"You cannot legislate the poor into freedom by legislating the wealthy out of freedom. What one person receives without working for, another person must work for without receiving. The government cannot give to anybody anything that the government does not first take from somebody else. When half of the people get the idea that they do not have to work because the other half is going to take care of them, and when the other half gets the idea that it does no good to work because somebody else is going to get what they work for, that my dear friend, is about the end of any nation. You cannot multiply wealth by dividing it."

—— Dr. Adrian Rogers of the Bellevue Baptist Church, Memphis, Tennessee.

Table of Contents

- Page 2 / Software System Introduction – Software Description / #1A ESS (Part 7)
  - Detailed analysis of the software "generic" program which a #1A ESS operates under.

- Page 38 / DMS−100 TOPS Holiday Table (HOLIDAY)
  - Ever wanted special telco billing on your birthday?

- Page 40 / DMS−100 TOPS Holiday Treatment Table (HOLTRT)
  - Call treatment options for the DMS−100 HOLIDAY table.

- Page 42 / Radio Shack PRO−2042 455 kHz I.F. Filter Modifications
  - Help improve the selectivity and interference rejection characteristics of the PRO−2042 scanner.

- Page 55 / Bonus
  - Teleprompter President

- Page 56 / The End
  - Editorial and rants.
B. Processor Configuration Recovery (Level 1)

6.56 Processor configuration program entries are the result of A-level or B-level interrupts. The first level of processor configuration recovery is concerned with the speed of recovery. The main memory and file stores are assumed to be good.

6.57 The processor configuration circuit establishes a basic processor for the recovery procedure consisting of a central control, a program store block, and a program store bus. These basic units are forced on-line and the processor sequencer begins a systematic effort to confirm the viability of each basic unit. Level 1 is comprised of 16 states and if the processor configuration is not recovered, escalation to level 2 is automatic. Figure 64 shows the general flow of the level 1 recovery sequence.

C. Processor Configuration Recovery (Level 2)

6.58 The second level of processor configuration recovery is reached when the first level fails to recover the systems or by a manual pump request (A-level interrupt) at the MCC. The level 2 recovery assumes that all software is mutilated and operates only on data that has been pumped from file store or data that has been hash summed. The hardware sequencer forces on-line a central control, a program store, a program store bus, and a file store with which to operate. All other program stores have been forced into the maintenance mode and all other auxiliary units are denied access to an auxiliary unit bus. See Fig. 65 for recovery sequence.

D. Hardware Initialization

6.59 After successful completion of the processor configuration recovery action, selected hardware must be initialized. Initializing, in this respect, includes a limited amount of testing of the selected hardware and updating the software image of the hardware. The hardware that is initialized includes:

- Coded enable peripheral unit bus
- Input/output unit selectors (IOUSs) and input/output unit controllers (IOUCs)
- Central pulse distributor enable address bus (CPDB)

- Central pulse distributors (CPDs)
- Peripheral unit address bus (PUAB).

See Fig. 66 for a flowchart covering hardware initialization.

6.60 Hardware initialization is always followed by validity procedures to ensure good software. Although main memory and file data may have been renewed, there is a chance that some faults may still exist. Since all base level processing has stopped during these high level interrupts (A or B-level), it is important that processing resume with a minimal chance of further interrupts. Following this philosophy, the software is continually verified following any events within the processor community. These events include both levels of processor configuration and system reinitialization.

SOFTWARE INITIALIZATION

A. General

6.61 There are two types of software data: nontransient and transit. Nontransient data is stored in main memory and backed up in file store. It includes the generic program, parameters, and translations. Obviously, this data must be good or the system will not operate properly. Transit data is just as important to the system sanity. Software initialization programs are provided to check the validity of transient data.

6.62 Transit data consists of variable call processing data, i.e., real-time activity data. Since this data is constantly changing, it is only backed up by a duplicate copy available in call store. Transit data is also routinely checked via audits to ensure the reliability of the data. These audits cannot use hash sum calculations due to the nature of the data. Therefore, the audits detect errors by redundancy checks and pattern matching. Serious hardware or software faults can cause mutilation and if recovery time is lengthy, invalid transit data may result. If the transit data is not quickly regenerated, the system will become very unstable. When this occurs, it is necessary to regenerate large portions of transient data. This regeneration is called software initialization and is performed in progressive stages called phases. During execution of some phases, nontransient data is checked via hash sum calculations and corrected, if necessary. However, phase ac-
Fig. 64—Processor Configuration Recovery Level 1—Flowchart (Sheet 1 of 3)
Fig. 64 — Processor Configuration Recovery Level 1 — Flowchart (Sheet 2 of 3)
Fig. 64—Processor Configuration Recovery Level 1—Flowchart (Sheet 3 of 3)
Fig. 65 — Processor Configuration Recovery Level 2 — Flowchart (Sheet 1 of 4)
tivity is primarily directed at regenerating transient data.

6.63 In the No. IA switch, there are four phases of initialization. The phases progress numerically from phase 1 to phase 6, excluding phases 2 and 3. Phases 1, 4, 5, and 6 consist of audits that are stitched together; i.e., the priority of the audits is raised, and they are executed consecutively. Each phase is more comprehensive than the previous one and has a more drastic effect on the system. For example, phase 1 initializes a relatively small portion of transient call store data. During a phase 6 all calls in progress are knocked down. Once a phase is trig-
triggered, it must run to completion. Phases are automatically or manually initiated in response to sanity affecting faults. A phase 1 can be triggered manually or automatically and, if the system is unable to resume call processing, the system automatically advances to a phase 4. If the system encounters difficulties while processing a phase 4, the system automatically advances to a phase 5. If a phase 5 fails to recover the system, it will loop in a phase 5 because the system cannot automatically advance to a phase 6. A phase 6 can only be initiated manually.

**B. Phase Triggers**

6.64 There are ten sources that trigger phases. There are three phase triggers generated internally by the phase control program. These triggers deal with base level malfunction, unanswered interject, and audit time-outs during a phase. The other seven phase triggers are generated externally to the phase control program. All ten phase triggers can be classified into four categories. Each one has several individual trigger code numbers which identify the source of phase activity. The four categories are:

(a) **Manual Request:** These triggers are related to manual actions by the maintenance personnel (eg, a manual phase 6 requested at the MCC).
Fig. 65 — Processor Configuration Recovery Level 2 — Flowchart (Sheet 4 of 4)
Fig. 66—Hardware Initialization—Flowchart (Sheet 1 of 4)
Fig. 66—Hardware Initialization—Flowchart (Sheet 2 of 4)
Fig. 66—Hardware Initialization—Flowchart (Sheet 3 of 4)
SECTION 231-045-005

Fig. 66—Hardware Initialization—Flowchart (Sheet 4 of 4)
(b) **Program Checks:** These triggers are related to program check failures which cause a phase (e.g. unanswered interrupt).

(c) **Phase Problems:** These triggers are related to problems encountered during a phase execution (e.g. a phase exceeds a time limit).

(d) **Automatic Interrupt Triggers:** These triggers are related to hardware faults that cause a phase (e.g. duplicated call store failure).

### C. Phase Activity

#### Phase 1

6.65 Phase 1 is the least comprehensive of the phases and has the shortest duration. Phase 1 is completed in 2 seconds. Phase 1 has virtually no effect on calls in progress in either a stable state (i.e., an established talk path) or an unstable state (e.g., digit reception, ringing, etc.). Any RCs in progress are removed when the phase is triggered. All nontransient data that is duplicated in the file store is hashed and corrected, if necessary. Overwrites in progress and active utility executes are removed and the appropriate interrupts are inhibited. The audits run in phase 1 include audits 3 through 11, 18, 32, 40, and 71. These audits (e.g., network management audit, ring line scan audits, etc.) check and, if necessary, generate constant values within the transient data. If phase 1 is triggered by data validation failure, additional audits are run in conjunction with audits normally requested. The additional audits are audits 24, 34, 39, 42, 43, 44, and 45. See Table E for an overview of phase activity.

#### Phase 4

6.66 Phase 4 is the lowest level phase that zeros a specified area of call store. This area is defined by office parameters. Calls in the stable state are maintained; however, calls in the unstable state are knocked down. Nontransient data that is duplicated in file store is hashed and corrected, if necessary. RCs in transition to call store or RCs already processed are removed. Overwrites in progress and active utility executes are removed and the appropriate interrupts are inhibited. The audits run in phase 4 include audits 0, 4-10, 16, 18, 19, 22, 32, 37-43, 46, 48, 50, 52, 58-63, and 71. Phase 4 completes in 25 seconds (see Table E).

#### Phase 5

6.67 Phase 5 consists of the same audits as phase 4, but it is more comprehensive. Phase 5 is the first possible pump phase; i.e., processor configuration state counter must be greater than or equal to 16. This implies that the system has been hashed and if necessary pumped; i.e., nontransient data in main memory has been replaced with data from file store or it has been verified via hash sums. Once all transient memory corrections have been attempted (via phases 1 and 4) and a phase still triggers, the problem is either nontransient data or hardware. Main memory and disk overwrites in the copy state are removed. Calls in the stable state are maintained while calls in the unstable state are knocked down. RCs are handled the same as in phase 4. An office parameter defined area of call store is zeroed. The utility executes are removed and the appropriate interrupts are inhibited. When phase 5 is repeated, additional interrupts are inhibited (see Table D). Phase 5 completes in approximately 25 seconds.

#### Phase 6

6.68 Phase 6 which can only be initiated manually is the most comprehensive and drastic phase. All calls, whether stable or unstable, are knocked down. The entire network and software subsystem is placed in the idle mode. Phase 6 like phase 5 is a pump phase. The processor configuration state counter must be greater than or equal to 16. Main memory and disk overwrites in the copy state are removed. Phase 6 also zeros an office parameter defined area of call store and removes any utility executes in progress. The H-, G-, F-, K-, and D-level sources are inhibited. When phase 6 fails, it steps down to run a phase 5 and it is treated as a repeated phase 5 and additional interrupts are inhibited (see Table E). Phase 6 completes in approximately 32 seconds.

### EMERGENCY MODE CONTROL SOFTWARE

6.69 Failure of the processor configuration recovery sequence to establish a viable processor configuration necessitates manual recovery procedures. These are invoked through controls at the MCC. The first manual recovery step taken consists of establishing a basic configuration using the override control keys and requesting the second level of processor configuration recovery. The override control keys have the advantage over the basic configuration sequence of being able to force a basic
### TABLE E

#### 1A PHASE ACTIVITY

<table>
<thead>
<tr>
<th>ACTIVITY</th>
<th>PHASE</th>
<th>1</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>OVERVIEW</td>
<td>Constant’s Audits</td>
<td>Zero CS Data</td>
<td>First Pump Phase</td>
<td>Manual Only</td>
<td></td>
</tr>
<tr>
<td>DURATION</td>
<td>1-2 sec</td>
<td>25 sec</td>
<td>25 sec</td>
<td>32 sec</td>
<td></td>
</tr>
<tr>
<td>EFFECT ON NONTRANSIENT DATA</td>
<td>Hash/Correct</td>
<td>Hash/Correct</td>
<td>Pump PC (state ≥ 16)</td>
<td>Pump PC (state ≥ 16)</td>
<td></td>
</tr>
<tr>
<td>EFFECT ON CALL IN PROGRESS</td>
<td>STABLE</td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>Knocked Down</td>
</tr>
<tr>
<td>EFFECT ON CALL IN PROGRESS</td>
<td>UNSTABLE</td>
<td>None</td>
<td>Knocked Down</td>
<td>Knocked Down</td>
<td>Knocked Down</td>
</tr>
<tr>
<td>RCS IN PROGRESS</td>
<td>Canceled</td>
<td>Canceled</td>
<td>Canceled</td>
<td>Canceled</td>
<td></td>
</tr>
<tr>
<td>CS ZEROED</td>
<td>None</td>
<td>Same Compool and Parameter Defined Data</td>
<td>Same Compool and Parameter Defined Data</td>
<td>Same Compool and Parameter Defined Data</td>
<td></td>
</tr>
<tr>
<td>UTILITY EXECUTES AND LIBRARY ACTIVITY</td>
<td>Removed</td>
<td>Removed</td>
<td>Removed</td>
<td>Removed</td>
<td></td>
</tr>
<tr>
<td>OVERWRITES IN PROGRESS</td>
<td>Main Memory in Copy State Removed</td>
<td>Main Memory in Copy State Removed</td>
<td>Main Memory and Disk in Copy State Removed</td>
<td>Main Memory and Disk in Copy State Removed</td>
<td></td>
</tr>
<tr>
<td>INTERRUPTS INHIBITED</td>
<td>H, G, F, K, and D (auxiliary unit read/write failure only)</td>
<td>H, G, F, K, and D (auxiliary unit read/write failure only)</td>
<td>First Phase 5 – Repeated phase 5 – H, G, F, E, K, D (all sources) and B (GBP sources only)</td>
<td>As requested</td>
<td></td>
</tr>
<tr>
<td>AUDITS RUN</td>
<td>3, 4, 5, 6, 7, 8 9, 10, 11, 18, 32, 40, 71</td>
<td>0, 4, 5, 6, 8, 9 10, 16, 17, 18 19, 22, 32, 37, 38, 39, 40, 41, 42, 43, 46, 48, 50, 52, 58, 59, 60, 61, 62, 63, 71</td>
<td>Same as phase 4 0, 4, 5, 6, 7, 8, 9, 10, 12, 14, 16, 18, 19, 20, 22, 24, 26, 32, 34, 36, 37, 38, 39, 40, 41, 42, 43, 46, 48, 50, 58, 60, 61, 62, 63, 64, 71</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
configuration which fault recovery programs cannot change.

6.70 If the system still fails, a final set of manual recovery procedures is initiated. It involves forcing the system into an emergency mode of operation in which only manually initiated tasks are executed. All other tasks (including call processing) are discarded. In the event of excessive call store, program store, or file store failures, this emergency mode can be entered with a minimal processor configuration (EMERMIN) that consists of a central control and only sufficient memory to execute maintenance tasks. The emergency mode may also be entered with a complete memory (EMERFULL) in the event of peripheral faults or program problems that cause a loss of system sanity (see Fig. 67).

6.71 Failure to recover system sanity through the override controls may be due to mutilated nontransient data in both the main memory stores and file data. Therefore, the next step in manual recovery is to reload this data from tape. This is called system reinitialization. The program initiates the load of data from tape and directs programs loaded with this data to configure a complete processor.

PROCESSOR RECOVERY SOFTWARE

A. General

6.72 The processor fault recovery programs (Fig. 68) are normally entered as a result of a maintenance interrupt. However, they may also be entered on interject or via the input/output handler program, via routine exercise programs, via the maintenance control program for deferred (base level) fault recovery testing, or via manual requests from the TTY. But, the primary purpose of these programs is to restore the system to call processing in the face of system errors or faults.

6.73 The fault recovery programs are designed to isolate faulty units or subsystems rather than to identify replaceable components. These programs recognize and isolate most call-affecting faults during a single interrupt interval.

6.74 The basic techniques of fault recovery strategy are centered around rapid resolution of problems and quick return to normal system operation. The fault recovery programs report error data to the error analysis programs. Error analysis maintains a history of interrupts and associated data.

6.75 After the fault recovery program has selected a working configuration of hardware, the program must perform several "housekeeping" tasks. The program must set appropriate flags that will cause base level after the system has returned to call processing. Also, the program must record the actions it has taken in the appropriate error analysis data history.

6.76 Finally, the fault recovery program initiates output messages to convey its actions to maintenance personnel. If several interrupts have failed to resolve a persistent problem, output messages may be utilized to supplement the automatic error analysis. The maintenance personnel may analyze the output messages and select a working configuration of hardware manually.

6.77 The 1A Processor community has six components, each of which could default at any time for various reasons. Fault recovery programs are written for each component and these programs interface with many other programs in an effort to recover the system sanity. Central control, call store, program store, file store, auxiliary unit buses, and data units are all subject to faults and their recovery is done on an individual basis.

B. Central Control Fault Recovery

6.78 Duplicated central controls are the primary functional elements of the processor community. The central controls interface with all internal and external signal and control buses and provide the processing capability for the system. For reliability purposes, the two central controls are connected in parallel. Either one can control the system operation. The normal system configuration provides for the two central controls to operate in step, each performing matching checks on the other.

6.79 One central control functions as the active unit and the other functions as the standby. During this normal mode of operation, both central controls are matched to ensure that they execute the same instructions, receive the same data, and make the same conditional decisions. In the event that the active central control malfunctions, the standby central control is designated active and assumes control of processor functions (the switch of active and standby central controls may be accomplished automatically under program control, by the processor configuration hardware, or by manual activation from the control and display panel).
Software System Introduction – Software Description / #1A ESS

SECTION 231-045-005

SEVERE HARDWARE (SOFTWARE FAULT THAT CANNOT BE CLEARED OR ISOLATED VIA PC, SI, OR IS ENCOUNTERED)

MANUALLY ESTABLISH AN EM CONFIGURATION

ADEQUATE FUNCTIONAL PROCESSOR HARDWARE AVAILABLE?

NO

YES

ONE CC
ONE PS BUS
A SIMPLEX PS COMMUNITY
ON CS BUS
A SIMPLEX CS COMMUNITY
ONE OFC (AQ BUS AND FS)
RELIABLE COPY OF ALL NON-TRANSIENT SYSTEM DATA

EMERMIN
ONLy PARTIAL PROCESSOR HARDWARE/SOFTWARE REQUIRED
ONE CC
ONE PS BUS
FOUR PS K-CODES
ON CS BUS,
TWO CS K-CODES
FEATURES AVAILABLE:
REQUEST PROCESSOR DIAGNOSTICS
PERFORM OVERWRITES AND DUMPS
WRITE TAPES

EMERFULL
SEVERE PERIPHERAL FAULTS OR DATA ERROR THAT PREVENTS NORMAL SYSTEM OPERATION
ADEQUATE FUNCTIONAL PROCESSOR HARDWARE
FEATURES AVAILABLE
ALL EMERMIN FEATURES, PLUS:
REQUEST PERIPHERAL DIAGNOSTICS
REQUEST SYSTEM DIAGNOSTICS
INITIATE LIBRARY PROGRAMS

NORMAL SYSTEM OPERATION

Fig. 67—Emergency Mode Configuration
6.80 System troubles are normally detected by trouble-detection circuits and the call processing interject. As stated earlier, its primary purpose is to verify the integrity of the active central control; however, the tests that are used to perform the verification vary and depend upon the conditions under which the program is entered.

6.81 Recovery is organized around a common control program (Fig. 69) which calls one or more independent test routines. Control is based on a dynamic control word. The recovery programs initialize the control word to specify which test routines to run on each entry. The control word is normally set to all zeros which is recognized as an invalid entry condition. The control word consists of three fields of bits. Within the first field, each bit has a one to one relationship to a test routine which tests a portion of the central control. The second field consists of a single bit and signifies a special test procedure used only on B-level interrupts caused by a pulse source failure. The remaining bits make up the third field and indicate the origin and the termination of the request.

6.82 The program has several different entry points, each with its own unique requirements. A preprocessor program is provided for each input to prepare the necessary information and perform the required initialization before entering the common control program. At the conclusion of the common control program, a termination program is provided for each entry. There is also a special termination program in case the control programs find the control word to contain all zeros. These terminating programs perform access tests on other subsystems, update status words, request subsystem normalization, update error counters, and in general perform cleanup and housekeeping tasks before returning the system to call processing.
C. Call Store Fault Recovery

6.83 The primary purpose of call store recovery is to return the system to normal call processing (Fig. 70) as quickly as possible after a fault or error has been detected in the call store community. Therefore, whenever possible, the program will remove the faulty store from service on a "first-look" basis. The first-look approach utilizes error indicators in the central control, the bus configuration, and the store status to identify the faulty store.

6.84 The first-look approach replaces the faulty unit with a duplicate if one is available. If a duplicate is not available, the recovery selects a store (from the duplicated call stores) and initiates a copy of the suspected unit into the selected unit. If tests are unable to detect trouble within the faulty store, it is restored to service before the update completes. When the update is completed, the updated store is placed into service and the suspect store is removed and diagnosed. If the faulty unit fails again before
the update is completed, the faulty unit is removed from service and the system must wait for the successful completion of the substitute store's update from a backup copy stored on file store.

6.85 However, when the failing store is not duplicated and a store is not available for selection as a substitute, the first-look approach is not used. Call store fault recovery uses the error history of the failing store to determine the action to be attempted. If the store's error history is acceptable, call store fault recovery corrects the failing word, performs a complete access test of the store, and restores it to service.

6.86 An unacceptable error history or a failure of the complete access test causes call store fault recovery to perform a bootstrap of the call store community. The bootstrap routine attempts to assemble a complete copy of the call store information by using only stores that pass the bootstrap tests. If necessary, call stores will be used regardless of their status prior to the failure. Should the bootstrap fail to establish a valid copy of call store, a program transfer is made to the processor configuration recovery program to switch central controls (B-level interrupt).

6.87 The call store service routines and bootstrap routines may also be entered from central control fault recovery, auxiliary unit fault recovery, and processor configuration programs. These programs may use the service routines to verify access to call stores. In addition, central control fault recovery may enter the call store bootstrap routine if one or more call store memory blocks are not error-free or are not provided in the configuration that has been established. Therefore, call store bootstrap is entered to recover a valid call store configuration.

**D. Program Store Fault Recovery**

6.88 The 1A Processor program store community is made up of a number of individual program
stores. The number of program stores varies according to the type and size of the switching office installation. System software (including program store fault recovery) can accommodate 33.

6.89 The central controls access the program stores via duplicated program store buses that interconnect every program store frame with both central controls. (The buses may have as many as two branches.)

6.90 Each program store word location is identified by a unique address. This address consists of a K-code and a data location address. The K-code portion of the address identifies the specific program store to be addressed. The data location identifies the specific location to be accessed within the program store.

6.91 Two program stores are normally designated as spares (called rovers) and can be assigned to replace any program store which malfunctions. During normal operation, the spare program stores contain duplicate copies of information stored in program store.

6.92 The primary purpose of program store fault recovery is to return the system to normal call processing (Fig. 71) as quickly as possible after a fault or error condition has been detected in the program store community. Therefore, whenever possible, the fault recovery will remove the faulty store on a first-look basis. The first-look approach works in the same manner as call store fault recovery first look. The first-look approach utilizes error indicators in the central control, the bus configuration, and the store status to identify the faulty store.

6.93 When the trouble is located in duplicated program store, the suspect unit is removed from service, and the remaining is set to operate as if it were not duplicated. An unduplicated block of memory requires further analysis. Program store fault recovery attempts to find a rover store that can be loaded with a copy of the suspect memory block. If it is able to select a rover store and initiate the copy of the suspect store, it then checks the error history (kept by program store fault recovery) of the suspect store. If the history is acceptable, the store is left in service until the rover update is completed.

6.94 If the error history is unacceptable, the suspected unit is removed from service and the system must wait until the rover is filled from file store. Also, the program checks to see if a previous error has resulted in a rover store being prepared as a duplicate for the suspected block of memory. If a rover has been updated, the rover is placed in service and the suspect unit is removed. If a rover is in the process of being updated, the system waits for the update to be completed and replaces the suspect unit with the rover store.

6.95 After a configuration of program stores has been selected, an access test is performed on each memory block to verify the integrity of the program store community. Failure of the access test after the suspect store has been removed from service causes the program to transfer to the program store bootstrap routine.

6.96 Bootstrap attempts to assemble a complete copy of program store using only stores which pass the bootstrap qualifying tests. The bootstrap is considered successful if a full copy of program store (with or without the use of rover stores) has been assembled. Failure of bootstrap results in a transfer to the processor configuration recovery program to switch central controls (B-level interrupt). After a successful fault recovery, control is returned to normal processing.

E. Auxiliary Unit Fault Recovery

6.97 The 1A Processor system has a bus system which enables autonomous processing units to access the call store and program store bus system of the central control. The autonomous processing units are referred to as auxiliary units and there may be as many as 16 auxiliary units on the auxiliary unit bus. The auxiliary unit bus is linked to the call store and program store buses by special hardware in the central control which is called the auxiliary unit bus sequence (AUSQ). The AUSQ resolves bus occupancy conflicts among auxiliary units on the auxiliary unit bus and resolves bus, store, or AU occupancy conflicts between the central control or any auxiliary unit on either the auxiliary unit, call store, or program store bus. This document will refer to the auxiliary units and the AUSQ as the auxiliary unit bus system.

6.98 The auxiliary unit bus system will have at least two and a maximum of four file store controllers in the file store environment. A file store controller may control from one to four disk files on
which a large amount of data can be stored. A file store controller is a wired logic processor which will process central control request(s) to transfer a data block between the relatively slow serial access memory of a disk file and the fast random access memory of a call store or program store. For reliability, the memory content of one set of disk files associated with a file store controller will be duplicated on an identical set of disk files associated with another file store controller. Collectively, all file store controllers with associated disk files are referred to as the file store system. The file store system serves as a primary data backup and bulk data storage facility for program, translation, and other information for the 1A Processor system.

6.99 #The auxiliary unit bus system, in the APS environment, will have at least 2 and a maximum of 16 APUs. The APS is a high capacity disk system for the 1A Processor. The API allows the sending and receiving of messages and blocks of data between the 1A Processor and up to eight 3B Processors. #

6.100 The auxiliary unit bus system will also have at least two and a maximum of four data unit selectors. A pair of data unit selectors may control from two to 16 data units. A data unit selector is similar to a file store controller except that instead of disk files, the data unit selector is designed to handle slower data devices such as tape units. The data unit
system serves as a backup to the file store system for system reinitialization and as a primary facility for program updating, automatic message accounting data recording, and other functions.

6.101 The auxiliary unit fault recovery program is designed to function under a number of diverse conditions: interject, D-level interrupt, base level maintenance and other processor interrupt levels. Basically, the program employs the first-look approach to fault recovery. The first-look approach involves the retrying of the failing operations utilizing simple and fast-testing techniques. If this approach fails to identify the source of the trouble, fault recovery will then resort to more detailed testing to isolate the problem.

6.102 The auxiliary unit fault recovery program will be entered basically under three conditions. The first condition involves maintenance action for the auxiliary unit bus system. This maintenance action will ordinarily be initiated through the interject request mechanism instead of the normal maintenance interrupt control hardware sequencer. This method is used because auxiliary unit processing is independent of central control processing and can be momentarily deferred without degrading system performance.

6.103 In the second condition, the fault recovery will also be entered on D level from the call store fault recovery program when the central control encounters an auxiliary unit read/write failure. An auxiliary unit read/write failure may occur when the central control addresses an auxiliary unit and an accessing error is detected by either the central control or auxiliary unit. The central control may also address an auxiliary unit which is in a troubled state and has requested maintenance action through the interject mechanism. An auxiliary unit which makes an interject request will not respond to central control addressing until it has been restored to service by the specific type of auxiliary unit fault recovery program. The program processes D-level entries basically as it processes interject entries.

6.104 Finally, the program may also be called by TTY request or by the processor configuration or another processor fault recovery program to test or reconstruct the auxiliary unit bus system interface with the central control system.

F. File Store Fault Recovery

6.105 The 1A Processor utilizes a file store or APS to provide backup storage for program and translation data. The file store or APS is also used to store programs and data that are infrequently used and consequently are not normally kept in the program or the call store. The disk memory used by the file store possesses serial rather than random access characteristics. Because of its serial character, the time required to retrieve or store data from the file store is variable (a function of the position of the disk when the request is made). Because time required to retrieve or store data is on the order of milliseconds, it is not practical for the central control to directly access disk memory. Instead, a file store controller is provided to perform this function. The file store controller is a special purpose wired logic processor which buffers requests from the central control to read from or to write into disk memory and transfers information from disk to main memory or from main memory to disk.

6.106 Each file store contains one to four disk files. File stores are arranged in pairs, and each pair is referred to as a community. The 1A Processor software is designed to accommodate a maximum of two communities. File store 0 (on bus 0) and file store 1 (on bus 1) make up one community; file store 2 (on bus 0) and file store 3 (on bus 1) make up another community.

6.107 When the fault recovery program for file store is entered, it determines the type of error, increments the counter, and checks to see if the counter limit has been reached. If the counter had not reached its limit, the error is recorded, and a return to the calling program is executed. If the counter has reached its limit, the file controller or disk file is removed from service.

6.108 Whenever a file store controller or disk file is removed from service, a diagnosis is requested. Because the removal of a file store controller from service could mean that as many as four disk files would be inaccessible, every effort will be made to leave in service as many disk files as possible. Therefore, when the source of the trouble may be either the file store controller or disk file, only the disk file will be removed from service. If it is determined that the disk file only contains an error-prone record, the record will be rewritten and verified instead of immediately requesting a diagnosis.

6.109 Because the read or write of a disk file record is a relatively time-consuming process, all file store fault recovery maintenance actions that
require a disk file access operation are deferred. All disk file access operations are processed as a standard job request through the normal file store administration program routines. Furthermore, those disk file related error sources which are expected to have a relatively high rate of occurrence are processed through the status failure report mechanism. Consequently, they will not require the more time-consuming and service-affecting actions of normal maintenance procedures.

G. Attached Processor System Single Strategy Fault Recovery (SSFR)

6.110 Fault recovery of 1A Processor subsystems has traditionally been handled by a single program for each subsystem. However, with the addition of the APS, fault recovery is handled by two programs with a common recovery control. Fault recovery in the APS is divided into two major categories. The first category is fault recovery on interrupt or interject level and is handled by either the auxiliary unit fault recovery program (AUF R) or the attached processor fault recovery program (APFR). The second is fault recovery on base level that is handled by the APFR. The size and complexity of a fault recovery package for each major category resulted in a single recovery package serving both functions. The single recovery package is called the single strategy fault recovery (SSFR). It provides common recovery control for both interrupt or interject level faults and base level maintenance faults.

6.111 The SSFR does fault recovery tasks for faults occurring in either the active or the standby APU. These faults or failures may be initiated by either the 1A or the 3B processor. The SSFR is divided into four major areas:

- Interrupt and interject control
- Base level maintenance control
- Common recovery control
- Timing administration

The interrupt and interject control takes place in the AUF R, and the base level maintenance control is handled within the APFR.

APS Organization

6.112 The APS replaces the disk file system on either the No.1A or No. 4 ESS switch. The APS consists of one to eight 3B processors connected to the 1A Processor through an API system. The APS provides the 1A Processor disk access to a high-capacity 3B disk system. The API allows the sending and receiving of messages and blocks of data between the 1A and 3B Processor Systems. The API supports the APCL protocol between the 1A and 3B processors. The APCL protocol has both efficient block transfer and message-handling capabilities. The APCL protocol also includes a high-priority maintenance message communication capability that is supported by the API. These messages are communicated in a closely coupled, synchronous, high-priority way by using the 3B input/output interrupt and the 1A auxiliary unit bus maintenance interject mechanisms.

6.113 The APS includes attached processor message handlers on both the 1A and 3B sides of the API (Fig. 72). Also included are the file manager interface, the file manager, the disk driver, and the disk file controller, all on the 3B side.

Interrupt and Interject Control

6.114 The AUF R program is the interrupt and interject control program for the SSFR. All auxiliary unit interjects and D-levels are first handled by AUF R; consequently, AUF R processes the interject or interrupt and tries to isolate the problem. The problem may be in the central control, the main memory, the auxiliary unit bus or in an individual auxiliary unit. In the APS version of the 1A Processor, the auxiliary units are the API and the data unit selector.

6.115 If the AUF R determines the fault was caused by an auxiliary unit, AUF R communicates with the fault recovery programs for the faulty auxiliary unit. The AUF R program communicates with the unique fault recovery programs through a transfer vector table. For interrupt and interject processing, AUF R requests the unique fault recovery programs to do these tasks:

- Load unique bins
- Process the unique trouble
- Report data
- Update plant measurements

If the API was the faulty auxiliary unit, APFR is the unique fault recovery program and is called to do the above tasks.
Section 231-045-005

**Base Level Maintenance Control**

6.116 The APFR program is the base level maintenance control program for the SSFR. During normal operations of the APS, there is continual checking for errors in these major areas:

- Link integrity monitor
- APS message handler
- System audit of disk

6.117 The link integrity monitor does a check every second on the active and standby links to the 3B. Base level maintenance is called when there is no access to the 3B or the API through either the active or standby link.

6.118 The APS message handler calls a service routine within APFR to get the K-code of the active API. If an active API cannot be found, a failure is returned to the message handler. The message handler will then call base level maintenance control.

6.119 There is an audit done by the System Audit for File Store Administration Program (SADK) that monitors disk activity to ensure that jobs are being completed. If jobs are being accepted but not completed within a certain time, base level maintenance is called.

6.120 The base level maintenance control routine, APFRBLM, does three primary functions:

(a) Gets a base level maintenance report printed on the TTY
(b) Stops all auxiliary units
(c) Sets all interrupt inhibits.

**Common Recovery Control**

6.121 The common recovery control routine, APFRTBL, may be called from either Inter-
rupt and Interject Control or Base Level Maintenance Control. The purpose of this routine is to determine the error and to recover from the error condition. The error and recovery information is formatted into proper form for printing on the TTY. The recovery interaction between APPRTBL and AUFRT is important since the AUFRT routine, known as double trouble (APPRTBL), is the backup recovery used by APFR.

6.122 Common recovery control functions are:
- Error detection
- Error analysis
- Error recovery
- Error termination.

6.123 The error information word is built during error detection. This word is used to record as much information about the fault condition as possible. This information is used by the recovery module to determine which course of action to take for recovery.

6.124 One of the basic strategies of error recovery is to initialize the buffers only when absolutely necessary. If recovery can be made without initializing the buffers, jobs to and from the 3B will not be affected; but, if the buffers require initializing, all the jobs in the buffers are lost.

6.125 Another strategy of error recovery is to have a configuration of the API regardless of the state of the finite state machine (routine APPRTBL). The current active API may be reconfigured or its mate can be configured. The API that is configured will be the active API upon return to the system. There is one exception. When a fault occurs in the standby API and it is removed from service, there will not be any configuration performed on the active API. If an API is configured, the following actions take place:

1. The peripheral interface controller (PIC) is reset.
2. The API is informed about the location of the common buffer resources.
3. The appropriate state for the API and update status is determined.

4. The 3B is informed of the configuration of this API.
5. The MCC lamps and power switch lamps are updated.

6.126 Following the recovery actions in any state, the active link to the 3B is tested to ensure communication between the 3B and 1A exists before the fault recovery ends.

6.127 Another strategy of error recovery is for all recovery routines to have a pass-fail indication. Error recovery does not assume that recovery actions are done successfully. The pass-fail indications from recovery modules allow for intelligent decisions to be made within error recovery.

6.128 The final strategy of error recovery is the recording of all recovery actions. These recovery actions are recorded in the recovery information word in memory. The recovery information word is included in the printout and is used for determining the exact recovery actions taken. If any recovery module fails, the reason for failure is saved in memory and is also included in the printout.

6.129 The error termination function used by all three states to end processing is the same for all states. It does three functions:

1. Saves all information gathered during processing
2. Formats information for printing
3. Updates lamps.

Timing Administration

6.130 The common recovery control routine, APPRTBL, has three states: 0, 1, and 2. The processing of a fault may begin in any state and will end in the same state. To return the routine to state 0, a sequence timer allows the state counter to be reset to 0 after a certain time has elapsed.

H. Data Unit Fault Recovery

6.131 The data unit fault recovery program performs the fault recovery tasks for the ADS consisting of data units connected to the auxiliary unit bus system. The data unit fault recovery ap-
SECTION 231-045-005

The approach consists of finding a set of data units which is capable of carrying out the normal tasks associated with data stored on tape. This determination is made by a detailed set of tests which are run on interject or interrupt level priority or on demand via a TTY message. In order to perform its functions, the fault recovery program interfaces with a number of other programs.

6.132 The fault recovery program is designed to operate under D-level interrupts, maintenance interject, base level maintenance, and on demand via TTY request. The data unit fault recovery performs the following functions:

- Removals and restorals of data units
- Tests of bus circuitry, internal data unit selector registers, and data unit selector flip-flops
- Performs configuration of the data unit community
- Administers diagnostic requests
- Administers TTY message input and output requests
- Provides a common interface for other system programs.

6.133 In some cases the fault recovery program utilizes the first-look approach to fault recovery which consists of retrying the failing operation with simple and fast-testing techniques. If this approach fails to identify the source of trouble, the program resorts to more detailed testing to isolate the problem.

6.134 The data unit fault recovery program attempts to keep the tape units normalized as much as possible. This means configuring an equal number of tape unit controllers to each data unit selector. System conditions may require that the data units be configured differently to maintain a viable auxiliary data system. The normalization scheme is exercised once a day at some nonbusy hour to equally divide the TUCs between the data unit buses. At this time, all tape unit controllers are switched from the data unit selector they have been configured with to the other data unit selector. This is done to ensure that all communication paths are exercised. The tape unit controllers and data unit selectors are also exercised (diagnosed) once a day to ensure that they are capable of being used as the need arises. This exercise is also performed at a nonbusy hour.

ERROR ANALYSIS SOFTWARE

A. General

6.135 The 1A Processor error analysis program is an on-line information storage and retrieval (data administration) program which collects system data used to aid in the resolution of difficult hardware and software problems. The data base maintained by this program is also useful in making relatively detailed assessments of system performance from a maintenance viewpoint.

6.136 A major objective of error analysis is to provide a data base comprising a relatively long-term history of system maintenance actions. Full access to this data base is provided in order to enhance the capabilities of the craftsman for maintaining the system. The data base is particularly useful in resolving faults of a transient, intermittent, or marginal nature which are not inherently reproducible. It is also useful in the investigation of subsystem interface faults whose symptoms may at first be misleading.

6.137 The 1A Processor error analysis program does not perform automatic analyses of the data it collects. However, it has extensive search/retrieval capabilities which maintenance personnel can use for manual analysis of the data.

6.138 The 1A Processor error analysis program does not have an active role in the processor recovery scheme. Its only function while on maintenance interrupt level is the collection of pertinent failure data. Furthermore, there is no automatic control over the configuration of the processor or over the decisions of other maintenance programs.

6.139 The 1A Processor error analysis program has two basic functions:

- Data collection
- Data retrieval.

6.140 In performing the data collection function, various types of data are collected. Data which is routinely collected includes...
(a) Maintenance interrupt data on A, B, C, D, E, and F levels. K-level interrupt data is collected from the application program if it is available.
(b) Error stop data.
(c) Maintenance interject data.
(d) Base level maintenance data.
(e) Diagnostic summaries.
(f) Phase histories.
(g) Deferred central control fault recovery failure data.
(h) Writable store audit failure reports.
(i) Application program error analysis "past history" tables if available.
(j) Application program daily plant measurements if available.

Data which may be collected on demand includes:
(a) Raw diagnostic data.
(b) Selected traffic and plant measurement reports (from the application program).
(c) Half-hour lists of out-of-service units,
(d) Repair data which consists of manually input diagnostic repair (pack replacement) data. This data is referred to as frame repair records.

B. Data Collection

6.141 The data which is routinely collected by error analysis programs falls into the following three main classes:
(a) Maintenance Level Data: Maintenance level data, includes data collected on maintenance interrupts, maintenance interjects, and also via base level maintenance reports. Maintenance interrupt data comes from the interrupt bins and consists mainly of the saved contents of internal central control registers. Such data is intended to provide information describing the general state of the processor at the time of an interrupt. Other maintenance level data comes from the fault recovery programs.
(b) Data collected on base level from diagnostic and deferred fault recovery programs: This data is collected in order to summarize the principal results obtained by running the diagnostic and deferred fault recognition programs—especially when they are run in response to a detected system trouble.
(c) Other data, including:
(1) Data from other maintenance programs such as peripheral error analysis in the application program
(2) Repair data (frame repair records) which is manually input for information purposes
(3) Data associated with phase histories, writable store audit failure report, traffic/plant measurement reports, and out-of-service unit reports.

C. Data Retrieval

6.142 The stored data is used in manual analysis. To provide information needed for manual analysis, retrieval routines are provided to search through the data base according to certain patterns of interest and according to numerical values of specified data elements. An example of the latter is a search based on the contents of saved central control operational registers. With the aid of multiple key (keyword) matching, maintenance personnel may request, for example, information retrieval in accordance with a pattern based on a combination of particular interrupt sources, system configuration, failing addresses, and unit type member number. Such searches have been made less difficult by the formation of brief descriptors of maintenance level files known as file descriptor blocks.

6.143 All forms of output which are intended for eventual maintenance personnel use are classified as reporting. Included in these outputs (reports) are:
(a) TTY output messages which give the desired information in response to a successful retrieval
(b) Error message for an invalid request

Page 183
(c) Automatic output messages which indicate that the data base is (nearly) full.

(d) Summaries of data which originated during any specified time period, containing information on the current usage of storage space.

6.144 Summaries, which only occur in response to manually input messages, provide assistance to maintenance personnel in making retrieval and editing requests. They may also be used in conjunction with follow-on retrieval requests to obtain an overall office profile or history for those maintenance troubles on which data has been collected.

7. SUPPORT DOCUMENTATION (USER-TYPE DOCUMENTS)

INTRODUCTION

7.01 This part is provided as an introductory guide to the support documentation that can be ordered for any Electronic Switching System. A brief description of the general content and uses of each type of document is given.

GENERIC PROGRAM DOCUMENTS

A. PR Documents

Program Listing (PR)

7.02 A program listing is a hard copy record of a program which should include a description of the objectives of the program, a list of the instructions used to accomplish those objectives, and definitions of all data items unique to the program. Each program consists of one or more subunits called pidents (program identifications). A pident is the smallest segment of a program (or group of instructions) which is assembled by the assembly program.

B. PK Documents

Test Access Documents (PK)

7.03 The Test Access documents provide the link between software and hardware. They are utilized when the software diagnostic and troubleshooting programs fail to isolate a problem, and manual troubleshooting must be performed. These documents identify the specific hardware monitor points associated with each diagnostic test.

7.04 Test Access documents are provided for each software diagnasable frame and are designated by the same number as the primary schematic drawing (SD) for the frame, prefixed with "PK". Each document contains a section providing descriptions and examples for reference.

7.05 The pages in these Test Access documents are arranged by mode and address. The diagnostic summary data message is used to access the correct page and bit number. Each of the 24 bits represents a different monitor point. The function name or software name is given in the first two columns and expanded, under Description, in the last column. The SD number gives the last two digits of the SD. The functional schematic/symbol numbers (FS/SYM) and lead designations then give a single point to commence troubleshooting. The name of the specific register or gate in the circuit pack schematic (CPS) is given for further information.

Raw Data Document (PK)

7.06 The PK program document specifies the type of diagnostic test performed and its expected, unprocessed, raw data as they are initially stored in memory. This document is associated with one of several maintenance programs which diagnose specific equipment units.

7.07 The PK supports the TLM for those cases where the trouble number cannot be found in the TLM or where the replacement of the equipment listed in the TLM does not correct the trouble. This may occasionally happen particularly when the fault is marginal in nature.

7.08 In the event that the trouble number does not lead to the trouble, the attendant can request, via the maintenance TTY, that the diagnostic program be reexecuted and that the test results be printed in an unprocessed form. For trunks, raw data is requested via a diagnostic from the trunk and line test panel. The raw data document helps the attendant to interpret this test data by describing the various tests and their expected results.

§ General Call Processing Related Document (PK)

7.09 There are three general call processing related documents:

(a) CIC manual (PKsA122): This manual contains a series of Compool items, macros, and pidents that are used to load orders into peripheral order buffers.
b. CIN manual (PKxA121): This manual describes the function and use of CIN macros and characterizes the network programs.

c. USER's Manual (PKxA120): This manual provides input and output specifications for routines making up translation programs.

Note: The "x" represents a 1 (for No. 1 ESS switch) or a 6 (for No. 1A ESS switch).

C. IM Documents

Input Message Manual (IM)

7.10 The Input Message Manual lists TTY messages that can be typed on the maintenance TTYs to request a system action or function. A description of the format and the use of each message, as well as cautions and expected results, are given for each message. The messages are arranged in alphabetic order, and a topical index guides the reader to the specific message to be used.

D. OM Documents

Output Message Manual (OM)

7.11 The Output Message Manual lists in alphabetic order all the system output messages printed by the TTY. This document contains a description of each message, the reason each message was issued, the actions to be taken, if any, as a result of the message having been issued, and alarm indications that should accompany the message.

E. TG-1A Documents

Translation Guide (TG)

7.12 The TG provides complete documentation of the software (translations) interface between, the telephone company assignment requirements for lines, trunks, routing, charging, measurements, etc. and, the Western Electric Company computer input requirements. Also, the document details the relationship of these input requirements to the actual feature, option or machine action desired, and the affect of the computer processes on the telephone company's administrative records maintained for the office.

F. PG-1A Documents

Parameter Guide (PG)

7.13 The PG-1A is used in the preparation of input data for the parameter data assembler (PDA). Its functional scope for parameter data is analogous to that of the Translation Guide (TG-1A) for translation data. The scope of the PG-1A includes almost all information covered in the current PA-6A001, Volume 1. The actual layout of parameter data in unduplicated call store is not within the scope of the PG-1A; this is left to the PA-6A002. The PG-1A is not designed for manual engineering of call store.

G. TLM Documents

Trouble Locating Manual (TLM)

7.14 The TLM is a maintenance document which supplements the Output Message Manual to help in locating troubles within system units. A TLM usually covers one functional unit of the system (for example, program store, call store, etc.). The TLM lists trouble numbers that are matched with numbers generated by the system from the diagnostic results. The suspected faulty package(s) (location and type) and any special procedure are specified adjacent to each trouble number. Except for TLM-1A001 on trunks and TLM-1A121 on TTYs, a TLM carries the same number as the SD of the functional unit with which it is associated. A few system units do not have an associated TLM.

APPLICATION DOCUMENTS—1A PROCESSOR

A. Diagnostic Program Applications—Description

7.15 This document describes for personnel in a telephone company switching office equipped with a 1A processor, the diagnostic program applications in terms of:

- Frame control switch requests
- Common TTY messages and options
- Interactive diagnostic options
- Diagnostic abort evaluation.

B. Generic Utility Program Applications—Description

7.16 This document describes the application of the GULP to the 1A processor. The information
provided orients thecraftsperson to the 1A processor GULP in areas not discussed in other documents.

7.17 The first part of the document is a high level discussion of the purpose, capabilities, and functions of the program. Part 2 describes the utility function verbs and provides examples to clarify the explanations. Part 3 consists of the binary layouts of the data tables and instructions used in GULP.

C. Error Analysis Program Applications—Description

7.18 This document provides information to assist telephone company personnel in the application of the 1A processor error analysis program (ERAP) and error analysis library program (ERLI). It includes a discussion of the following:

- Purpose of ERAP and ERLI
- Function and organization of ERAP
- ERAP input/output messages
- Functions of ERLI
- Preparations for running ERLI
- ERLI input/output messages
- ERAP procedures.

D. Program Listing—Software Description

7.19 This document describes the information (format and layout) contained in a program listing (PR) that is used in the Telephone Company switching offices equipped with a No. 1 or 1A processor. This section also describes the information contained in both a standard listing and a diagnostic phase program listing. The diagnostic phase program listing is different in use from other PRs and is therefore covered separately in this section.

7.20 A program listing is a software-generated hard-copy record, an output of the switching assembly program (SWAP). This section provides a discussion of this hard-copy record that contains information on the following basic topics:

- Program listing format
- Line format of instructions
- Diagnostic phase program listing

E. Diagnostic Language (DL-1)—Software Description

7.21 This document describes the diagnostic language (DL-1) and provides the following:

(a) Description of the basic structure of DL-1
(b) Description of statement format and definition of terms
(c) Detailed explanation of each DL-1 statement
(d) Alphabetical listing of the DL-1 statements.

8. GLOSSARY

8.01 A glossary of terms used in this section is described below:

Base Level (L-Level): The operational level in which the central control performs the majority of its work. All call processing is done on base level.

Buffer: A general purpose call store memory area used to store data when necessary to compensate for a difference in data flow rate.

Busy/Idle Bit: One of 16 bits in a call store word which denotes the status of an item (e.g., a link in the network map). Normally, a busy bit is equal to zero and an idle is equal to one.

Busy/Idle Word: A word in call store which contains 16 activity bits (bit positions 6 through 21) corresponding to 16 different A, B, or C links or junctions.

Client: A program that is currently requesting service from other programs or routines.

Flag: Usually a bit that, when set, indicates a request for service.

Generate Control Pulse: An instruction used to generate direct pulses to various points in the system and provide for possible responses.

Generic Program: The program controlling all system operations including diagnostic and maintenance activities.

Global: A common address to which many pldents transfer.
H-Level: High priority J-level work.

Hopper: An area of call store to store information being referred from an input/output program to a call processing program.

J-Level: An interrupt level at which clock controlled input/output programs are executed.

K-Code: A numerical definition of the address limits of a particular store.

Line Equipment Number: A number which uniquely identifies an appearance on a line switch frame.

Link Map: An area of call store containing busy/idle status of links and junctors in the network.

Local: An address which is only available from locations defined in the same pident.

Macro: A high level statement that the assembly program interprets and expands into a predefined sequence of instructions or data.

Multiline Hunt Group: A group of lines that provide the means to supply a set of special originating and terminating services on a group basis rather than an individual basis. In addition to these services, line hunting is also provided.

Multi-MAC: Refers to the ability of MACP to run concurrent jobs.

Page: One or more file store resident program sections, each of which is functionally complete, including the subroutines called by the program sections.

Paging: The operations required to bring a paged program from file store into core memory before execution can begin.

Path Memory: A part of temporary memory where enough information about a connection is stored to enable the system to reconstruct the connection.

Queue: A call store memory area used to record a waiting list of work which temporarily cannot be completed.

Register: A call store memory area used to store information required to process a particular call in progress or to record administrative or maintenance information.

Scratch Pad: A memory area allocated to the program for temporary data storage.

Task Dispenser: A program which unloads an assigned buffer and, for each buffer entry, transfers control to another program until the buffer is empty. Control is returned to the task dispenser after each buffer entry is processed.

Task Program: A program called in by a task dispenser to process a single entry in a hopper, queue, or other buffer.

T1: An activity memory bit associated with each ferrod to indicate the state of the ferrod.

T2: A control bit associated with each T1 activity bit which indicates whether the state of the T1 bit should be reported when a change occurs.

Volume Controlled Calls: Certain types of calls which are limited in number by the software at any instant of time in a particular office.

9. ABBREVIATIONS AND ACRONYMS

9.01 The following is a defined list of abbreviations and acronyms used in this section.

ADS  Auxiliary Data System
AEX  Automatic Routine Exercise
AIFR Fault Recognition Program
AIOD Automatic Identified Outward Dialing
ALIT Automatic Line Insulation Test
AMA  Automatic Message Accounting
AMAC AMA Data Accumulation Program
AMDX AMA Data Transfer Program
ANI Automatic Number Identification
AOVD Automatic Overload Control Program
APCL Attached Processor Communication Link
<table>
<thead>
<tr>
<th>API</th>
<th>Attached Processor Interface</th>
<th>COPR</th>
<th>Report and Miscellaneous Subroutines</th>
</tr>
</thead>
<tbody>
<tr>
<td>APMH</td>
<td>Attached Processor Message Handler</td>
<td>CPD</td>
<td>Central Pulse Distributor</td>
</tr>
<tr>
<td>APS</td>
<td>Attached Processor System</td>
<td>CPDB</td>
<td>Central Pulse Distributor Enable Address</td>
</tr>
<tr>
<td>APT</td>
<td>Automatic Progression Testing</td>
<td>CPFR</td>
<td>Central Pulse Distributor Fault</td>
</tr>
<tr>
<td>ASW</td>
<td>All Seems Well</td>
<td>CR</td>
<td>Call Register</td>
</tr>
<tr>
<td>ATAL</td>
<td>Audible, Disconnect, and Line Termination</td>
<td>CRFI</td>
<td>Common System Recorded Announcement</td>
</tr>
<tr>
<td>ATTT</td>
<td>Automatic Trunk Test Termination</td>
<td>CS</td>
<td>Call Store Frame Input Analysis Program</td>
</tr>
<tr>
<td>AU</td>
<td>Auxiliary Unit</td>
<td>CSDS</td>
<td>Circuit Switched Digital Capability</td>
</tr>
<tr>
<td>AUBSQ</td>
<td>Auxiliary Unit Bus Sequence</td>
<td>CSRAF</td>
<td>Common System Recorded Announcement Frame</td>
</tr>
<tr>
<td>BCD</td>
<td>Binary Coded Decimal</td>
<td>CTYP</td>
<td>Call Type</td>
</tr>
<tr>
<td>BINK</td>
<td>1024 words (BINary one K)</td>
<td>CXIX</td>
<td>Centrex Tandem Tie Line Program</td>
</tr>
<tr>
<td>CC</td>
<td>Central Control</td>
<td>CXBV</td>
<td>Busy Verify-Trunk Test Program</td>
</tr>
<tr>
<td>CCAD</td>
<td>Customer Changeable Speed Calling Program</td>
<td>CXDS</td>
<td>Disconnect for Centrex Program</td>
</tr>
<tr>
<td>CCIS</td>
<td>Common Channel Interoffice Signaling</td>
<td>CXIC</td>
<td>Incoming Digit Analysis for Centrex</td>
</tr>
<tr>
<td>CCOL</td>
<td>Chart Column</td>
<td>CXIO</td>
<td>Centrex Input/Output Scan Program</td>
</tr>
<tr>
<td>CFUP</td>
<td>Call Forwarding Usage Program</td>
<td>CXYK</td>
<td>Centrex Key Signal Director</td>
</tr>
<tr>
<td>CHRN</td>
<td>Channel Request Number</td>
<td>CXKYS</td>
<td>Centrex Attendant Line and Trunk Seizure</td>
</tr>
<tr>
<td>CIC</td>
<td>Change in Circuit</td>
<td>CXLO</td>
<td>Centrex Digit Analysis Program</td>
</tr>
<tr>
<td>CIN</td>
<td>Change in Network</td>
<td>CXOR</td>
<td>Centrex Simulated Facilities Program</td>
</tr>
<tr>
<td>CLID</td>
<td>Calling Line Identification List</td>
<td>CXSF</td>
<td>Centrex Trunk Code-Call Answer Program</td>
</tr>
<tr>
<td>CMB</td>
<td>Channel Memory Block</td>
<td>CXTA</td>
<td>Centrex Trunk Preemption Program</td>
</tr>
<tr>
<td>CNC</td>
<td>Coin Charge Register</td>
<td>CXTAP</td>
<td></td>
</tr>
<tr>
<td>CNLP</td>
<td>Centrex Console Lamp Control Program</td>
<td>CXTB</td>
<td></td>
</tr>
<tr>
<td>COCN</td>
<td>Coin Control Program</td>
<td>CXTCP</td>
<td></td>
</tr>
<tr>
<td>COIN</td>
<td>Coin Charge Program</td>
<td>CXTI</td>
<td></td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>----------------------------------</td>
<td>--------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>CXYH</td>
<td>Seize and Release Routines</td>
<td>DUC</td>
<td>Data Unit Controller</td>
</tr>
<tr>
<td>CZO</td>
<td>Coin Zone Operator</td>
<td>DUS</td>
<td>Data Unit Selector</td>
</tr>
<tr>
<td>DAC</td>
<td>Design Aid Computerised</td>
<td>ECIO</td>
<td>Executive Control Input/Output Program</td>
</tr>
<tr>
<td>DCON</td>
<td>Diagnostic Control Program</td>
<td>ECMP</td>
<td>Executive Control Main Program</td>
</tr>
<tr>
<td>DCONMAIN</td>
<td>Diagnostic Control Program (piden)</td>
<td>EMERFULL</td>
<td>Emergency Mode Control Full Configuration</td>
</tr>
<tr>
<td>DCS</td>
<td>Duplicated Call Store</td>
<td>EMERMIN</td>
<td>Emergency Mode Control Minimum</td>
</tr>
<tr>
<td>DCT</td>
<td>Digital Carrier Trunk</td>
<td>EML</td>
<td>Emergency Manual Line</td>
</tr>
<tr>
<td>DDD</td>
<td>Direct Distance Dialing</td>
<td>ERAP</td>
<td>1A Processor Error Analysis Program</td>
</tr>
<tr>
<td>DIAG</td>
<td>Diagnostic</td>
<td>ERLI</td>
<td>Error Analysis Library Program</td>
</tr>
<tr>
<td>DISC</td>
<td>Disconnect Program</td>
<td>FDIP</td>
<td>Frame Dependent Interface Program</td>
</tr>
<tr>
<td>DKAD</td>
<td>Disk Administration Program</td>
<td>FM</td>
<td>File Manager</td>
</tr>
<tr>
<td>DKADI</td>
<td>Disk Administration Interface</td>
<td>FMI</td>
<td>File Manage Interface</td>
</tr>
<tr>
<td>DMA</td>
<td>Direct Memory Access</td>
<td>FOR</td>
<td>Fault Recognition</td>
</tr>
<tr>
<td>DMAPAPPL</td>
<td>Data Mapping Control and Linking Program</td>
<td>FS</td>
<td>File Store</td>
</tr>
<tr>
<td>DMERT</td>
<td>Duplex Multi-Environment Real-Time</td>
<td>FSAP</td>
<td>File Store Administration Answer</td>
</tr>
<tr>
<td>DOC</td>
<td>Dynamic Overload Control</td>
<td>FSC</td>
<td>File Store Controller</td>
</tr>
<tr>
<td>DOCT</td>
<td>Dictionary Trouble Number Program</td>
<td>FSSP</td>
<td>File Store Administration Submit Program</td>
</tr>
<tr>
<td>DP</td>
<td>Dial Pulse</td>
<td>FSSR</td>
<td>File Store Service Routine</td>
</tr>
<tr>
<td>DRE</td>
<td>Directional Reservation of Equipment</td>
<td>GCP</td>
<td>Generate Control Pulse</td>
</tr>
<tr>
<td>DRPP</td>
<td>Diagnostic Results Post-Processing</td>
<td>GRC</td>
<td>Growth Recent Change</td>
</tr>
<tr>
<td>DSP</td>
<td>Dynamic Service Protection</td>
<td>GULP</td>
<td>Generic Utility Program</td>
</tr>
<tr>
<td>DTST</td>
<td>Dial Tone Speed Test Program</td>
<td>HMTL</td>
<td>Hotel-Motel Program</td>
</tr>
<tr>
<td>DU</td>
<td>Data Unit</td>
<td>HUC</td>
<td>Higher Unduplicated Call Store</td>
</tr>
<tr>
<td>DUAD</td>
<td>Data Unit Administration Program</td>
<td>ICB</td>
<td>Input Character Buffer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IM</td>
<td>Input Manual</td>
</tr>
<tr>
<td>I/O</td>
<td>Input-Output</td>
<td>MCCP</td>
<td>Maintenance Control Center Program</td>
</tr>
<tr>
<td>--------</td>
<td>-------------------------------</td>
<td>------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>IOCP</td>
<td>Input/Output Control Program</td>
<td>MCLM</td>
<td>System Alarms Program</td>
</tr>
<tr>
<td>IOT</td>
<td>Intraoffice Trunk</td>
<td>MCTWADMN</td>
<td>Master Control Center Administration</td>
</tr>
<tr>
<td>IOU</td>
<td>Input/Output Unit</td>
<td>MFJR</td>
<td>Multifrequency Signaling Junior Register</td>
</tr>
<tr>
<td>IOUC</td>
<td>Input/Output Unit Controller</td>
<td>MTS</td>
<td>Message Telecommunications Service</td>
</tr>
<tr>
<td>IOUS</td>
<td>Input/Output Unit Selector</td>
<td>MURL</td>
<td>Maintenance Unexpected Results List</td>
</tr>
<tr>
<td>ITTT</td>
<td>Incoming Trunk Test Termination</td>
<td>NCD</td>
<td>Noncheck Dummy</td>
</tr>
<tr>
<td>KB/S</td>
<td>Kilobits per Second</td>
<td>NETG</td>
<td>Network Growth Program</td>
</tr>
<tr>
<td>L-L</td>
<td>Line to Line</td>
<td>NMFA</td>
<td>Network Fabric Routines Program</td>
</tr>
<tr>
<td>L-T</td>
<td>Line to Trunk</td>
<td>NMFL</td>
<td>Network Maintenance Action Program</td>
</tr>
<tr>
<td>LDR</td>
<td>Loader</td>
<td>NMIN</td>
<td>Network Management Indicator Program</td>
</tr>
<tr>
<td>LENCL-4</td>
<td>Line Equipment Number Class 4</td>
<td>NMMP</td>
<td>Network Management Maintenance Program</td>
</tr>
<tr>
<td>LIBR</td>
<td>Library Control Program</td>
<td>NMMX</td>
<td>Network Matrix Exercise Program</td>
</tr>
<tr>
<td>LIBRTRP1</td>
<td>Library Control Common Traps Administrator</td>
<td>NMRF</td>
<td>Network Fault Recognition Program</td>
</tr>
<tr>
<td>LIFO</td>
<td>Last in First Out</td>
<td>NMTD</td>
<td>Transmit Dynamic Overload Control Signals</td>
</tr>
<tr>
<td>LLN</td>
<td>Line Link Network</td>
<td>NMTG</td>
<td>Network Management Program</td>
</tr>
<tr>
<td>LUC</td>
<td>Lower Unduplicated Call Store</td>
<td>NTWK</td>
<td>Network Program</td>
</tr>
<tr>
<td>LULPUTIL</td>
<td>Local Generic Utility Program</td>
<td>OFGT</td>
<td>Miscellaneous Outgoing to Switchboards and Desk Program</td>
</tr>
<tr>
<td>MAACA</td>
<td>No. 1A ESSS Scheduler</td>
<td>OFML</td>
<td>Emergency Manual Line Service Program</td>
</tr>
<tr>
<td>MAC</td>
<td>Maintenance Control</td>
<td>OFNT</td>
<td>Operator No Test Program</td>
</tr>
<tr>
<td>MACP</td>
<td>Maintenance Control Program</td>
<td>OPTR</td>
<td>Toll Switch and Recording Completing</td>
</tr>
<tr>
<td>MACR</td>
<td>Maintenance Control Peripheral Program</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MALM</td>
<td>System Alarm Program</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAUD</td>
<td>Maintenance Audit Program</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MCC</td>
<td>Master Control Console</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MCCM</td>
<td>Common Control and Monitor Program</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------------------</td>
<td>--------------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td>OM</td>
<td>Output Message</td>
<td>QSIF</td>
<td>Queue State Information Feature</td>
</tr>
<tr>
<td>OMO</td>
<td>Overtime Monitoring Operator</td>
<td>QTAL</td>
<td>Give Audible, Disconnect, and Line</td>
</tr>
<tr>
<td>OMR</td>
<td>Output Message Register</td>
<td>RACT</td>
<td>Relay Activity Bit</td>
</tr>
<tr>
<td>OOS</td>
<td>Out of Service</td>
<td>RADR</td>
<td>Receiver Attachment Delay Report</td>
</tr>
<tr>
<td>OPCL</td>
<td>Outpulsing Control Register</td>
<td>RAF</td>
<td>Recorded Announcement Frame</td>
</tr>
<tr>
<td>OR</td>
<td>Originating Register</td>
<td>RAM</td>
<td>Random Access Memory (Read-Write)</td>
</tr>
<tr>
<td>PAGS</td>
<td>Paging Supervision Program</td>
<td>RAMP</td>
<td>Recorded Announcement Machine Program</td>
</tr>
<tr>
<td>PATT</td>
<td>Processor Application Transfer Table</td>
<td>RBB</td>
<td>Rollback Block</td>
</tr>
<tr>
<td>PBX</td>
<td>Private Branch Exchange</td>
<td>RC</td>
<td>Recent Change</td>
</tr>
<tr>
<td>PC</td>
<td>Processor Configuration</td>
<td>RCSS</td>
<td>Recent Change Subsystem</td>
</tr>
<tr>
<td>PDA</td>
<td>Parameter Data Assembler</td>
<td>REX</td>
<td>Routine Exercise</td>
</tr>
<tr>
<td>PG</td>
<td>Generic Program Documentation Index</td>
<td>RI</td>
<td>Register Identification</td>
</tr>
<tr>
<td>PGID</td>
<td>Generic identification and Compatibility</td>
<td>RI-PT</td>
<td>Register Identification-Program Tag</td>
</tr>
<tr>
<td>PLUG</td>
<td>Line Termination Denied Program</td>
<td>ROH</td>
<td>Receiver Off-hook</td>
</tr>
<tr>
<td>POB</td>
<td>Peripheral Order Buffer</td>
<td>RPPS</td>
<td>Regional Parameter Processing System</td>
</tr>
<tr>
<td>PR</td>
<td>Program Listing</td>
<td>RRT</td>
<td>Routine Request Table</td>
</tr>
<tr>
<td>PRE</td>
<td>Protected Reservation of Equipment</td>
<td>RVRT</td>
<td>Reverting Call Program</td>
</tr>
<tr>
<td>PS</td>
<td>Program Store</td>
<td>SACT</td>
<td>Customer Program for Growth</td>
</tr>
<tr>
<td>PSDC</td>
<td>Public Switched Digital Capability</td>
<td>SADT</td>
<td>System Audit Program</td>
</tr>
<tr>
<td>PT</td>
<td>Program Tag</td>
<td>SAWS</td>
<td>Writable Store Audits</td>
</tr>
<tr>
<td>PTW</td>
<td>Primary Translation Word</td>
<td>SCFR</td>
<td>Scanner Fault Recognition Program</td>
</tr>
<tr>
<td>PUAB</td>
<td>Peripheral Unit Address Bus</td>
<td>SI</td>
<td>System Initialization</td>
</tr>
<tr>
<td>QAPR</td>
<td>Queue and Administration Processing</td>
<td>SIRE</td>
<td>System Interrupt Recovery Program</td>
</tr>
<tr>
<td>QCIA</td>
<td>Customer Interface and Special Auditing</td>
<td>SR</td>
<td>System Reinitialization</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------------------------------</td>
<td>--------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>SKTT</td>
<td>Station Ringer and TOUCH-TONE Program</td>
<td>TTKW</td>
<td>Teletypewriter Work Register Program</td>
</tr>
<tr>
<td>SUPERV</td>
<td>Supervision Modernization</td>
<td>TTYM</td>
<td>Teletypewriter Translation Input/Output</td>
</tr>
<tr>
<td>SWAP</td>
<td>Switching Assembly Program</td>
<td>TUC</td>
<td>Tape Unit Controller</td>
</tr>
<tr>
<td>SYPI</td>
<td>System Performance Indicator Program</td>
<td>TVNDX</td>
<td>Transfer Vector Index</td>
</tr>
<tr>
<td>SYUP</td>
<td>System Update Program</td>
<td>TWR</td>
<td>Teletypewriter Work Register</td>
</tr>
<tr>
<td>T-T</td>
<td>Trunk to Trunk</td>
<td>TXFR</td>
<td>Call Forwarding Program</td>
</tr>
<tr>
<td>TAND</td>
<td>Tandem Connection Program</td>
<td>VFHC</td>
<td>Verification of H and C Register</td>
</tr>
<tr>
<td>TBR</td>
<td>Teletypewriter Buffer Register</td>
<td>WAIT</td>
<td>Call Waiting Program</td>
</tr>
<tr>
<td>THTF</td>
<td>Through Balance Test Facility</td>
<td>WPADAPL2</td>
<td>Write Protect Administration Application</td>
</tr>
<tr>
<td>TCC</td>
<td>Trunk Class Code</td>
<td>WPADCOMM</td>
<td>Write Protect Administration Common</td>
</tr>
<tr>
<td>TG</td>
<td>Translation Guide</td>
<td>WPACTRL</td>
<td>Write Protect Administration Control</td>
</tr>
<tr>
<td>TGC</td>
<td>Trunk Group Control</td>
<td>WQUE</td>
<td>Queue Administration Program</td>
</tr>
<tr>
<td>TLM</td>
<td>Trouble Locating Manual</td>
<td>WRDN</td>
<td>Word Number</td>
</tr>
<tr>
<td>TLN</td>
<td>Trunk Line Network</td>
<td>YAH</td>
<td>Seize and Release Routines</td>
</tr>
<tr>
<td>TODA</td>
<td>Ringing and Tone Plant Diagnostic</td>
<td>YCCK</td>
<td>Register Link Routine</td>
</tr>
<tr>
<td>TOMK</td>
<td>Ringing and Tone Plant Monitor</td>
<td>YCLK</td>
<td>Register Linking Routine</td>
</tr>
<tr>
<td>TOPR</td>
<td>Toll Operator Signaling Program</td>
<td>YFDS</td>
<td>Scan of Single Master Scanner Point</td>
</tr>
<tr>
<td>TRCE</td>
<td>Call Trace Program</td>
<td>YFTO</td>
<td>Incoming Trunk to Busy Overflow</td>
</tr>
<tr>
<td>TRNS</td>
<td>Transition State</td>
<td>YMRG</td>
<td>Miscellaneous Register Subroutines</td>
</tr>
<tr>
<td>TSAH</td>
<td>Trunk Seizure and Answer Hopper</td>
<td>YTTO</td>
<td>Originating Line to Busy Overflow</td>
</tr>
<tr>
<td>TSPS</td>
<td>Traffic Service Position System</td>
<td>ZERO</td>
<td>Call Store Zeroing Program</td>
</tr>
<tr>
<td>TTIA</td>
<td>Teletypewriter Input Messages Directory</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TTOX</td>
<td>Teletypewriter Output Program</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TPP</td>
<td>Teletypewriter Output Phases Program</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Page 192 of 192 Pages*
**Table Name**

TOPS Holiday Table

**Overview**

Table HOLDAY is used by the operating company to list the holidays and the holiday treatment associated with each holiday for each schedule.

Public holidays must be identified when they involve rate treatment different from that which are normally given on the day of the week they occur. For example, on a particular rate schedule, Christmas is treated as a Sunday for rating purposes regardless of the day of the week in which it falls.

**International Traffic Operator Position System (ITOPS)**

Calls requiring operator assistance result in a charge to the customer that is calculated by a downstream process using the call information (call type, destination, answer time, duration of call). There are cases, however, where the charges on a call must be quoted to the customer. This is true for the following call types:

- Hotel Calls
- Coin Calls
- Calls Requiring Time-and-Charges Quote

In all three cases, the ITOPS system calculates the charges on the call. For hotel calls, these charges are reported to the Hotel Billing Center (HOBIC) for quoting; in the latter two cases, the operator quotes the charges.

**Caribbean Expansion Plan (CEP)**

The dialing plan for the Caribbean islands is based on the North American plan (NPA−NXX−XXXX). The islands have numbering plan area (NPA) 809, with each island having one or more unique NXX associated with it.

**CEP ITOPS Rating Zones**

The following terminology is used within this document to define the different CEP ITOPS rating zones:

- **Local**  Calls completed within the same rate zone as the calling customer (calls within the same NXX are always considered to be in the local rate zone).
- **Domestic**  Calls completed within NPA 809 but to an NXX in a different rate zone as the calling customer.
- **North American**  Calls completed from NPA 809 to any destination based on the NPA−NXX dialing plan (other than domestic calls).
- **International (or Overseas)**  Calls completed from NPA 809 to a foreign country not based on the NPA−NXX dialing plan (outside of World Zone 1).
• **CEP: Call Origination Time and Day** Both the date and the time of day can have an effect on the charge calculation on a call.

First, the date is checked to see if it is a holiday by verifying if the date falls on one of the defined holidays in table HOLDAY. If the date is found to be a holiday, table HOLTRT is accessed to see if this holiday affects the rate schedule for this call. If a value is not found in this table, the holiday is deemed to not affect charges applicable to the rate schedule.

If a holiday is found to apply to the rate schedule, the treatment specifies that the call is treated as if it originated on Saturday or Sunday (as specified).

**Functional Description of Table HOLDAY**

Table HOLDAY is used to list the name and date of the holidays to receive holiday treatment.

**Oversea Operator Center (OOC)**

Table HOLDAY gives the holiday name corresponding to the month and day of the year.

For OOC, table HOLDAY is not used but is retained for future enhancements. It is recommended that this table be left blank until further notice.

**Datafill Sequence**

There is no requirement to datafill other tables prior to table HOLDAY.

**Datafill**

The following table describes datafill for table HOLDAY:

<table>
<thead>
<tr>
<th>Field</th>
<th>Subfield</th>
<th>Entry</th>
<th>Explanation and Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOLNAME</td>
<td></td>
<td>Alphanumeric</td>
<td>Holiday Name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(1 to 10 characters)</td>
<td>Enter the name assigned to the holiday.</td>
</tr>
<tr>
<td>MONTH</td>
<td></td>
<td>JAN, FEB, MAR, APR,</td>
<td>Month of Holiday</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAY, JUN, JUL, AUG,</td>
<td>Enter the month of the holiday.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SEP, OCT, NOV, or DEC</td>
<td></td>
</tr>
<tr>
<td>DAY</td>
<td></td>
<td>1 to 31</td>
<td>Day of the Holiday</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Enter the day of the holiday.</td>
</tr>
</tbody>
</table>

**Datafill Example**

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

<table>
<thead>
<tr>
<th>HOLNAME</th>
<th>MONTH</th>
<th>DAY</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEWYEAR</td>
<td>JAN</td>
<td>01</td>
</tr>
<tr>
<td>CHRISTMAS</td>
<td>DEC</td>
<td>25</td>
</tr>
</tbody>
</table>
Table Name

TOPS Holiday Treatment Table

Functional Description of Table HOLTRT

Table HOLTRT specifies what holiday treatment, if any, is to be given to the holidays listed in table HOLDAY, for each schedule; that is, each holiday can be treated differently for each schedule.

Feature V0178 (TOPS Mass Table Control) permits data changes in table HOLTRT to be mass-table-controlled. In other words, the feature permits the simultaneous activation of data changes in the table by entering the data changes for the table into table HOLTRTI (TOPS Holiday Treatment Inactive), and then, when all the required changes are entered, swap the contents of table HOLTRT with table HOLTRTI.

For further information on Feature V0178, refer to table CHARGEI (TOPS Charge Inactive Table).

Overseas Operator Center (OOC)

Table HOLTRT provides the mapping of the holiday name to the corresponding holiday treatment for each schedules.

For OOC, table HOLTRT is not used but is retained for future enhancements. It is recommended that this table be left blank until further notice.

For related information, refer to table HOLDAY.

Datafill

The following table describes datafill for table HOLTRT:

<table>
<thead>
<tr>
<th>Field</th>
<th>Subfield</th>
<th>Entry</th>
<th>Explanation and Action</th>
</tr>
</thead>
</table>
| HOLSCH  | See Subfields |                     | Holiday Treatment Key
|         |          |                      | This field consists of subfields HOLNAME and SCHNAME.                                  |
| HOLNAME | Alphanumeric | (up to 16 characters) | Holiday Name
|         |          |                      | Enter the holiday name as previously defined table HOLDAY.                             |
| SCHNAME | Alphanumeric | (1 to 16 characters) | Schedule Name
|         |          |                      | Enter the schedule name. This name must be known to table SCHED.                      |
| HOLTRT  | SAT, SPL, SUN, or NON |                 | Holiday Treatment
|         |          |                      | Enter the holiday treatment.                                                          |
|         |          |                      | Enter "SAT" (Saturday), "SPL" (special), or "SUN" (Sunday).                           |
Saturday or Sunday means the holiday is treated as Saturday or Sunday; that is, the Saturday or Sunday rate break sets are used.

Special means the treatment prescribed for the actual day on which the holiday falls is used. However, if the selected rate table is “no discount,” discount 1 rates are to be used instead.

Datafill Example

The first example shows datafill for North American TOPS.

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

<table>
<thead>
<tr>
<th>HOLSCH</th>
<th>HOLTRT</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEWYEAR ONTQUE</td>
<td>SUN</td>
</tr>
<tr>
<td>NEWYEAR CANUSA</td>
<td>SUN</td>
</tr>
</tbody>
</table>

The second example shows datafill for the Caribbean Expansion Plan – International TOPS.

<table>
<thead>
<tr>
<th>HOLSCH</th>
<th>HOLTRT</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEWYEARS USA</td>
<td>SAT</td>
</tr>
<tr>
<td>NEWYEARS CANADA</td>
<td>SUN</td>
</tr>
<tr>
<td>XMAS USA</td>
<td>SUN</td>
</tr>
<tr>
<td>XMAS CANADA</td>
<td>SUN</td>
</tr>
<tr>
<td>XMAS UK</td>
<td>SUN</td>
</tr>
</tbody>
</table>
Radio Shack PRO–2042 455 kHz I.F. Filter Modifications

Overview

The "selectivity" parameter of a communications receiver or radio scanner is its ability to reject any adjacent channel interference. This parameter is mainly determined by the bandwidth of the last Intermediate Frequency (IF) filter in the receiver's RF demodulating chain. If you have ever tried to tune a scanner to a weak signal, only to have it squashed by nearby higher–powered transmission, then you understand the need for improved selectivity in a receiver.

In Radio Shack scanners, and most other conventional narrowband FM receivers, the last IF is 455 kHz. The service manual for the Radio Shack PRO–2006 scanner, which is arguably the "best" of the GRE–based Radio Shack scanners, lists the narrowband FM selectivity at −6 dB rejection +/− 9 kHz from the IF's center. These stock filter specifications are good, but we can easily improve them slightly by replacing the last 455 kHz IF filter with something a little "tighter" to improve the attenuation skirt.

In the Radio Shack PRO–2042 scanner, which we'll be using for this example, the main narrowband FM IF demodulation IC is based around the Toko TK10420, which is a slight improved version of the Motorola MC3357. The datasheet for the MC3357 goes into much more detail on the technical operations of the IF mixing, amplifying, filtering, and demodulation. You'll want to study the MC3357's datasheet carefully, but we're only interested in improving the final IF filtering aspect right now.

The PRO–2042's 455 kHz IF filter is most likely a Murata CFUM455D model. I say "likely" because I don't have the actual part number for that particular scanner, and Murata doesn't fully label their line of miniature IF filters. It doesn't really matter though, as the only thing we are interested in is the "D" in the part number. This corresponds to a filter with a −6 dB bandwidth of 10 kHz, which closely matches that of the specification listed in the PRO–2006 service manual, which is very similar to the PRO–2042, circuit–wise.

The filter modification is quite simple, just replace the 10 kHz wide bandwidth "D" model with a slightly narrower "E" or "F" model, which are 7.5 kHz and 6 kHz wide, respectively. This will help to narrow the final the IF bandwidth slightly, reducing any adjacent channel interference, but can cause overmodulated or wideband FM signals to become "clipped" or distorted. This may or may not be a drawback, you'll have to experiment for yourself. You may even wish to toggle the two filters in–and–out of the IF chain using a mechanical relay or PIN diodes.

Sourcing new 455 kHz IF filters is getting to be quite difficult nowadays. Thankfully, thrift stores are full of useful RF parts in the form of old 49 MHz cordless phones and baby monitors. Buy everyone you see, take them all apart, and study the receiver's IF chain for a little rectangle or square filter next to a 16–pin DIP with a label similar to "55E" or "55F." You'll want to study the filter's pin–out carefully, as different models will have different pin–outs. All the filter's will have the same overall concepts. They have an input, output, and ground, so tweaking a filter with the wrong pin–out to work is always possible. You may also have to do some impedance matching in extreme cases, but all the different manufactures seem to use the same general filter impedance of around 1,500 to 3,000 ohms.
Another useful and little-known trick to help improve your receiver's performance is to increase the value of the damping resistor used in the IF receiver's chip external quadrature tank circuit. This circuit is used to provide a 90° phase shift to the 455 kHz IF to recover the final audio via mixing. As this resistor's value is lowered, separation and bandwidth are increased but the recovered audio is also decreased. The stock value is usually around 33,000 ohms and we'll be increasing it to around 47,000 ohms.

**Pictures & Construction**

Example IF filters in a commercial VHF two-way radio.

The radio's receiver chain is also based around the Toko 10420 and uses two Murata CFU455E 455 kHz IF filters in series to help improve receiver selectivity.
Example IF filter in a 49 MHz narrowband FM baby monitor.

This receiver chain is based around a Motorola MC3359.

The circuit board label above the IF filter says "CFW455E," which is a Murata part number, but the filter is a Kyocera model with similar specifications and a slightly different pin−out.
Example IF filters in a Radio Shack PRO−2051 "Trunk Tracker" scanner.

Because the PRO−2051 has to continuously monitor the data channel in a trunked radio system, this scanner actually has two receivers in it. The IF filter on the left is the standard "D" model, and the filter on the right is the "H" model. The "H" model has a very narrow bandwidth, 3 kHz or so. This is useful in low−speed data applications, but not very useful in receiving wider bandwidth FM audio transmissions.

Also note that this scanner uses a 450 kHz last IF frequency instead of the common 455 kHz.
Murata CFM455E high-performance mechanical 455 kHz IF filter.

If you want to improve your scanner's selectivity even more, try to track down IF filters like the one shown above. These basically have the same bandwidth as the Murata "E" model filters, but the filter's "skirt," or attenuation factor, is much greater away from the center operating frequency. These mechanical filters have at least 60 dB of attenuation only +/- 16 kHz from their center, compared to only 40 dB of attenuation +/- 20 kHz from the center of the stock "D" model.
Internal view of a PRO−2042 scanner showing the last IF strip.

The scanner’s second IF is 48.5 MHz and is sent to the TK10420 IF chip to be mixed with a 48.045 crystal−based local oscillator, which produces the final 455 kHz IF. The crystal on the left marked "XF1" is the 48.5 second IF resolution crystal filter, and the crystal next to the TK10420 is the 48.045 MHz LO crystal.

The large black rectangle marked "55D" is the narrowband FM IF filter, the yellow−orange one is a narrow bandwidth model for further filtering during AM demodulation.
Bottom view of the PRO−2042 main circuit board showing the pin locations of the TK10420.

The red arrow marks the TK10420's pin 1, while the bottom−left red dot marks pin 8.

The upper−left red dot is then pin 9 and the upper−right red dot is pin 16.

The blue arrow points out the stock 33,000 ohm damping resistor in the quadrature tank circuit.

Note the narrowband FM IF filter has been desoldered and removed. This particular IF filter had only three pins, input, output, and ground. Some IF systems, such as most Radio Shack scanners, use a "floating" ground in their final IF system, so the IF filter's "ground" pin may not be at actual ground potential.
Installation of a new Murata mechanical 455 kHz IF filter.

The filter is resting on a piece of double-sided tape to isolate it from ground.

The three ground pins on the filter are all connected together and soldered to the IF's floating ground.

The input and output pins on the IF filter are reciprocal, so you can mount the filter however is most convenient.
Changing the quadrature coil's damping resistor from 33,000 ohms to 47,000 ohms.

This helps to increase the "Q" of the tank circuit used to provide the 90° phase shift in the FM demodulator circuit. This then increases the output audio level from the TK10420 slightly, improving the scanner's overall audio response.
# PIEZO FILTERS

## MULTI-ELEMENT FILTERS, RESIN MOLDED HIGHLY SELECTIVE

The CFWS 455 series of ceramic filters are 6-element devices connected in ladder form. These compact, highly selective filters are recommended for use in applications ranging from two-way radio to auxiliary filters in high class transceivers. (Also available in 450kHz version.)

## SPECIFICATIONS

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Nominal Center Frequency (kHz)</th>
<th>6dB Bandwidth (kHz) min.</th>
<th>6dB Bandwidth (kHz) max.</th>
<th>Attenuation 455x10kHz (dB) min.</th>
<th>Ripple (dB) max. kHz</th>
<th>Insertion Loss (dB) max.</th>
<th>Input/Output Impedance (Ohms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>*CFWS455B</td>
<td>455</td>
<td>±15</td>
<td>±30</td>
<td>35</td>
<td>3 (455 ± 10)</td>
<td>4</td>
<td>1500</td>
</tr>
<tr>
<td>*CFWS455C</td>
<td>455</td>
<td>±12.5</td>
<td>±24</td>
<td>35</td>
<td>3 (455 ± 8)</td>
<td>4</td>
<td>1500</td>
</tr>
<tr>
<td>*CFWS455D</td>
<td>455</td>
<td>±10</td>
<td>±20</td>
<td>35</td>
<td>3 (455 ± 7)</td>
<td>4</td>
<td>1500</td>
</tr>
<tr>
<td>*CFWS455E</td>
<td>455</td>
<td>±7.5</td>
<td>±15</td>
<td>35</td>
<td>3 (455 ± 5.6)</td>
<td>6</td>
<td>1500</td>
</tr>
<tr>
<td>*CFWS455F</td>
<td>455</td>
<td>±6</td>
<td>±12.5</td>
<td>35</td>
<td>3 (455 ± 4)</td>
<td>6</td>
<td>2000</td>
</tr>
<tr>
<td>*CFWS455G</td>
<td>455</td>
<td>±4.5</td>
<td>±10</td>
<td>35</td>
<td>2 (455 ± 3)</td>
<td>6</td>
<td>2000</td>
</tr>
<tr>
<td>CFWS455HT</td>
<td>455</td>
<td>±3</td>
<td>±9</td>
<td>60</td>
<td>2 (455 ± 2)</td>
<td>6</td>
<td>2000</td>
</tr>
<tr>
<td>CFWS455IT</td>
<td>455</td>
<td>±2</td>
<td>±7.5</td>
<td>60</td>
<td>2 (455 ± 1.5)</td>
<td>6</td>
<td>2000</td>
</tr>
</tbody>
</table>

## DIMENSIONS: mm

![DIMENSIONS](image)

*Available as standard through authorized Murata Electronics Distributors.

*Note: For safety purposes, connect the output of filters to the IF amplifier through a DC blocking capacitor. Avoid applying a direct current to the output of ceramic filters.

## CIRCUIT

![CIRCUIT](image)

## CHARACTERISTICS

![CHARACTERISTICS](image)
# PIEZO FILTERS
## MULTI-ELEMENT, ULTRA-MINIATURE

The CFUM 455 and CFWM 455 lines of ceramic filters are miniaturized versions of the CFU/CFWS lines. These ultra-miniature versions consume approximately 40% less volume while still offering the same high performance filter characteristics available with the CFU/CFWS lines. (Also available in 450kHz version.)

## CFUM 455kHz

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Nominal Center Frequency (kHz)</th>
<th>6dB Bandwidth (kHz) min.</th>
<th>6dB Bandwidth (kHz) max.</th>
<th>40dB Bandwidth (kHz) min.</th>
<th>40dB Bandwidth (kHz) max.</th>
<th>Insertion Loss (dB) max.</th>
<th>Input/Output Impedance (ohms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>*CFUM455B</td>
<td>455</td>
<td>±15</td>
<td>±30</td>
<td>27</td>
<td>27</td>
<td>4</td>
<td>1500</td>
</tr>
<tr>
<td>*CFUM455C</td>
<td>455</td>
<td>±12.5</td>
<td>±24</td>
<td>27</td>
<td>27</td>
<td>4</td>
<td>1500</td>
</tr>
<tr>
<td>*CFUM455D</td>
<td>455</td>
<td>±10</td>
<td>±20</td>
<td>27</td>
<td>27</td>
<td>4</td>
<td>1500</td>
</tr>
<tr>
<td>*CFUM455E</td>
<td>455</td>
<td>±7.5</td>
<td>±15</td>
<td>27</td>
<td>27</td>
<td>6</td>
<td>1500</td>
</tr>
<tr>
<td>*CFUM455F</td>
<td>455</td>
<td>±6</td>
<td>±12.5</td>
<td>27</td>
<td>27</td>
<td>6</td>
<td>2000</td>
</tr>
<tr>
<td>*CFUM455G</td>
<td>455</td>
<td>±4.5</td>
<td>±10</td>
<td>25</td>
<td>25</td>
<td>6</td>
<td>2000</td>
</tr>
<tr>
<td>*CFUM455H</td>
<td>455</td>
<td>±3</td>
<td>±9</td>
<td>25</td>
<td>25</td>
<td>6</td>
<td>2000</td>
</tr>
<tr>
<td>*CFUM455I</td>
<td>455</td>
<td>±2</td>
<td>±7.5</td>
<td>36</td>
<td>36</td>
<td>6</td>
<td>2000</td>
</tr>
</tbody>
</table>

## CFWM 455kHz

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Nominal Center Frequency (kHz)</th>
<th>6dB Bandwidth (kHz) min.</th>
<th>6dB Bandwidth (kHz) max.</th>
<th>40dB Bandwidth (kHz) min.</th>
<th>40dB Bandwidth (kHz) max.</th>
<th>Insertion Loss (dB) max.</th>
<th>Input/Output Impedance (ohms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>*CFWM455B</td>
<td>455</td>
<td>±15</td>
<td>±30</td>
<td>27</td>
<td>27</td>
<td>4</td>
<td>1500</td>
</tr>
<tr>
<td>*CFWM455C</td>
<td>455</td>
<td>±12.5</td>
<td>±24</td>
<td>27</td>
<td>27</td>
<td>4</td>
<td>1500</td>
</tr>
<tr>
<td>*CFWM455D</td>
<td>455</td>
<td>±10</td>
<td>±20</td>
<td>27</td>
<td>27</td>
<td>4</td>
<td>1500</td>
</tr>
<tr>
<td>*CFWM455E</td>
<td>455</td>
<td>±7.5</td>
<td>±15</td>
<td>27</td>
<td>27</td>
<td>4</td>
<td>1500</td>
</tr>
<tr>
<td>*CFWM455F</td>
<td>455</td>
<td>±6</td>
<td>±12.5</td>
<td>27</td>
<td>27</td>
<td>6</td>
<td>1500</td>
</tr>
<tr>
<td>*CFWM455G</td>
<td>455</td>
<td>±4.5</td>
<td>±10</td>
<td>25</td>
<td>25</td>
<td>6</td>
<td>2000</td>
</tr>
<tr>
<td>*CFWM455H</td>
<td>455</td>
<td>±3</td>
<td>±9</td>
<td>25</td>
<td>25</td>
<td>6</td>
<td>2000</td>
</tr>
<tr>
<td>*CFWM455I</td>
<td>455</td>
<td>±2</td>
<td>±7.5</td>
<td>36</td>
<td>36</td>
<td>6</td>
<td>2000</td>
</tr>
</tbody>
</table>

*CFWM455C series filters are 6-element ceramic filters and ultraminiature versions of CFUM455C series.

## DIMENSIONS: mm

![Diagram](image1)

## CIRCUIT

![Circuit Diagram](image2)

## CHARACTERISTICS

![Characteristics Graph](image3)

*Available as standard through authorized Murata Electronics Distributors.

*Note: For safety purposes, connect the output of filters to the IF amplifier through a DC blocking capacitor. Avoid applying a direct current to the output of ceramic filters.
PIEZO FILTERS
MULTI-ELEMENT
HIGH PERFORMANCE

CFM/CFJ/CFR/CFS/CFL 455kHz

The following lines of filters are high performance devices that achieve ultimate stopband attenuation through the use of multiple piezoelectric elements connected in ladder form. A few of the recommended applications for these filters include high class receivers, SSB communications equipment, pocket pagers and mobile radios.

CFM 455 9 Ceramic Elements
CFJ 455K 11 Ceramic Elements
CFR 455 11 Elements Filters
CFS 455 15 Element Filters
CFL 455 9 Element Filters (GDT Improved)

(NOT available in 450kHz.)

SPECIFICATIONS

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Nominal Center Frequency (MHz)</th>
<th>3dB Bandwidth (kHz) min.</th>
<th>3dB Bandwidth (kHz) max.</th>
<th>Ripple (dB) max.</th>
<th>Bandwidth (kHz) max.</th>
<th>At (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFM455A</td>
<td>455</td>
<td>±13</td>
<td>±17/5</td>
<td>3</td>
<td>±30</td>
<td></td>
</tr>
<tr>
<td>CFM455B</td>
<td>455</td>
<td>±10</td>
<td>±15</td>
<td>3</td>
<td>±29</td>
<td></td>
</tr>
<tr>
<td>CFM455C</td>
<td>455</td>
<td>±9</td>
<td>±13</td>
<td>3</td>
<td>±23</td>
<td>60</td>
</tr>
<tr>
<td>CFM455D</td>
<td>455</td>
<td>±7</td>
<td>±10</td>
<td>3</td>
<td>±20</td>
<td>70</td>
</tr>
<tr>
<td>CFM455E</td>
<td>455</td>
<td>±5.5</td>
<td>±8</td>
<td>3</td>
<td>±16</td>
<td></td>
</tr>
<tr>
<td>CFM455F</td>
<td>455</td>
<td>±4.2</td>
<td>±6</td>
<td>3</td>
<td>±12</td>
<td></td>
</tr>
<tr>
<td>CFM455G</td>
<td>455</td>
<td>±4</td>
<td>±3</td>
<td>3</td>
<td>±10</td>
<td></td>
</tr>
<tr>
<td>CFM455H</td>
<td>455</td>
<td>—</td>
<td>±3</td>
<td>3</td>
<td>±7.5</td>
<td></td>
</tr>
<tr>
<td>CFR455A</td>
<td>455</td>
<td>±13</td>
<td>±17/5</td>
<td>3</td>
<td>±30</td>
<td></td>
</tr>
<tr>
<td>CFR455B</td>
<td>455</td>
<td>±10</td>
<td>±15</td>
<td>3</td>
<td>±29</td>
<td></td>
</tr>
<tr>
<td>CFR455C</td>
<td>455</td>
<td>±9</td>
<td>±13</td>
<td>3</td>
<td>±23</td>
<td></td>
</tr>
<tr>
<td>CFR455D</td>
<td>455</td>
<td>±7</td>
<td>±10</td>
<td>3</td>
<td>±20</td>
<td></td>
</tr>
<tr>
<td>CFR455E</td>
<td>455</td>
<td>±5.5</td>
<td>±8</td>
<td>3</td>
<td>±16</td>
<td></td>
</tr>
<tr>
<td>CFR455F</td>
<td>455</td>
<td>±4.2</td>
<td>±6</td>
<td>3</td>
<td>±12</td>
<td></td>
</tr>
<tr>
<td>CFR455G</td>
<td>455</td>
<td>±4</td>
<td>±3</td>
<td>3</td>
<td>±10</td>
<td></td>
</tr>
<tr>
<td>CFR455H</td>
<td>455</td>
<td>—</td>
<td>±3</td>
<td>3</td>
<td>±7.5</td>
<td></td>
</tr>
<tr>
<td>CJF455K</td>
<td>455</td>
<td>±1.5</td>
<td>±3</td>
<td>3</td>
<td>±4.5</td>
<td></td>
</tr>
<tr>
<td>CJF455K1</td>
<td>455</td>
<td>±1.1 - ±1.3</td>
<td>±4.5 (Total)</td>
<td>2</td>
<td>±6</td>
<td></td>
</tr>
<tr>
<td>CJF455K2</td>
<td>455</td>
<td>±1.0 (Total)</td>
<td>1.5</td>
<td>3</td>
<td>±0 (Total)</td>
<td></td>
</tr>
<tr>
<td>CFS456A</td>
<td>456</td>
<td>±13</td>
<td>±17/5</td>
<td>3</td>
<td>±30</td>
<td></td>
</tr>
<tr>
<td>CFS456B</td>
<td>456</td>
<td>±10</td>
<td>±15</td>
<td>3</td>
<td>±29</td>
<td></td>
</tr>
<tr>
<td>CFS456C</td>
<td>456</td>
<td>±9</td>
<td>±13</td>
<td>3</td>
<td>±23</td>
<td></td>
</tr>
<tr>
<td>CFS456D</td>
<td>456</td>
<td>±7</td>
<td>±10</td>
<td>3</td>
<td>±20</td>
<td></td>
</tr>
<tr>
<td>CFS456E</td>
<td>456</td>
<td>±5.5</td>
<td>±8</td>
<td>3</td>
<td>±16</td>
<td></td>
</tr>
<tr>
<td>CFS456G</td>
<td>456</td>
<td>±4.2</td>
<td>±6</td>
<td>3</td>
<td>±12</td>
<td></td>
</tr>
<tr>
<td>CFS456H</td>
<td>456</td>
<td>—</td>
<td>±3</td>
<td>3</td>
<td>±7.5</td>
<td></td>
</tr>
<tr>
<td>CFS456J</td>
<td>456</td>
<td>±1</td>
<td>±3</td>
<td>3</td>
<td>±5</td>
<td></td>
</tr>
<tr>
<td>CFS456K</td>
<td>456</td>
<td>±1</td>
<td>±3</td>
<td>3</td>
<td>±5</td>
<td></td>
</tr>
<tr>
<td>CLF455G</td>
<td>455</td>
<td>±13.5</td>
<td>±13.5</td>
<td>0.5</td>
<td>±27.5</td>
<td>60</td>
</tr>
<tr>
<td>CLF455K1</td>
<td>455</td>
<td>±12.0</td>
<td>±12.0</td>
<td>0.5</td>
<td>±25.5</td>
<td>60</td>
</tr>
<tr>
<td>CLF455K2</td>
<td>455</td>
<td>±10.0</td>
<td>±10.0</td>
<td>0.5</td>
<td>±21.0</td>
<td>60</td>
</tr>
<tr>
<td>CLF455K3</td>
<td>455</td>
<td>±5.0</td>
<td>±7.0</td>
<td>0.5</td>
<td>±18</td>
<td>60</td>
</tr>
</tbody>
</table>

DIMENSIONS: mm

CFM 455

CFR 455

CFJ 455K

CFS 455

CFL 455G

Note: For safety purposes, connect the output of filters to the F-amplifier through a DC blocking capacitor. Avoid applying a direct current to the output of ceramic filters.

438

53
He Can’t Put Two Words Together Without It!
End of Issue #60

Any Questions?

Editorial and Rants

Yeah... Like this fucker isn’t just a puppet.
"Mr. Obama uses them for everyday routine announcements, and even for the opening statement at his news conference.

He used them during a visit to a Caterpillar plant in Peoria, Ill. He used them to make brief remarks at the opening of his 'fiscal responsibility summit.' He used them during a visit to the Interior Department to discuss endangered species, even as he recalled a visit to some national parks as an 11-year-old. ‘That was an experience I will never forget,’ he said, reading from the teleprompter.”

(www.nytimes.com/2009/03/06/us/politics/06web−baker.html)

"After the teleprompter malfunctioned a few times last summer and Obama delivered some less-than-soaring speeches, reports surfaced that he was training to wean himself off of the device while on vacation in Hawaii. But no such luck.”

...

"In a break from his routine, Obama did not use a teleprompter during his pre-Inauguration speech at a factory in Bedford Heights, Ohio – and his delivery seemed to suffer. He paused too long at parts. He accentuated the wrong words. And overall he sounded hesitant and halting as he spoke from the prepared remarks on the podium.”


How about doing your little investigations before the presidential election, guys?
Since when do you need “permission” to hold a party? Did those dirty spics get permission (and insurance) for their pro–illegal immigration rallies?

Cape "Tea Party" Canceled; City Fears Too Many Attendees

March 27, 2009 – From: www.winknews.com

CAPE CORAL, Fla. – A tea party to protest government spending and taxing is canceled. Canceled by the government.

Why? They feel too many people could show–up.

Lynn Rosko planned to hold a tax payer tea party at Jaycee Park in Cape Coral on April 1st. The idea was announced at a Cape Coral City Council meeting, then an e–mail blast by the Republican Party and it was mentioned in the local media.

With all of that attention, the City of Cape Coral felt there could be more than 500 people attending the tea party.

Therefore Rosko needed to get a permit and insurance for the event. Rosko says she's not willing to get insurance and accept liability for something that a stranger could do. Rosko told WINK News, "I have rescinded any organizing or supervision or what ever you want to call it over this tea party on April 1st."

WINK News spoke to the director of parks for Cape Coral. He says that even now if Rosko is willing to get insurance for the event he'll likely re–authorize it.

For now Rosko’s event is canceled, she's encouraging people to attend the April 15th Tax Payer Tea Party in Centennial Park in Fort Myers.

Pro–Illegal Immigration Rally in Los Angeles

Looks like non–U.S. citizens don’t have to obey our own laws – while in this country!
"Separation of Church and State!"

"Obama's not a Muslim!"

"Race doesn't matter!"

Wait... Is that nothing but silence I hear? LOL!

Obama Gets List of Top Muslim Americans

March 27, 2009 – From: www.denverpost.com

CHICAGO — In a bid to get more Muslim Americans working in the Obama administration, a book with resumes of 45 of the nation's most qualified — Ivy League grads, Fortune 500 executives and public servants, all carefully vetted — has been submitted to the White House.

The effort, driven by community leaders and others, including U.S. Rep. Keith Ellison, D–Minn., was bumped up two weeks because White House officials heard about the venture, said J. Saleh Williams, program coordinator for the Congressional Muslim Staffers Association, who sifted through more than 300 names.

"It was mostly under the radar," Williams said. "We thought it would put (the president) in a precarious position. We didn't know how closely he wanted to appear to be working with the Muslim American community."

Lovelace Mixon died before realizing his dream of becoming a community organizer for the DNC.
Speed Cameras Proposed in Illinois

March 26, 2009 – From: www.chicagotribune.com

By Jon Hilkevitch

If you’re a driver who hates cameras that ticket you for running red lights, you won’t be revved up to support the next version of “cops in a box” possibly coming to Illinois.

Automated enforcement of speed limits would be allowed in the Chicago region and other areas under a proposed state law.

The move is part of a bill that would permit some counties and municipalities to mail speeding tickets of up to $100 for drivers caught going too fast by unmanned, stationary radar cameras positioned alongside roadways.

"I cannot feel sorry for those people caught by camera, because they are breaking the law," said state Sen. Terry Link (D−Waukegan), a sponsor of the legislation, which could move to a Senate vote next week. If approved by the General Assembly and signed by the governor, the law would take effect Jan. 1.

"If people start to slow down, they wouldn't have to worry about the fines," Link said.

More than 500 speed–related traffic deaths occurred in Illinois in 2007, according to state records. Nationally, about 13,000 people died that year in accidents where speeding was the cause or a contributing factor, according to the National Highway Traffic Safety Administration.

Under the proposed Illinois law, speeding tickets issued by automated surveillance would be treated as non–moving offenses, like parking tickets and red–light violations, and convictions or guilty pleas would not go on drivers' records, officials said.

Pictures of the offending license plate would be mailed to the registered owner of the vehicle. The driver and passengers would not be photographed.

Speed cameras would be permitted on roads in eight counties that have a history of speed–related accidents, where insufficient police manpower exists to enforce speed limits and where on–site enforcement is "inherently difficult."

The counties are Cook, DuPage, Kane, Lake, Madison, McHenry, St. Clair and Will.

In addition to permitting municipalities and counties to install automated speed–enforcement cameras, the legislation would lift a ban on recorded images for speed enforcement unless a police officer is present, officials said.
The bill makes no mention of signs that tell drivers how fast they are going. But the Illinois State Police’s roving photo−enforcement vans can operate only with the electronic “Your Speed Is ...” sign, under the law written for the ISP vans.

The measure is aimed at reducing accidents and fatalities and it is not an attempt to increase revenue, said Link and state Rep. Joe Lyons (D−Chicago), the bill’s sponsors.

The legislation is supported by the City of Chicago, Secretary of State Jesse White, law enforcement groups and other groups. “The reason people speed is because they can,” said John Ulczycki, vice president for research at the National Safety Council. “When people perceive that a law is not being enforced, speeding increases.” In Arizona, where stationary speed−enforcement cameras are deployed on a broad scale, speeding on highways has been cut by 9 m.p.h. on average, according to the state. Speed−related crashes along U.S. Highway 101 near Scottsdale have decreased 44 percent since the cameras were installed last year, officials said.

Last year, Rod Blagojevich, then governor, announced a plan to put speed cameras on interstates in Illinois. He said the plan could raise $50 million a year and allow the state to hire hundreds of state troopers.
The Little Red Hen in the Age of Obama

"Who will help me plant my wheat?" said the little red hen.

"Not I," said the cow.

"Not I," said the duck.

"Not I," said the pig.

"Not I," said the goose.

"Then I will do it by myself," said the little red hen, and so she did. She planted her crop, and the wheat grew very tall and ripened into golden grain.

"Who will help me reap my wheat?" asked the little red hen.

"Not I," said the duck.

"Out of my classification," said the pig.

"I'd lose my seniority," said the cow.

"I'd lose my unemployment compensation," said the goose.

"Then I will do it by myself," said the little red hen, and so she did.

At last it came time to bake the bread.

"Who will help me bake the bread?" asked the little red hen.

"That would be overtime for me," said the cow.

"I'd lose my welfare benefits," said the duck.

"I'm a dropout and never learned how," said the pig.

"If I'm to be the only helper, that's discrimination," said the goose.

"Then I will do it by myself," said the little red hen.

She baked five loaves and held them up for all of her neighbors to see. They wanted some and, in fact, demanded a share. But the little red hen said, "No, I shall eat all five loaves."

"Excess profits!" cried the cow. (Nancy Pelosi)

"Capitalist leech!" screamed the duck. (Barbara Boxer)

"I demand equal rights!" yelled the goose. (Jesse Jackson)

The pig just grunted in disdain. (Ted Kennedy)
And they all painted 'Unfair!' picket signs and marched around and around the little red hen, shouting obscenities. Then the farmer (Obama) came. He said to the little red hen, "You must not be so greedy."

"But I earned the bread," said the little red hen.

"Exactly," said Barack the farmer. "That is what makes our free enterprise system so wonderful. Anyone in the barnyard can earn as much as he wants. But under our modern government regulations, the productive workers must divide the fruits of their labor with those who are lazy and idle."

And they all lived happily ever after, including the little red hen, who smiled and clucked, "I am grateful, for now I truly understand." But her neighbors became quite disappointed in her. She never again baked bread because she joined the 'party' and got her bread free. And all the Democrats smiled. 'Fairness' had been established. Individual initiative had died, but nobody noticed; perhaps no one cared... so long as there was free bread that 'the rich' were paying for.

Then the bread stopped coming.