pack.



Through-the-wall motion detection device attached to a door window.

Since the device is very sensitive to any motion, you'll want to attach it to something solid so it doesn't move. You can moisten the rubber of the suction cup for better attachment to some wood surfaces. It should be noted that this device *does not* work through metal doors and brick walls can severely attenuate the signal.

The unit also takes a few seconds to "warm up" after power is applied. You'll also want to experiment with the threshold settings ahead of time to get a feel for the range of this device.

A video showing the testing of this device is available at:
<http://www.youtube.com/watch?v=RyyXbm7Wsd8>

DMS-100 Test Cross Connections Table (TSTXCON)

Table Name

Test Cross Connections

Functional Description of Table TSTXCON

Table TSTXCON acts as a look-up table to define the relationship between Maintenance and Administration Position (MAP) terminal jacks and headsets, and External Trunk Numbers (EXTRKNM). Table TSTXCON uses the terminal descriptions provided in table TERMDEV, using field TERMDDES as the key. The EXTRKNMs are selected from the datafill in table TRKMEM. Jack, headset, and EXTRKNM connections are made from the MANUAL sublevel of the Trunk Test Position (TTP) level of the MAP.

Datafill Sequence & Size

The following tables must be datafilled before table TSTXCON:

- TERMDEV (Terminal Device)
- TRKMEM (Trunk Member)

Table size and memory size are defined by the maximum number of MAP devices entered in table TERMDEV.

Datafill

The following table describes datafill for table TSTXCON:

Table TSTXCON Field Descriptions

Field	Subfield	Entry	Explanation and Action
TERMDDES		Alphanumeric (up to 8 characters)	<i>Terminal Designation</i> Enter the MAP terminal device name. The entry must be a valid device name datafilled in field TERMDDES in table TERMDEV. This is the key field.
JKHSETAB		See Subfields	<i>Jack and Headset A & B Trunk Combinations</i> This field consists of subfields TRKNAME and EXTRKNM. A vector of up to 24 multiples of jack (or headset) and external trunk number combinations can be datafilled. If less than 24 multiples are required, end the list with a "\$".
	TRKNAME	JACK or HSET	<i>Trunk Name</i> Enter the name of the trunk test connection. Entries are a vector consisting of the type of test connection (JACK or HSET) followed by the external trunk number. Datafill subfield EXTRKNM to complete each combination.

EXTRKNM 1 to 9,999

External Trunk Number
Enter the external trunk number associated with the trunk name datafilled in subfield TRKNAME to complete each combination. Entries outside this range are invalid.

-End-

Datafill Example

The following example MAP display shows sample datafill for table TSTXCON:

TERMDES	JKHSETAB
MAP	(HSET 4) \$

Two-Tone Pager Decoder Using Multimon

Overview

Alot of rural volunteer fire departments still rely on the Motorola two-tone sequential paging system and analog Motorola Minitor pagers for dispatching their crews to a fire scene. The standard "Motorola Quick Call 2" paging protocol consists of playing two separate audio tones, the "A" and "B" tone. The "A" tone is played first for one second, then the "B" tone for three seconds. Both of these tones are transmitted on the fire dispatch frequency (VHF usually) which the pager is tuned to. Inside the older Minitor pagers, a mechanical reed is used to filter and decode each of the proper tones. While this may sound primitive, it is actually very reliable. A modern tweak to this type of paging system would be for the fire dispatch page to also be sent to your computer or cellular phone via text or email message. That is what this project will attempt to cover, with the pager tone decoding being done in software instead of having to tie up an additional pager.



Actual Two-Tone Page at 398.1 and 912.0 Hz

For the tone decoding software, we will be using a slightly modified version of Thomas Sailer's `multimon` Linux radio transmission decoder, which is available at: www.baycom.org/~tom/ham/linux/multimon.html. This program uses a standard PC soundcard to acquire the signal from a radio scanner (or equivalent) and the decoding is done completely in software.

The modification will consist of replacing the DTMF tone values in `demod_dtmf.c` with tone values which are equal to that of the two-tone pager we wish to receive.

For example, in `demod_dtmf.c`, we see this matrix:

DTMF Frequencies

	1209	1336	1477	1633	
697	1	2	3	A	
770	4	5	6	B	
852	7	8	9	C	
941	*	0	#	D	

These are the standard 16-character DTMF tones we've known to love. The stock code array looks like this:

```
static const unsigned int dtmf_phinc[8] = {
    PHINC(1209), PHINC(1336), PHINC(1477), PHINC(1633),
    PHINC(697), PHINC(770), PHINC(852), PHINC(941)
};
```

As you can see, to decode a DTMF "1," the decoding routine looks for a simultaneous dual-tone value of 1209 Hz and 697 Hz in the incoming audio. Since two-tone pagers use *single* tones, and *not* DTMF tones, all we really need to do is replace both the "column" and "row" decoding array with the same tone value. Example:

```
static const unsigned int dtmf_phinc[8] = {
    PHINC(398), PHINC(912), PHINC(1477), PHINC(1633),
    PHINC(398), PHINC(912), PHINC(852), PHINC(941)
};
```

To decode a two-tone pager which uses a 398.1 Hz and 912.0 Hz tone, replace the "column" and "row" decoding structure with "398" and "912." This then maps a DTMF "1" to 398 Hz and the DTMF "5" to 912 Hz. Since the second tone is played for three seconds, you'll get three individual decodes of that tone.

Operation

Connect the audio output from a radio scanner or two-way radio tuned to the pager's dispatch frequency to the "line in" (`/dev/audio`) on the soundcard and wait for the proper two-tone page. An isolation transformer can be used to reduce any "hum" interference from ground loops. This is what you should see in `multimon`:

```
bash# ./multimon -a DTMF
multimod (C) 1996/1997 by Tom Sailer HB9JNX/AE4WA
available demodulators: POCSAG512 POCSAG1200 POCSAG2400 AFSK1200 AFSK2400 AFSK2400_2 HAPN4800
FSK9600 DTMF ZVEI SCOPE
Enabled demodulators: DTMF
DTMF: 1
DTMF: 5
DTMF: 5
DTMF: 5
```

When combined with an additional script or program which parses the output from `multimon`, you can then trigger a text or email message, or any other form of notification to be sent out.

A potential Perl script to trigger an external command would look something like the following code. This is just an example, and there are probably better methods (and better coders) than this.

```

-----CUT-----
#!/usr/bin/perl

# Path to multimon
$mm = "/usr/local/bin/multimon -q -a DTMF";

# Tone/DTMF string to trigger on
$on_str = "1555";

# Command or script to execute
$on_cmd = "echo Fire Fire Fire | mail -s Fire you@cellphone.com";

select STDOUT;
$| = 1;
$i = 0;

sub System {
if ((0xffff & system $args) != 0 ) {
    print STDERR "error: $!\n";
    exit 1;
}
}

open M, "$mm |" || die "Can't open $mm: $!\n";

while (<M>) {

    ($a, $b) = split ':';
    $b =~ tr/0-9*#ABCD//csd; # Allow 0-9 * # A B C D
    $ans .= $b;
    $i++;

    if ($i == (length $on_str)) {

        if ($ans eq $on_str) {
            System($args = $on_cmd);
            undef $ans;
            $i = 0;
        }

        else {
            undef $ans;
            $i = 0;
        }
    }
}
}
-----CUT-----

```

Below is patch to `multimon` which adds a "quiet output" option to the DTMF decoding and also flushes `stdout` for better reliability when used in this application.

To apply the patch;

1. `tar xvzf multimon.tar.gz`
2. `patch -p0 < multimon.patch`

```
-----CUT-----
diff -ur multimon.orig/demod_dtmf.c multimon/demod_dtmf.c
--- multimon.orig/demod_dtmf.c  Mon Dec  8 10:56:06 1997
+++ multimon/demod_dtmf.c      Fri Sep 10 03:27:30 1999
@@ -25,6 +25,7 @@
 #include "filter.h"
 #include <math.h>
 #include <string.h>
+#include <stdio.h>

/* ----- */

@@ -137,6 +138,7 @@
         i = process_block(s);
         if (i != s->ll.dtmf.lastch && i >= 0)
             verbprintf(0, "DTMF: %c\n", dtmf_transl[i]);
+            fflush(stdout);
         s->ll.dtmf.lastch = i;
     }
}

diff -ur multimon.orig/gen.c multimon/gen.c
--- multimon.orig/gen.c  Mon Dec  8 10:56:06 1997
+++ multimon/gen.c      Fri Sep 10 03:38:12 1999
@@ -367,7 +367,7 @@
 " -s <freq> : encode sine\n"
 " -p <text> : encode hdlc packet\n";

-void main(int argc, char *argv[])
+int main(int argc, char *argv[])
{
    int c;
    int errflg = 0;
diff -ur multimon.orig/unixinput.c multimon/unixinput.c
--- multimon.orig/unixinput.c  Mon Dec  8 10:56:06 1997
+++ multimon/unixinput.c      Fri Sep 10 03:41:43 1999
@@ -370,12 +370,14 @@
 "(C) 1996 by Thomas Sailer HB9JNX/AE4WA\n"
 " -t <type> : input file type (any other type than raw requires sox)\n"
 " -a <demod> : add demodulator\n"
-" -s <demod> : subtract demodulator\n";
+" -s <demod> : subtract demodulator\n"
+" -q          : quiet output messages\n";

int main(int argc, char *argv[])
{
    int c;
    int errflg = 0;
+    int quiet = 0;
    int i;
    char **itype;
    int mask_first = 1;
@@ -383,16 +385,15 @@
 unsigned int overlap = 0;
 char *input_type = "hw";

```

```

-     fprintf(stdout, "multimod (C) 1996/1997 by Tom Sailer HB9JNX/AE4WA\n"
-         "available demodulators:");
-     for (i = 0; i < NUMDEMOD; i++)
-         fprintf(stdout, " %s", dem[i]->name);
-     fprintf(stdout, "\n");
-     while ((c = getopt(argc, argv, "t:a:s:v:")) != EOF) {
+     while ((c = getopt(argc, argv, "t:a:s:v:q")) != EOF) {
+         switch (c) {
+             case '?':
+                 errflg++;
+                 break;
+
+             case 'q':
+                 quiet++;
+                 break;
+
+             case 'v':
+                 verbose_level = strtoul(optarg, 0, 0);
@@ -445,17 +446,33 @@
                }
            }
+
+     if (!quiet)
+     {
+         fprintf (stdout, "multimod (C) 1996/1997 by Tom Sailer HB9JNX/AE4WA\n"
+             "available demodulators:");
+         for (i = 0; i < NUMDEMOD; i++)
+             fprintf (stdout, " %s", dem[i]->name);
+             fprintf (stdout, "\n");
+     }
+
+     if (errflg) {
+         (void)fprintf(stderr, usage_str);
+         exit(2);
+     }
+     if (mask_first)
+         memset(dem_mask, 0xff, sizeof(dem_mask));
+
+     if (!quiet)
+     {
+         fprintf (stdout, "Enabled demodulators:");
+     }
-
-     fprintf(stdout, "Enabled demodulators:");
-     for (i = 0; i < NUMDEMOD; i++)
-         if (MASK_ISSET(i)) {
-             fprintf(stdout, " %s", dem[i]->name);
+             if (!quiet)
+             {
+                 fprintf (stdout, " %s", dem[i]->name);
+             }
+             memset(dem_st+i, 0, sizeof(dem_st[i]));
+             dem_st[i].dem_par = dem[i];
+             if (dem[i]->init)
@@ -463,16 +480,24 @@
                if (sample_rate == -1)
                    sample_rate = dem[i]->samplerate;
                else if (sample_rate != dem[i]->samplerate) {
-                 fprintf(stdout, "\n");
-                 fprintf(stderr, "Error: Current sampling rate %d, "
-                     " demodulator \"%s\" requires %d\n",
-                     sample_rate, dem[i]->name, dem[i]->samplerate);
-                 exit(3);
                }

```

```

+
+         if (!quiet)
+         {
+             fprintf (stdout, "\n");
+             fprintf (stderr, "Error: Current sampling rate %d, "
+                 " demodulator \"%s\" requires %d\n",
+                 sample_rate, dem[i]->name, dem[i]->samplerate);
+         }
+
+         exit(3);
+     }
+
+         if (dem[i]->overlap > overlap)
+             overlap = dem[i]->overlap;
+     }
-     fprintf(stdout, "\n");
+     if (!quiet)
+     {
+         fprintf (stdout, "\n");
+     }

+     if (!strcmp(input_type, "hw")) {
+         if ((argc - optind) >= 1)

```

-----CUT-----

Bonus



From The Collaborative International Dictionary of English v.0.48 [gcide]:

Assassination \As*sas`si*na"tion\, n.
The act of assassinating; a killing by treacherous violence.
[1913 Webster]

From WordNet (r) 2.0 [wn]:

assassination
n 1: an attack intended to ruin someone's reputation [syn: {character
assassination}, {blackwash}]
2: murder of a public figure by surprise attack

From Bouvier's Law Dictionary, Revised 6th Ed (1856) [bouvier]:

ASSASSINATION, crim. law. A murder committed by an assassin. By
assassination is understood a murder committed for hire in money, without
any provocation or cause of resentment given by the person against whom the
crime is directed. Ersk. Inst. B. 4, t. 4, n. 45.

End of Issue #57



Any Questions?

Editorial and Rants

Schools today are nothing more than centers for brainwashing.

Words Associated with Christianity and British History Taken Out of Children's Dictionary

December 8, 2008 – From: www.telegraph.co.uk

by Julie Henry

Oxford University Press has removed words like "aisle", "bishop", "chapel", "empire" and "monarch" from its Junior Dictionary and replaced them with words like "blog", "broadband" and "celebrity". Dozens of words related to the countryside have also been culled.

The publisher claims the changes have been made to reflect the fact that Britain is a modern, multicultural, multifaith society.

But academics and head teachers said that the changes to the 10,000 word Junior Dictionary could mean that children lose touch with Britain's heritage.

"We have a certain Christian narrative which has given meaning to us over the last 2,000 years. To say it is all relative and replaceable is questionable," said Professor Alan Smithers, the director of the centre for education and employment at Buckingham University. "The word selections are a very interesting reflection of the way childhood is going, moving away from our spiritual background and the natural world and towards the world that information technology creates for us."

An analysis of the word choices made by the dictionary lexicographers has revealed that entries from "abbey" to "willow" have been axed. Instead, words such as "MP3 player", "voicemail" and "attachment" have taken their place.

Lisa Saunders, a worried mother who has painstakingly compared entries from the junior dictionaries, aimed at children aged seven or over, dating from 1978, 1995, 2000, 2002, 2003 and 2007, said she was "horrified" by the vast number of words that have been removed, most since 2003.

"The Christian faith still has a strong following," she said. "To eradicate so many words associated with the Christianity will have a big effect on the numerous primary schools who use it."

Ms Saunders realised words were being removed when she was helping her son with his homework and discovered that "moss" and "fern", which were in editions up until 2003, were no longer listed.

"I decide to take a closer look and compare the new version to the other editions," said the mother of four from Co Down, Northern Ireland. "I was completely horrified by the vast number of words which have been removed. We know that language moves on and we can't be fuddy-duddy about it but you don't cull hundreds of important words in order to get in a different set of ICT words."

Anthony Seldon, the master of Wellington College, a leading private school in Berkshire, said: "I am stunned that words like "saint", "buttercup", "heather" and "sycamore" have all gone and I grieve it.

"I think as well as being descriptive, the Oxford Junior Dictionary, has to be prescriptive too, suggesting not just words that are used but words that should be used. It has a duty to keep these words within usage, not merely pander to an audience. We are looking at the loss of words of great beauty. I would rather have "marzipan" and "mistletoe" than "MP3 player."

Oxford University Press, which produces the junior edition, selects words with the aid of the Children's Corpus, a list of about 50 million words made up of general language, words from children's books and terms related to the school curriculum. Lexicographers consider word frequency when making additions and deletions.

Vineeta Gupta, the head of children's dictionaries at Oxford University Press, said: "We are limited by how big the dictionary can be – little hands must be able to handle it – but we produce 17 children's dictionaries with different selections and numbers of words.

"When you look back at older versions of dictionaries, there were lots of examples of flowers for instance. That was because many children lived in semi-rural environments and saw the seasons. Nowadays, the environment has changed. We are also much more multicultural. People don't go to Church as often as before. Our understanding of religion is within multiculturalism, which is why some words such as "Pentecost" or "Whitsun" would have been in 20 years ago but not now."

She said children's dictionaries were trialed in schools and advice taken from teachers. Many words are added to reflect the age-related school curriculum.

Non-whites destroy "small town feel" in Utah and the main-stream media stands silent. More evidence that wherever blacks or Mexicans move, they ruin the neighborhood.

Garland Loses 'Mayberry' Feel / Leaders Urge Anti-Gang Action Before It's Too Late

December 26, 2008 – From: www.standard.net

by Di Lewis

GARLAND — Marty Hiles said when he moved from California to Garland, he "called it Mayberry."

But now, gang-related break-ins, fights, graffiti and drugs are working their way into his small town, so Hiles, a member of the planning and zoning commission, is doing something about it.

After seeing gang violence transform his California neighborhood, Hiles is determined to act before the problem gets out of control in his new town.

"Utah is being reactive to the growth of crime, and we need to be proactive," he said.

He began a Neighborhood Watch program two months ago, and he and City Councilwoman Jonna Constock are supporting legislation to make it a crime to affiliate with a known gang.

Constock and Hiles also want the community to keep some of the money from juvenile convictions, rather than having the full fine go to the state.

Communities spend a lot of money on rehab and counseling programs to help teens escape criminal activity, Constock said.

"Once a person gets into the prison system, there's not a lot of help for them to get out or trained. There still is for a juvenile. There's a lot of help at the juvenile level."

Part of Hiles' work is making the community aware of what gangs look like.

He has shown the 50 people in the Neighborhood Watch how to recognize gang signs and colors and to report suspicious activity. Also, a member of the Ogden Police Department gang unit spoke to high school students and community members in November.

Garland Police Chief Linda Bourne said this is the first year the community has seen gang problems.

She said she believes gangs are coming to smaller towns, places where law enforcement resources are stretched thin.

Members are outwardly wearing gang colors now, and Bourne said there have been fights between gang members at Bear River High School.

"I think the challenge is just making everybody aware of what's going on where it's never been here before," she said.

"When you think of gangs, you think California, not here, right under your nose."

Because of educating the community, Bourne said, more people have begun to notice gang signs, tattoos and colors and are telling police.

The legislation Hiles and Constock support is a good step, Bourne said, but she believes other laws, such as a gang enhancement for crimes, would be helpful.

"Anywhere that you can try and prevent stuff or help with the prevention is a good thing. You have to start somewhere."

Hiles wants to enact laws to prevent gang problems from growing.

He wants to make the community and the state aware that gangs could quickly become problems where they had not been before.

As director of Neighborhood Watch, he said people have been telling him about increased crime in Garland and Tremonton.

"It's a growing problem, and it seems to be growing rapidly," he said.

Constock, a teacher's aide at Bear River High, said she has seen an increase in gang influence in the school and town and wants parents and residents to start addressing the problem of gangs earlier.

She hopes that by making affiliation with a gang illegal, a teen getting cited for gang activity can be a wake-up call to parents who may not have known their kid was involved with a gang.

"We're seeing the red flags and we need to get something in place, or this will nail the state of Utah," Constock said.

When the Legislature opens in January, Ogden Police Chief and state Sen. Jon Greiner plans to introduce a controversial gang bill that some believe would allow racial profiling.

The bill would allow law enforcement to order gang members to leave areas designated as gang-free. Those who refuse to leave or who return within eight hours could be charged with a misdemeanor.

Dirty spics are not human.

Mexico City Starts Grope-Free Buses for Women

January 22, 2008 – From: www.reuters.com

by Mica Rosenberg

MEXICO CITY (Reuters) – Mexico City has started a women-only bus service to protect female passengers from groping and verbal abuse common on the city's packed public transportation system.

Millions of people cram into subway trains and buses in the Mexican capital, one of the world's largest cities, and women have long complained of abuse from men taking advantage of overcrowding to sneak in an inappropriate grab.

"One time a man stuck his hand up my skirt. They grab your butt ... It's gross," said 27-year-old office assistant Lourdes Zendejas, who waited 20 minutes during the evening rush hour to catch one of the new buses.

The special buses pull up at ordinary stops but have large pink "women only" signs on the front and side. They were added to two busy routes last week and the city government plans to expand the program to 15 other routes by April.

Mexico City's transport system, which also includes hundreds of privately operated "micro" buses, carries twice as many riders as New York's.

"We were constantly receiving complaints of women being leered at, kissed or followed," said Carlos Cervantes, spokesman for the city's public bus system.

Mexico City already had reserved the first three cars in subway trains for women and children but this is the first time the model has been tried in buses.

Women using the new service on Monday had space to sit down and giggled as the driver turned away men at the door.

"This is wonderful. Men never give up their seat for us old people, no one is a gentleman," said 73-year-old Beatriz Perez, whose bulging shopping bags were tucked under her seat.

But not everyone was convinced that having only women would make the ride more pleasant.

"Women can be aggressive too," said telephone operator Rosa Maria Vargas, 42, traveling with her 9-year-old son. "When it gets really crowded, I've been pushed and punched before by men and women."

Amazing! But watch Hollywood ignore him now...

David Spade Buys High-Powered Rifles for Local Police Department

December 23, 2008 – From: www.foxnews.com

by Adam Housley

Actor and one-time Phoenix resident David Spade has donated \$100,000 to the Phoenix Police Department. The department will use the much needed funds to buy high-powered rifles to defend the city from the growing influence of Mexican drug cartels.

Through his publicist, Spade explained that "these guys need to be able to do their jobs, and I am just happy I could help."

Spade says he got the idea for the donation after seeing a story on FOX News. Phoenix police say Spade called asking to donate to their rifle program after he saw that officers, outgunned and desperate for more firepower, wanted to buy their own semi-automatic rifles.

"Mr. Spade has stepped forward and has given a gift to our officers of increased safety," said Police Chief Jack Harris. "I am thrilled that we were able to accept that money that will hopefully bring us to 300 rifles on the street."

Phoenix Police Sgt. Alan Hill says 50 AR-15 rifles to be purchased with the donation will be given to patrol officers.

Spade, 44, grew up in the Phoenix area and graduated from Arizona State University. The "Rules of Engagement" star has helped out cops before, donating \$25,000 to the family of a fallen Phoenix police officer last year.

Most of Bernie Madoff's dirty kike money went to Democrats! Shocking, huh?

Madoff is Active Political Donor

December 14, 2008 – From: www.ft.com

by Ellen Kelleher

Bernard Madoff has been an active political donor in recent years, backing the Democrats more often than not.

Records show that he has contributed tens of thousands of dollars to the campaigns of powerful politicians from his home state, New York, such as Charles Schumer, Hillary Clinton and Charles Rangel, as well as Frank Lautenberg, Christopher Dodd and Richard Gephardt.

His most generous donation was a gift of \$100,000, made over four years, to the Democratic Senatorial Campaign Committee.

He also kept up his ties to the Securities Industry Association, Wall Street's largest trade group where he once led the trading committee, increasing his yearly contributions from \$2,000 in the nineties to \$5,000 in the past few years.

From time to time, Mr Madoff offered financial assistance to Republicans.

In 2000, he wrote a \$1,000 cheque to Vito Fossella, a congressman from Staten Island, whose career has taken a tumble following revelations that he fathered a child out of an extramarital affair, and he also donated thousands of dollars in the nineties to Jack Fields, a former Republican congressman from Houston, Texas.



Niggers! Democrats! Corrupt Labor Unions! Change! LOL!

Average Detroit Home Price \$18,513 – Unemployment Rate 21%

December 21, 2008 – From: www.tribbleagency.com

The Great Depression has reached Detroit. The average price of a home is now \$18,513 and unemployment has reached 21%, and it's expected to get worse. Detroit is facing a crisis of epic proportions that officially puts Detroit statistically (and real term) on par with the great depression. Many readers of Tribble Ad Agency are advertising centric.. and due to the rash of layoffs within all Detroit Advertising firms has put the city on the map for the wrong reasons.

It has become the center of all that is wrong with America... and nothing of what is right.

For example, the crime rate has fallen... because of lack of targets within the city. Meaning there is nothing left to steal. In fact, even the criminals don't want to leave jail.

Heard confirmed that some offenders, notably those without homes of their own, were now expressing reluctance to leave jail when their sentences were done.

Home values have plummeted to levels not seen in 1/2 a century... and the 21% unemployment has in some cases been projected to double within 12 months if the auto industry totally collapses.

To make matters even worse, Detroit has superseded New Orleans as the "worst city" in America... but New Orleans had a Hurricane they could assign blame to... Detroit has no such natural disaster crutch.

"It's a depression – not a recession," McDuell said, with the authority of someone who has lived through both. "It will get worse before it gets better."

It's a man-made disaster.

Regarding a local food bank in Detroit that has seen record numbers of individuals entering the system:

"Many people are first-timers – they have no idea how to navigate the system, how to qualify for food stamps," Wells said. "Last year, some were donors – now they're clients."

In short, last year they donated money into the system... now they are feeding from it because they themselves are in hard financial times.

Detroit needs a miracle, the chances of it showing a resurgence is slim to none in the current economic outlook.

Good article about the fraud of the eternal "victim" mentality.

The Truth About 'Hate Crimes' and the Racial Justice Racket

December 3, 2008 – From: www.baltimoresun.com

by Ron Smith

On Thanksgiving morning, the top right-hand corner of this page quoted Mark Potok of the Southern Poverty Law Center on what he said was the reaction of hate groups to the dawning of the Age of Obama: "We've seen everything from cross-burnings on lawns of interracial couples to effigies of [President-elect Barack] Obama hanging from nooses to unpleasant exchanges in schoolyards. I think we're in a worrying situation right now."

The Southern Poverty Law Center is a thriving business. The Alabama-based "nonprofit" firm has become a font of riches for founder Morris Dees and his associates. Its last tax return (2005) showed it took in nearly \$111 million in donations the previous four years alone and reported assets of \$189.4 million at the end of 2005.

Its business is fundraising, and its success at raking in the cash is based on its ability to sell gullible people on the idea that present-day America is awash in white racism and anti-Semitism, which it will fight tooth-and-nail as the public interest law firm it purports to be. That might lead a skeptic to wonder why it spends little on litigation and why Mr. Dees pockets a lot of money sent in by panicked donors who buy into the smear campaigns against organizations or prominent individuals who question racial preference programs.

To me and to other observant conservatives, the Southern Poverty Law Center is a clever scam, relentlessly cultivating for profit the fear that this nation is filled with Klansmen and rife with people eager to perpetrate genocide. If you're curious about this organization and its legitimacy, spend some time on the Internet and assess it for yourself, because I want to move on to something else related to the comment by Mr. Potok. He mentions cross-burnings on the lawns of interracial couples. If this is true, shame on those who do such things, but what you probably don't know about – and what the law center ignores – is the atrocity committed on an interracial couple in Winchester, Calif.: Marine Sgt. Jan Pawel Pietrzak, a Polish immigrant, and his African-American bride of two months, Quiana Jenkins Pietrzak. Four African-American Marines, two of them under Sergeant Pietrzak's command (including Emrys Justin John, 18, of Baltimore), are accused of breaking into the couple's home and killing them both (one is also charged with a sex crime). In the weeks since the brutal murders, the media have been largely silent about the grisly incident. Would that be the case had the alleged perpetrators been white? Don't be silly.

And as we have come to expect, the authorities won't attribute the Pietrzaks' deaths to "hate." The Riverside County prosecutor's office says the crime was motivated by robbery. But the mother of the murdered Marine, Henryka Pietrzak-Varga, wrote a letter to the president-elect about what happened to her son and daughter-in-law, wondering, "If it was a robbery, why didn't they come when nobody was home instead of in the dead of night, armed to the teeth? ... What was it about my son and daughter-in-law that inspired such hatred and loathing?" As columnist and blogger Nicholas Stix notes, "The questions are, of course, rhetorical. Mrs. Pietrzak-Varga obviously knows full well why her son and daughter-in-law were murdered." "Hate crimes," as trumpeted by the likes of the Southern Poverty Law Center, are a questionable legal construct used almost exclusively against whites.

Hateful or not, interracial violent crime is overwhelmingly black on white or black on Asian. The Department of Justice's figures show that between 2001 and 2003, blacks were 39 times more likely to commit violent crimes against whites than the reverse. Of the nearly 770,000 violent interracial crimes committed every year involving blacks and whites, blacks commit 85 percent and whites commit 15 percent. You won't hear about that from the Southern Poverty Law Center or see it on the evening newscasts, because the truth is one thing and the liberal agenda is another.

Now here's a real hate crime you won't be hearing about!

Two More Face Charges in Restaurant Managers Killing

January 5, 2009 – From: www.freep.com

by Pat Murphy

Two additional arrests have been made in connection with the Oct. 15 shooting of a pregnant restaurant manager killed in a botched robbery attempt in Lathrup Village. Only one of the suspects, however, has been charged with murder.

Deandre A. Sturges, 19, of Beverly Hills and Brandon K. Davis, 20, of Southfield, were arraigned in 46th District Court, Southfield, on Monday afternoon. Each stood mute with not guilty pleas entered on their behalf.

They are the second and third suspects to be charged in the death of Catherine Solinski-Blain, 21, who was shot once in the head about 10:45 p.m. after she had closed the Rib Rack Restaurant, where she was a manager.

Arrested in late November was Jerome A. Hamilton, who turned 16 Wednesday. He was charged with first degree murder but that has been changed to homicide/open murder and his preliminary examination is scheduled for 9:30 a.m. Tuesday.

At the time Hamiltons was charged, Lathrup Village police and the Oakland County Prosecutors office said additional arrests were pending.

Of the two arraigned Monday, only Sturges is charged with open murder, which carries a maximum penalty of life in prison. He is also charged with carrying a firearm in commission of a felony.

He was scheduled for a pre-examination conference at 8:30 a.m. Friday. Judge Susan Moiseev denied bond.

Davis is charged with conspiring to commit armed robbery and being an accessory, after the fact, to the commission of a felony. Through his attorney, Jonathan Jones, Davis waived preliminary examination and was bound over to Oakland Circuit Court.

Jones also asked Judge Susan Moiseev to set bond for his client citing his having cooperated with the investigation. Moiseev, however, denied bond, in part because Lathrup Village Sgt. Vincent Lynch, the lead detective on the case asked bond not be set due to the nature of the crime.

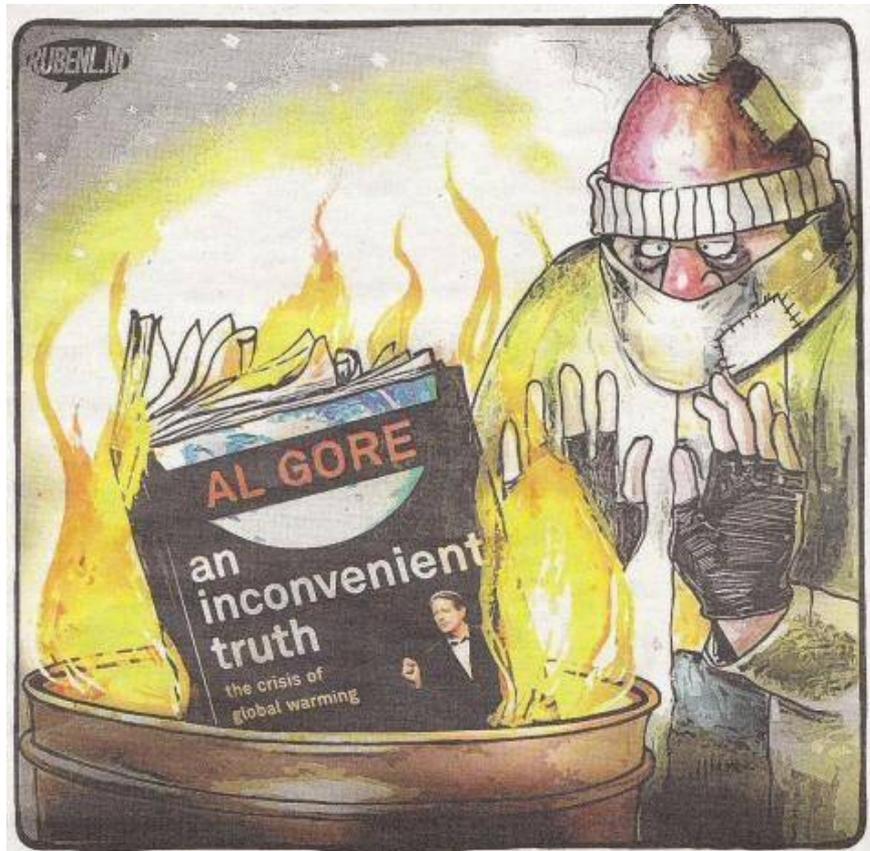


Catherine Solinski-Blain

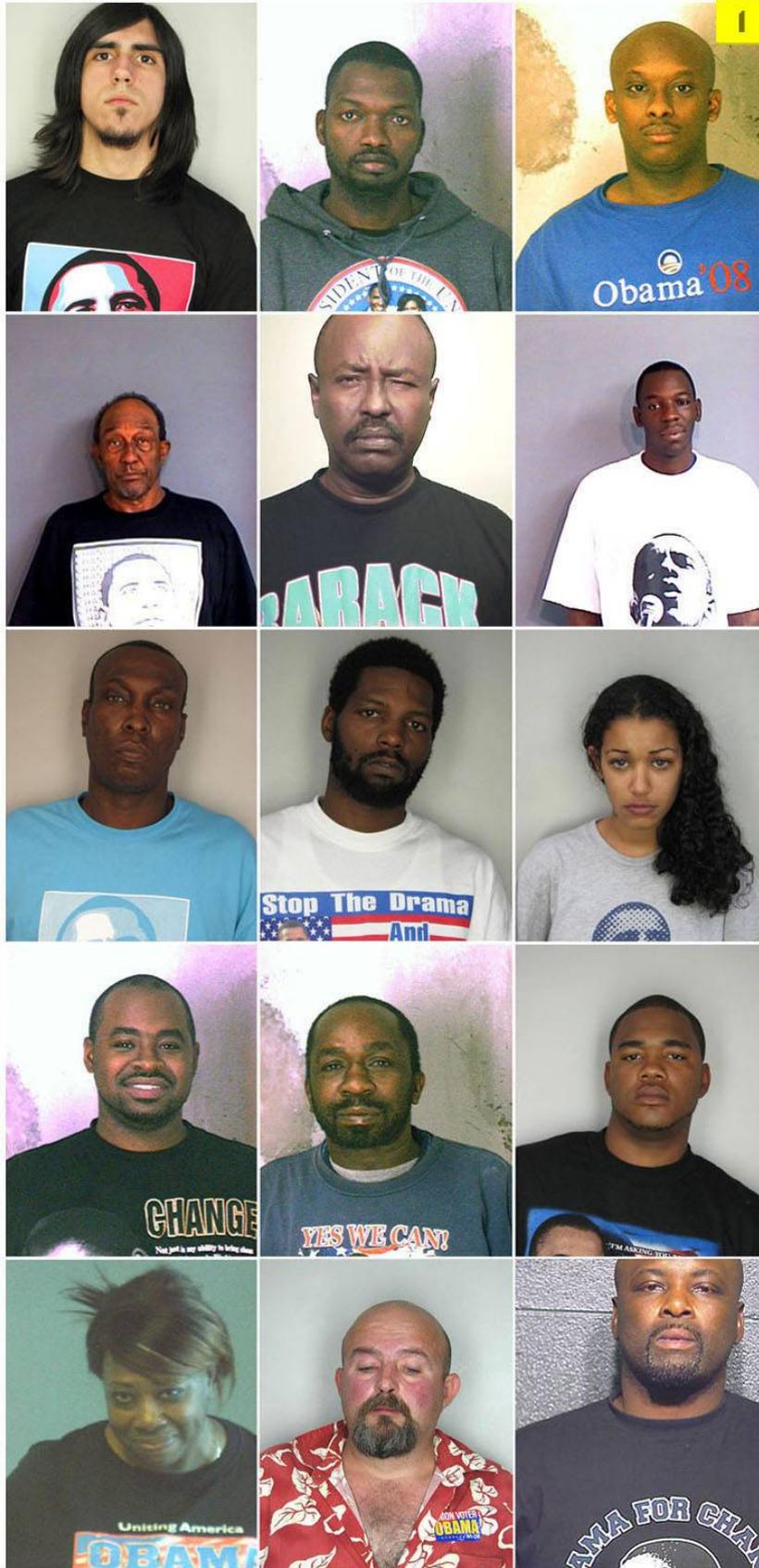


Niggers

From freezing Belgium:



Chicago Police Mug Shots



See a pattern?