

"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 (HOLDAY)
 - Ever wanted special telco billing on your birthday?
- ◆ Page 40 / DMS-100 TOPS Holiday Treatment Table (HOLTRT)
 - ♦ Call treatment options for the DMS-100 HOLDAY 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.

processor configuration state counter reaches the second level or pump state.

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

Word

ASIC9

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)

ISS 2, SECTION 231-045-005

- 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.

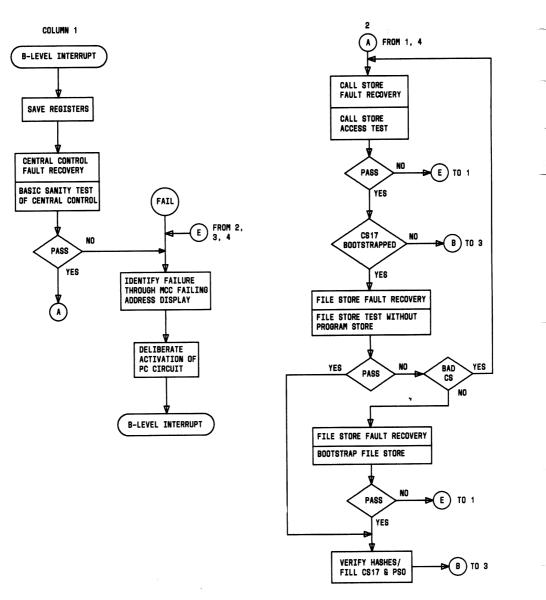
SOFTWARE INITIALIZATION

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. Transient data is just as important to the system sanity. Software initialization programs are provided to check the validity of transient data. v

Transient data consists of variable call pro-6.62 cessing data, ie, real-time activity data. Since this data is constantly changing, it is only backed up by a duplicate copy available in call store. Transient 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 transient data may result. If the transient 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-

SECTION 231-045-005







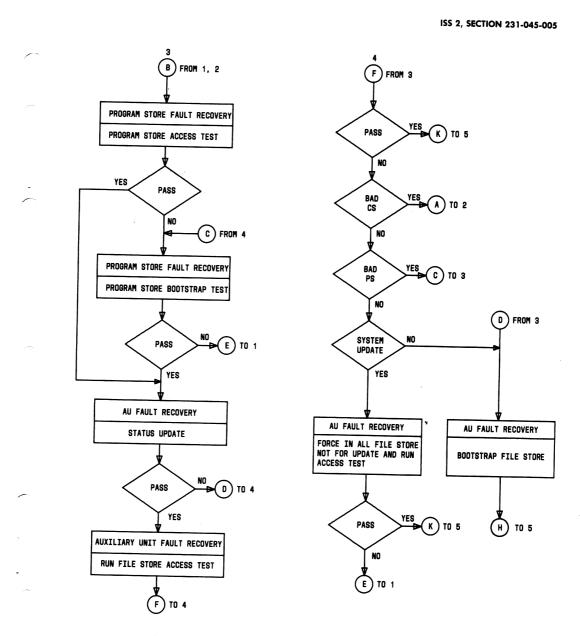
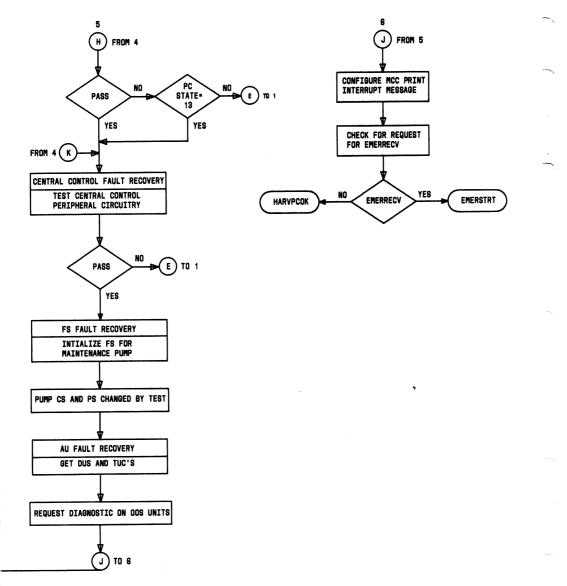


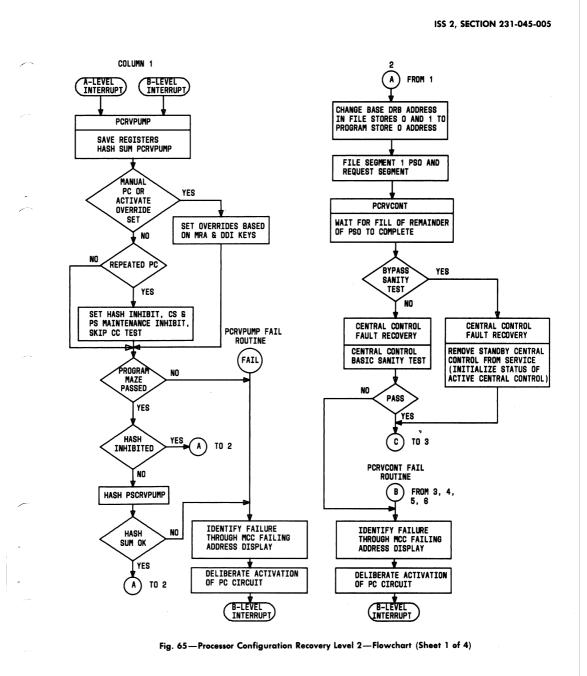
Fig. 64—Processor Configuration Recovery Level 1—Flowchart (Sheet 2 of 3)

SECTION 231-045-005









Page 161

6

SECTION 231-045-005

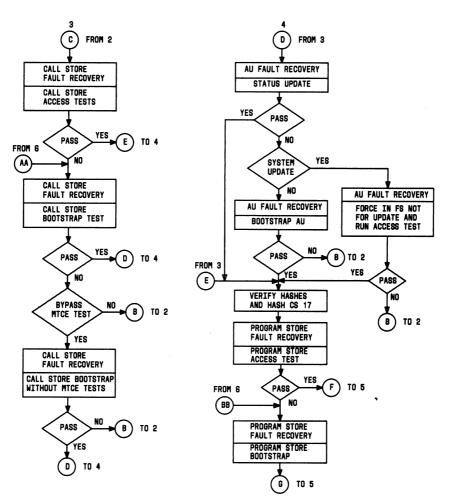


Fig. 65—Processor Configuration Recovery Level 2—Flowchart (Sheet 2 of 4)

tivity is primarily directed at regenerating transient data.

6.63 In the No. 1A 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; ie, 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-

Page 162

7

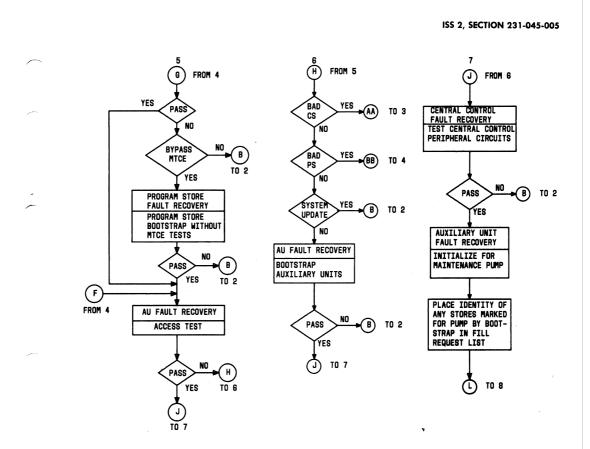


Fig. 65—Processor Configuration Recovery Level 2—Flowchart (Sheet 3 of 4)

gered, 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).

SECTION 231-045-005

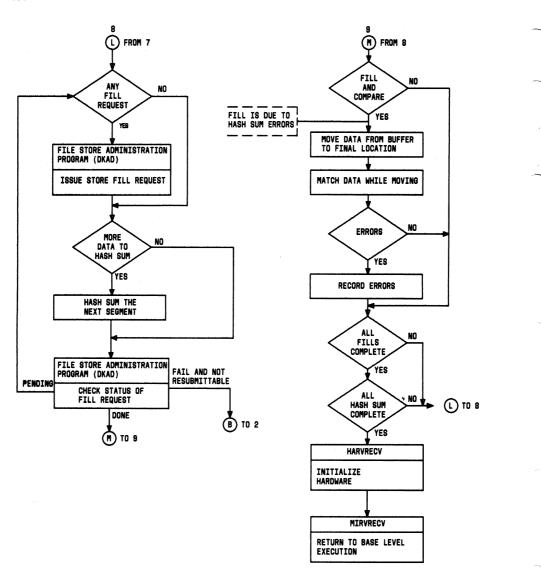
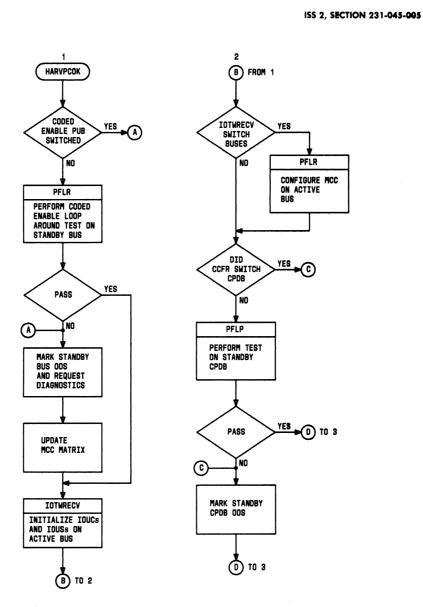


Fig. 65—Processor Configuration Recovery Level 2—Flowchart (Sheet 4 of 4)



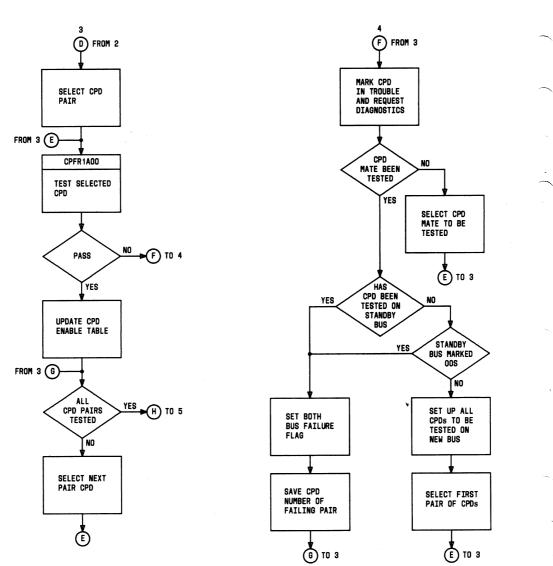




Page 165

10

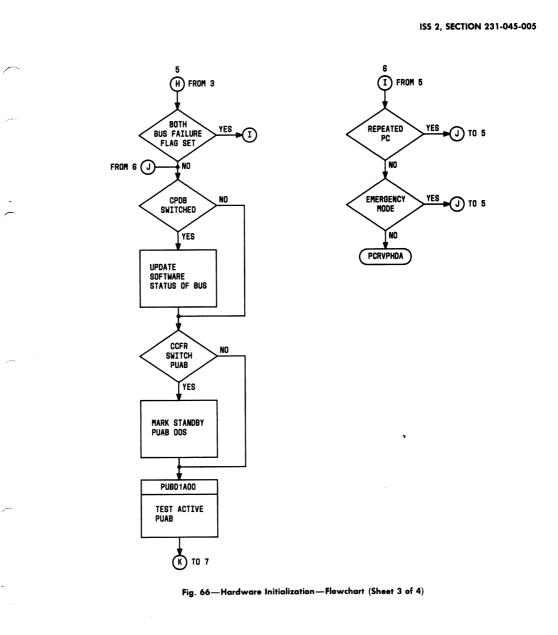
SECTION 231-045-005





Page 166

11



SECTION 231-045-005

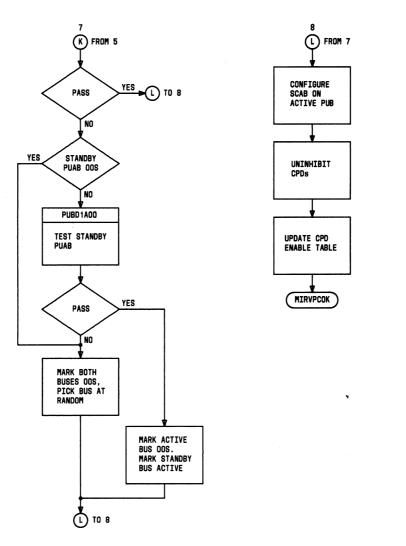


Fig. 66—Hardware Initialization—Flowchart (Sheet 4 of 4)

(b) Program Checks: These triggers are related to program check failures which cause a phase (eg, unanswered interject).

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

(d) Automatic Interrupt Triggers: These triggers are related to hardware faults that cause a phase (eg, 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 (ie, an established talking path) or an unstable state (eg, digit reception, ringing, etc). Any RCs in progress are removed when the phase in 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 (eg, network management audit, ring tip scan audits, etc) check and, if necessary, generate constant values within the transient data. If phase 1 is triggered by a data validation failure, additional audits are run in conjunction with audits normally requested. The additional audits are audits 24, 34, 36, 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).

ISS 2, SECTION 231-045-005

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; ie, processor configuration state counter must be greater than or equal to 16. This implies that the system has been hashed and if necessary pumped; ie, 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

SECTION 231-045-005

TABLE E

1A PHASE ACTIVITY

	PHASE		· · ·		
ACTIVITY		1	4	5	6
OVERVIEW		Constant's Audits	Zero CS Data	First Pump Phase	Manual Only
DURATION		1-2 sec	25 sec	25 sec	32 sec
EFFECT ON NONTRANSIENT DATA		Hash/Correct	Hash/Correct	Pump PC (state ≥ 16)	Pump PC (state ≥ 16)
EFFECT ON	STABLE	None	None	None	Knocked Down
CALL IN PROGRESS	UNSTABLE	None *	Knocked Down	Knocked Down	Knocked Down
RCs IN PROG	RESS	Canceled	Canceled	Canceled	Canceled
CS ZEROED		None	Same Compool and Parameter Defined Data	Same Compool and Parameter Defined Data	Same Compool and Parameter Defined Data
UTILITY EXECUTES AND LIBRARY ACTIVITY		Removed	Removed	Removed	Removed
OVERWRITES IN PROGRESS		Main Memory in Copy State Removed	Main Memory in Copy State Removed	Main Memory and Disk in Copy State Removed	Main Memory and Disk in Copy State Removed
INTERRUPTS INHIBITED		H, G, F, K, and D (auxiliary unit read/write failure only)	H, G, F, K, and D (auxiliary unit read/write failure only)	First Phase 5 — Same as phase 4 Repeated phase 5 — H, G, F, E, K, D (all sources) and B (GBP sources only)	As requested
AUDITS RUN		3, 4, 5, 6, 7, 8 9, 10, 11, 18, 32, 40, 71	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	Same as phase 4	$\begin{array}{c} 0, \ 4, \ 5, \ 6, \ 7, \ 8, \\ 9, \ 10, \ 12, \ 14, \\ 16, \ 18, \ 19, \ 20, \\ 22, \ 24, \ 26, \ 32, \\ 34, \ 36, \ 37, \ 38, \\ 39, \ 40, \ 41, \ 42, \\ 43, \ 46, \ 48, \ 50, \\ 58, \ 60, \ 61, \ 62, \\ 63, \ 64, \ 71 \end{array}$

Page 170

~

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.

ISS 2, SECTION 231-045-005

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, the 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 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).

SECTION 231-045-005

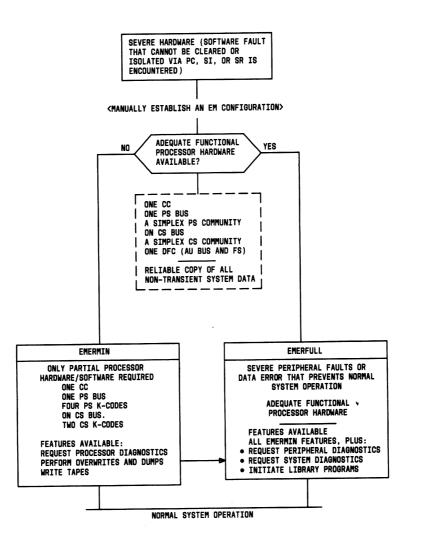


Fig. 67—Emergency Mode Configuration



17

ISS 2, SECTION 231-045-005

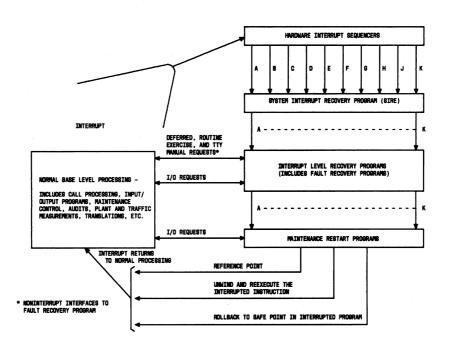


Fig. 68—General Program Flow for Fault Recovery

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.

SECTION 231-045-005

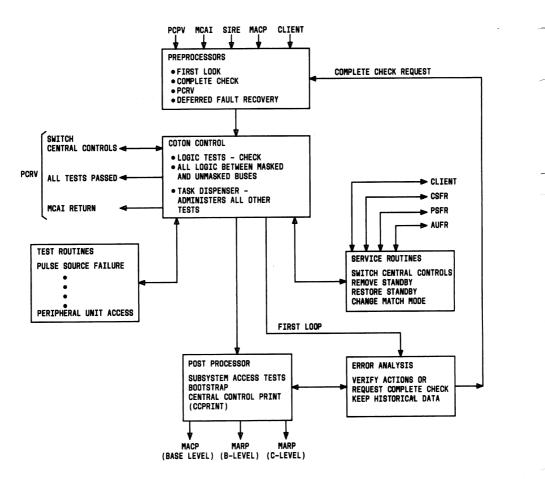


Fig. 69—Central Control Fault Recovery—Flowchart

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.

Page 174

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

ISS 2, SECTION 231-045-005

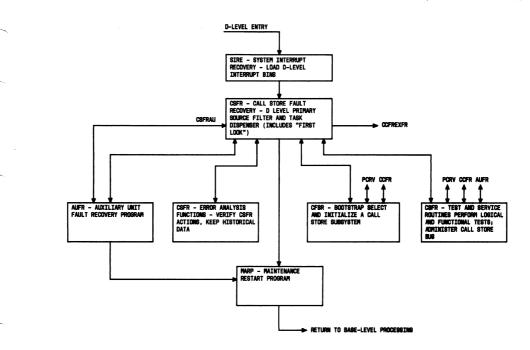


Fig. 70—Call Store Fault Recovery—Program Flow and Interfaces—Simplified

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

SECTION 231-045-005

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 as service until the rover update is completed.

6.94 If the error history is unacceptable, the suspected unit is removed from service and

Page 176

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

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 (AUBSQ). The AUBSQ 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 AUBSQ 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

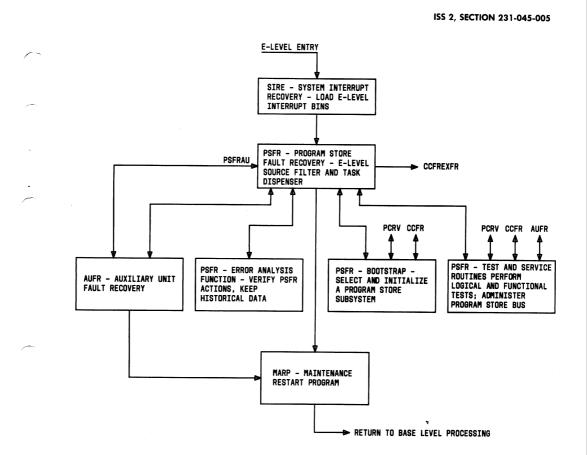


Fig. 71—Program Store Fault Recovery—Program Flow and Interfaces—Simplified

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 APIs. 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

SECTION 231-045-005

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 condi-

tions. 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

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.

translation data. The file store or APS is also used to

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 timeconsuming 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 (AUFR) 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

APIs. 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 AUFR, 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

ISS 2, SECTION 231-045-005

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 highcapacity 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 AUFR program is the interrupt and interject control program for the SSFR. All auxiliary unit interjects and D-levels are first handled by AUFR; consequently, AUFR 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 AUFR determines the fault was caused

by an auxiliary unit, AUFR communicates with the fault recovery programs for the faulty auxiliary unit. The AUFR program communicates with the unique fault recovery programs through a transfer vector table. For interrupt and interject processing, AUFR 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

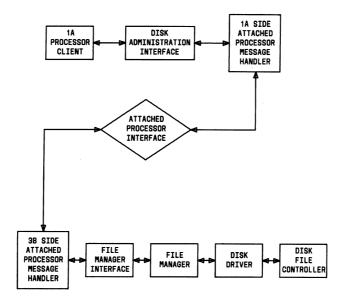


Fig. 72—Attached Processor Interface Layout (

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 APFRTBL and AUFR is important since the AUFR routine, known as double trouble (AUFRDTBL), 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 APFRTBL). 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 configured 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.

ISS 2, SECTION 231-045-005

- (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, APFRTBL, 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

proach 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, mainte-

nance 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 flipflops
- 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

Page 182

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 craftsperson 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

ISS 2, SECTION 231-045-005

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

SECTION 231-045-005

- (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 DOCU-MENTS)

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 troublelocating 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 diagnosable frame and are desig-

Page 184

nated 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)

buffers.

7.09 There are three general call processing related documents:

(a) CIC manual (PKxA123): This manual contains a series of Compool items, macros, and pidents that are used to load orders into peripheral order

(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 alphanumerical 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 alphanumeric 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.

ISS 2, SECTION 231-045-005

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; 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 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

SECTION 231-045-005

provided orients the craftsperson 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 (for-

mat 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

• Definitions of terms (GLOSSARY).

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 (eg, a link in the network map). Normally, a busy bit is equal to zero and a 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 junctors.

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 pidents transfer.

ISS 2, SECTION 231-045-005

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. buffer entry is processed. K-Code: A numerical definition of the address limits of a particular store. or other buffer. 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. 9. Multiline Hunt Group: A group of lines that provide the means to supply a set of special originating and 9.01 terminating services on a group basis rather than an individual basis. In addition to these services, line hunting is also provided. ADS Multi-MAC: Refers to the ability of MACP to run AEX concurrent jobs. AIF Page: One or more file store resident program sections, each of which is functionally complete, includ-AIO ing the subroutines called by the program sections. Paging: The operations required to bring a paged pro-ALI' gram from file store into core memory before execu-AM tion can begin. Path Memory: A part of temporary memory where AM enough information about a connection is stored to enable the system to reconstruct the connection. AM] Queue: A call store memory area used to record a waiting list of work which temporarily cannot be ANI completed. AOV Register: A call store memory area used to store information required to process a particular call in APC 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

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

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.

ABBREVIATIONS AND ACRONYMS

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

s	Auxiliary Data System		
X	Automatic Routine Exercise		
FR	Fault Recognition Program		
)D	Automatic Identified Outward Dialing		
IT	Automatic Line Insulation Test		
A	Automatic Message Accounting		
AC	AMA Data Accumulation Pro- gram		
DX	AMA Data Transfer Program		
I	Automatic Number Identification		
VD	Automatic Overload Control Pro- gram		
CL	Attached Processor Communica- tion Link		

SECTION 231-045-005

API	Attached Processor Interface	CO
АРМН	Attached Processor Message Han- dler	CI
APS	Attached Processor System	CI
APT	Automatic Progression Testing	01
ASW	All Seems Well	CI
ATAL	Audible, Disconnect, and Line Termination	CI CI
ATTT	Automatic Trunk Test Termina- tion	CS
AU	Auxiliary Unit	
AUBSQ	Auxiliary Unit Bus Sequence	CS
BCD	Binary Coded Decimal	CS
BINK	1024 words (BINary one K)	
CC	Central Control	C'l
CCAD	Customer Changeable Speed Call- ing Program	C
CCIS	Common Channel Interoffice Sig- naling	C2 C2
CCOL	Chart Column	C
CFUP	Call Forwarding Usage Program	
CHRN	Channel Request Number	C
CIC	Change in Circuit	C
CIN	Change in Network	C
CLID	Calling Line Identification List	
CMB	Channel Memory Block	C
CNC	Coin Charge Register	C
CNLP	Centrex Console Lamp Control Program	C
COCN	Coin Control Program	C
COIN	Coin Charge Program	U.

COPR	Report and Miscellaneous Subrou-
CPD	Central Pulse Distributor
CPDB	Central Pulse Distributor Enable
CPFR	Central Pulse Distributor Fault
CR	Call Register
CRFI	Common System Recorded An-
CS	Call Store Frame Input Analysis Program
CSDS	Circuit Switched Digital Capabil- ity
CSRAF	Common System Recorded An- nouncement Frame
CTYP	Call Type
CX1X	Centrex Tandem Tie Line Pro- gram
CXBV	Busy Verify-Trunk Test Program
CXDS	Disconnect for Centrex Program
CXIC	Incoming Digit Analysis for Cen- trex
CXIO	Centrex Input/Output Scan Pro- gram
CXKY	Centrex Key Signal Director
CXLO	Centrex Attendant Line andTrunk Seizure
CXOR	Centrex Digit Analysis Program
CXSF	Centrex Simulated Facilities Pro- gram
CXTA	Centrex Trunk Code-Call Answer Program
CXTP	Centrex Trunk Preemption Pro- gram

-

Page 188

33

ISS 2, SECTION 231-045-005

\sim	СХҮН	Seize and Release Routines	DUC	Data Uni
-	CZO	Coin Zone Operator	DUS	Data Uni
	DAC	Design Aid Computerized	ECIO	Executive Program
	DCON Diagnostic Control Program		ECMP	Executive
	DCONMAIN	Diagnostic Control Program (pident)	EMERFULL	Emergen Configura
,	DCS	Duplicated Call Store	EMERMIN	Emergen
<u> </u>	DCT	Digital Carrier Trunk		mum
	DDD	Direct Distance Dialing	EML	Emergen
	DIAG	Diagnostic	ERAP	1A Proce gram
	DISC	Disconnect Program	ERLI	- Error An
	DKAD	Disk Administration Program	FDIP	Frame D
	DKADI	Disk Administration Interface		gram
	DMA	Direct Memory Access	FM	File Man
$(-)^{-1}$	DMAPAPPL	Data Mapping Control and Link-	FMI	File Man
		ing Program	FOR	Fault Re
	DMERT	Duplex Multi-Environment Real- Time	FS	File Stor
	DOC	Dynamic Overload Control	FSAP	File Stor
	DOCT	Dictionary Trouble Number Pro-	FSC	File Stor
		gram	FSSP	File Stor Program
	DP	Dial Pulse	FSSR	File Stor
	DRE	Directional Reservation of Equip- ment	GCP	Generate
	DRPP	Diagnostic Results Post-	GRC	Growth I
		Processing	GULP	Generic I
r.	DSP	Dynamic Service Protection	HMTL	Hotel-Mo
	DTST	Dial Tone Speed Test Program	HUC	Higher U
	DU	Data Unit	ICB	
	DUAD	Data Unit Administration Pro- gram	IM	Input Ch Input Ma

Data Unit Controller
Data Unit Selector
Executive Control Input/Output Program
Executive Control Main Program
Emergency Mode Control Full Configuration
Emergency Mode Control Mini- mum
Emergency Manual Line
1A Processor Error Analysis Pro- gram
Error Analysis Library Program
Frame Dependent Interface Pro- gram
File Manager
File Manage Interface
Fault Recognition
File Store
File Store Administration Answer
File Store Controller
File Store Administration Submit Program
File Store Service Routine
Generate Control Pulse
Growth Recent Change
Generic Utility Program
Hotel-Motel Program
Higher Unduplicated Call Store
Input Character Buffer
Input Manual

SECTION 231-045-005

I/0	Input-Output	MCCP	Maintenance Control Center Pro- gram	
IOCP	Input/Output Control Program	MCLM	System Alarms Program	
IOT	Intraoffice Trunk	MCTWADMN	Master Control Center Adminis-	
IOU	Input/Output Unit		tration	``````````````````````````````````````
IOUC	Input/Output Unit Controller	MFJR	Multifrequency Signaling Junior Register	ŗ.
IOUS	Input/Output Unit Selector	MTS	Message Telecommunications Ser-	
ITTT	Incoming Trunk Test Termination		vice	-
KB/S	Kilobits per Second	MURL	Maintenance Unexpected Results List	$\left(\begin{array}{c} \\ \end{array} \right)$
L-L	Line to Line	NCD	Noncheck Dummy	
L-T	Line to Trunk	NETG	Network Growth Program	
LDR	Loader	NMFA	Network Fabric Routines Pro-	
LENCL-4	Line Equipment Number Class 4		gram	
LIBR	Library Control Program	NMFL	Network Maintenance Action Pro- gram	
LIBRTRP1	Library Control Common Traps Administrator	NMIN	Network Management Indicator Program	\sim
LIFO	Last in First Out	NMMP	Network Management Mainte- nance Program	
LLN	Line Link Network	NMMX	Network Matrix Exercise Pro-	
LUC	Lower Unduplicated Call Store		gram ,	
LULPUTIL	Local Generic Utility Program	NMRF	Network Fault Recognition Pro- gram	
MAACA	No. 1A ESSS Scheduler	NMTD	Transmit Dynamic Overload Con-	
MAC	Maintenance Control		trol Signals	1 - <
MACP	Maintenance Control Program	NMTG	Network Management Program	
MACR	Maintenance Control Peripheral	NTWK	Network Program	
	Program	OFGT	Miscellaneous Outgoing to Switch- boards and Desk Program	
MALM	System Alarm Program	OFML	Emergency Manual Line Service	1
MAUD	Maintenance Audit Program	OFML	Program	
MCC	Master Control Console	OFNT	Operator No Test Program	
MCCM	Common Control and Monitor Program	OFTR	Toll Switch and Recording Com- pleting	\sim

Page 190

35

ISS 2, SECTION 231-045-005

-	ОМ	Output Message	QSIF
	ОМО	Overtime Monitoring Operator	QTAL
	OMR	Output Message Register	D 4 69
	OOS	Out of Service	RACT
	OPCL	Outpulsing Control Register	RADR
	OR	Originating Register	RAF
	PAGS	Paging Supervision Program	RAM
	PATT	Processor Application Transfer Table	RAMP
	PBX	Private Branch Exchange	
	PC	Processor Configuration	RBB
	PDA	Parameter Data Assembler	RC
	PG	Generic Program Documentation	RCSS
	Date	Index	REX
~	PGID	Generic identification and Com- patibility	RI
	PLUG	Line Termination Denied Pro- gram	RI-PT
	POB	Peripheral Order Buffer	ROH
	PR	Program Listing	RPPS
	PRE	Protected Reservation of Equip- ment	RRT
	PS	Program Store	RVRT
	PSDC	Public Switched Digital Capabil-	SACT
		ity	SADT
	PT	Program Tag	SAWS
	PTW	Primary Translation Word	SCFR
	PUAB	Peripheral Unit Address Bus	SI
	QAPR	Queue and Administration Pro- cessing	SIRE
	QCIA		SIRE
	WUL	Customer Interface and Special Auditing	SR

, _____

1,1

~

1

ſ

Queue State Information Feature
Give Audible, Disconnect, and Line
Relay Activity Bit
Receiver Attachment Delay Report
Recorded Announcement Frame
Random Access Memory (Read-Write)
Recorded Announcement Machine Program
Rollback Block
Recent Change
Recent Change Subsystem
Routine Exercise
Register Identification
Register Identification-Program Tag
Receiver Off-hook
Regional Parameter Processing System
Routine Request Table
Reverting Call Program
Customer Program for Growth
System Audit Program
Writable Store Audits
Scanner Fault Recognition Pro- gram
System Initialization
System Interrupt Recovery Pro- gram
System Reinitialization

•

Page 191

-

Software System Introduction – Software Description / #1A ESS

SECTION 231-045-005

SRTT	Station Ringer and TOUCH- TONE Program	TTWK	Teletypewriter Work Register
SUPERV	Supervision Modernization	TTYM	Teletypewriter Translation Input/
SWAP	Switching Assembly Program		Output
SYPI	System Performance Indicator	TUC	Tape Unit Controller
	Program	TVNDX	Transfer Vector Index
SYUP	System Update Program	TWR	Teletypewriter Work Register
T-T	Trunk to Trunk		
TAND	Tandem Connection Program	TXFR	Call Forwarding Program
TBR	Teletypewriter Buffer Register	VFHC	Verification of H and C Register
TBTF	Through Balance Test Facility	WAIT	Call Waiting Program
TCC	Trunk Class Code	WPADAPL2	Write Protect Administration
TG	Translation Guide		
TGC	Trunk Group Control	WPADCOMM	Write Protect Administration Common
TLM	Trouble Locating Manual	WPADCTRL	Write Protect Administration
TLN	Trunk Line Network		Control
TODA	Ringing and Tone Plant Diagnos- tic	WQUE	Queue Administration Program
томк	Ringing and Tone Plant Monitor	WRDN	Word Number
TOPR	Toll Operator Signaling Program	YAHA	Seize and Release Routines
TRCE	Call Trace Program	YCCK	Register Link Routine
TRNS	Transition State	YCLK	Register Linking Routine
TSAH	Trunk Seizure and Answer Hop- per	YFDS	Scan of Single Master Scanner Point
TSPS	Traffic Service Position System	YFTO	Incoming Trunk to Busy Overflow
TTIA	Teletypewriter Input Messages Directory	YMRG	Miscellaneous Register Subrou- tines
TTOX	Teletypewriter Output Program	YTTO	Originating Line to Busy Overflow
TTPP	Teletypewriter Output Phases Program	ZERO	Call Store Zeroing Program

Page 192 192 Pages

37

s.,

DMS-100 TOPS Holiday Table (HOLDAY)

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-XXX). 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.

Overseas 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.

<u>Datafill</u>

The following table describes datafill for table HOLDAY:

Table HOLDAY Field Descriptions

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

-End-

Datafill Example

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

HOLNAME	MONTH	DAY
NEWYEAR	JAN	01
CHRISTMAS	DEC	25

DMS-100 TOPS Holiday Treatment Table (HOLTRT)

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 HOLTRT Field Descriptions

Field Subfield		Entry	Explanation and Action
HOLSCH		See Subfields	Holiday Treatment Key This field consists of subfields HOLNAME and SCHNAME.
	HOLNAME	Alphanumeric (up to 16 characters)	<i>Holiday Name</i> Enter the holiday name as previously defined table HOLDAY.
	SCHNAME	Alphanumeric (1 to 16 characters)	<i>Schedule Name</i> 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:

HOLSCH	HOLTRT
NEWYEAR ONTQUE	SUN
NEWYEAR CANUSA	SUN

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

HOLSCH	HOLTRT
NEWYEARS USA	SAT
NEWYEARS CANADA	SUN
XMAS USA	SUN
XMAS CANADA	SUN
XMAS UK	SUN

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.

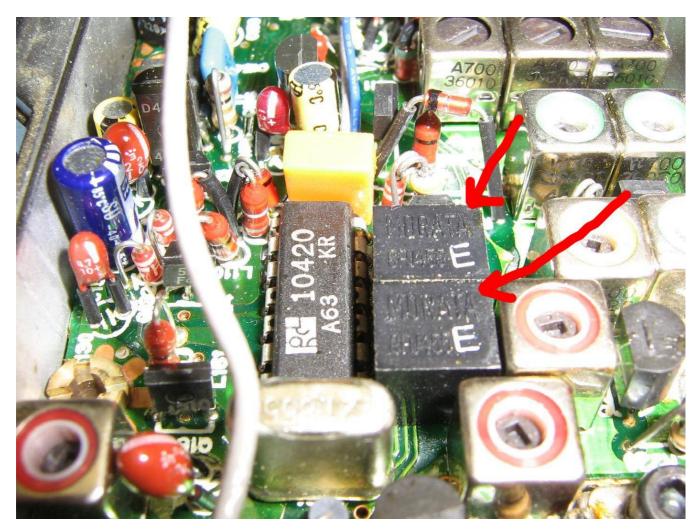
The PRO–2042's 455 kHz IF filter is most likely a Murata CFUM455D model. I say "likely" because I don't have the actual part number for that particular scanner, and Murata doesn't fully label their line of miniature IF filters. It doesn't really matter though, as the only thing we are interested in is 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 a 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 gerneral 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.

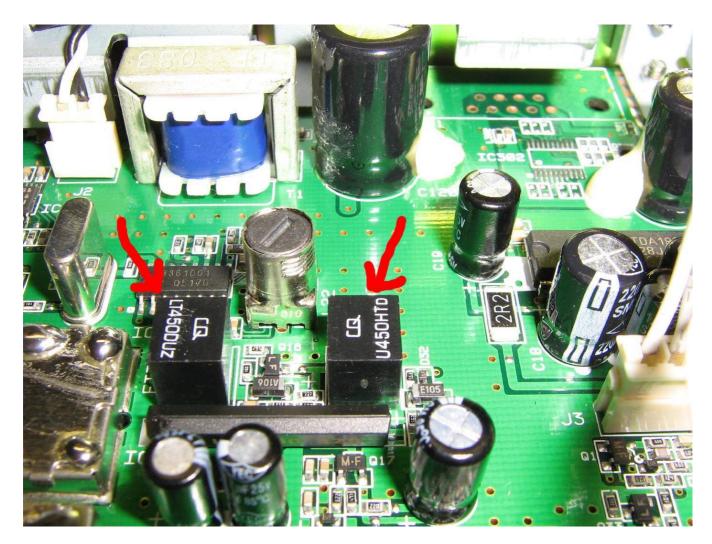
The radio's receiver chain is also based around the Toko 10420 and uses two Murata CFU455E 455 kHz IF filters in series to help improve receiver selectivity.



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 Kyocerra model with similar specifications and a slightly different pin–out.



Example IF filters in a Radio Shack PRO-2051 "Trunk Tracker" scanner.

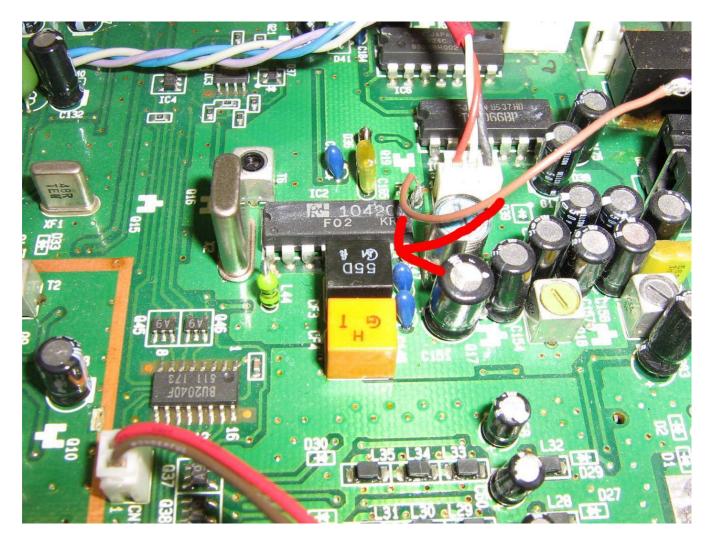
Because the PRO–2051 has to continuously monitor the data channel in a trunked radio system, this scanner actually has two receivers in it. The IF filter on the left is the standard "D" model, and the filter on the right is the "H" model. The "H" model has a very narrow bandwidth, 3 kHz or so. This is 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.

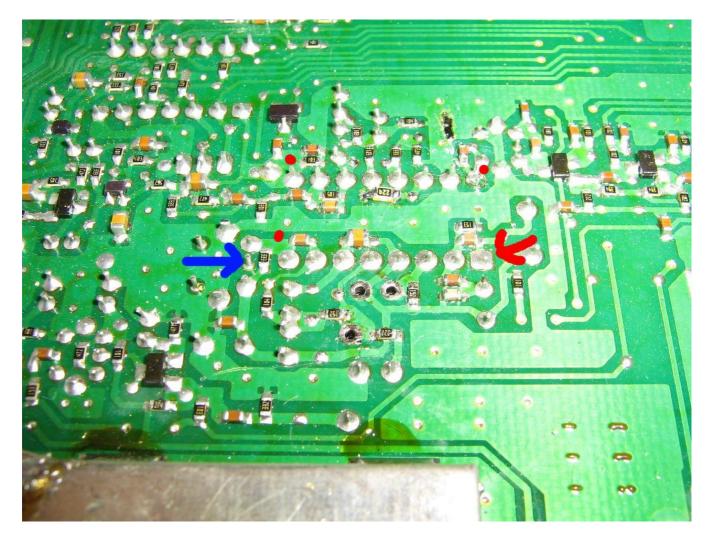
If you want to improve your scanner's selectivity even *more*, try to track down IF filters like the one shown above. These basically have the same bandwidth as the Murata "E" model filters, but the filter's "skirt," or attenuation factor, is much greater away from the center operating frequency. These 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.



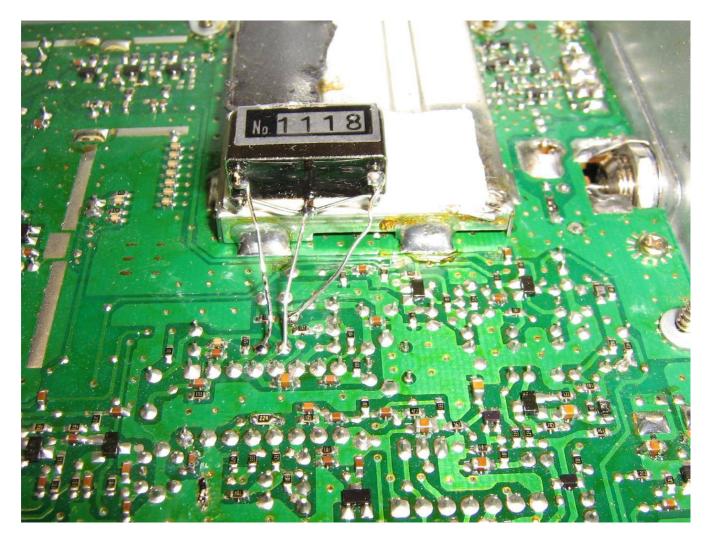
Bottom view of the PRO-2042 main circuit board showing the pin locations of the TK10420.

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.

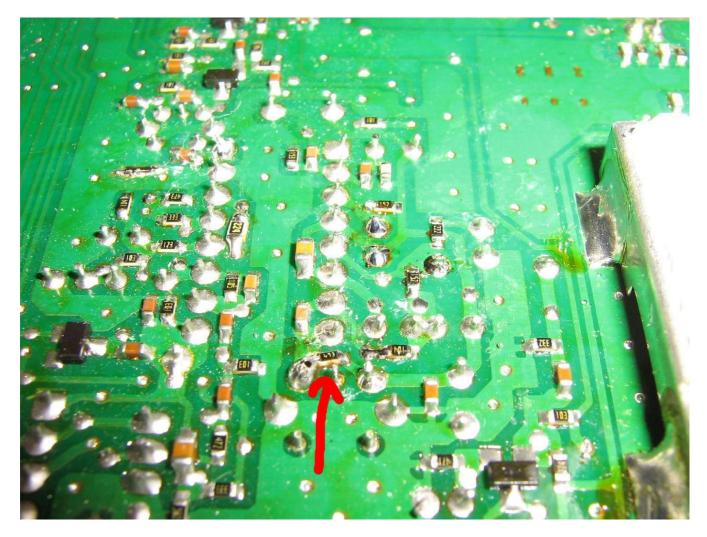


Installation of a new Murata mechanical 455 kHz IF filter.

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.

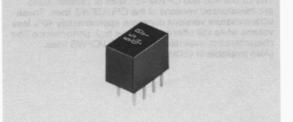


Changing the quadrature coil's damping resistor from 33,000 ohms to 47,000 ohms.

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



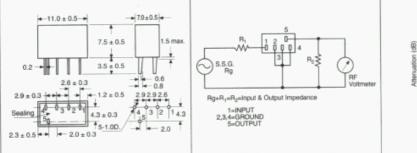


The CFWS 455 line of ceramic filters are 6-element devices connected in ladder form. These compact, highly selective filters are recommended for use in applications ranging from two-way radio to auxiliary filters in high class transceivers. (Also available in 450kHz version.)

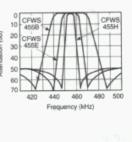
Part Number	Nominal Center Frequency (kHz)	6dB Bandwidth (kHz) min.	40dB Bandwidth (kHz) max.	Attenuation 455±100kHz (dB) min.	Ripple (dB) max. kHz	Insertion Loss (dB) max.	Input/Output Impedance (Ohms)
*CFWS455B	455	±15	±30	35	3 (455 ± 10)	4	1500
*CFWS455C	455	±12.5	±24	35	3 (455 ± 8)	4	1500
*CFWS455D	455	±10	±20	35	3 (455 ± 7)	4	1500
*CFWS455E	455	±7.5	±15	35	3 (455 ± 5.0)	6	1500
*CFWS455F	455	±6	±12.5	35	3 (455 ± 4)	6	2000
*CFWS455G	455	±4.5	±10	35	2 (455 ± 3)	6	2000
CFWS455HT	455	±3	±9	60	2 (455 ± 2)	6	2000
CFWS455IT	455	±2	±7.5	60	2 (455 ± 1.5)	6	2000

DIMENSIONS: mm

CIRCUIT



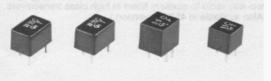
CHARACTERISTICS



*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, ULTRA-MINIATURE



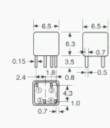
muRata CFUM/CFWM 455kHz

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

SPECIFICATIONS

CEUM 455kHz

Part Number	Nominal Center Frequency (kHz)	6dB Bandwidth (kHz) min.	40dB Bandwidth (kHz) max.	Attenuation 455±100kHz (dB) min.	Insertion Loss (dB) max.	Input/Output Impedance (Ohms)
*CFUM455B	455	±15	±30	27	4	1500
*CFUM455C	455	±12.5	±24	27	4	1500
*CFUM455D	455	±10	±20	27	4	1500
*CFUM455E	455	±7.5	±15	27	6	1500
*CFUM455F	455	±6	±12.5	27	6	2000
*CFUM455G	455	±4.5	±10	25	6	2000
*CFUM455H	455	±3	±9	35	6	2000
*CFUM4551	455	±2	±7.5	35	7	2000
DIMENSIONS:	mm	CIRCUIT			CHARACTERISTICS	





CFUM455

Rg

🖓 s.s.g.

w

R₁

1=INPUT 3,4=GROUND 2=OUTPUT

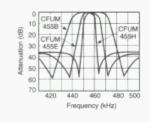
30.0

Rg+R1=R2=Input/Output Impedance

B.3

RF Volt





SPECIFICATIONS

9.5

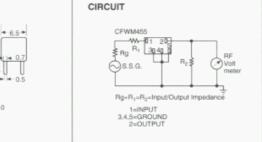
0.15-

2.5

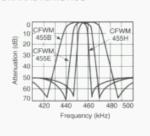
CFWM 455kHz

Part Number	Nominal Center Frequency (kHz)		40dB Bandwidth (kHz) max.	Attenuation 455±100kHz (dB) min.	Insertion Loss (dB) max.	Input/Output Impedance (Ohms)	
*CFWM455B	455	±15	±30	35	4	1500	
*CFWM455C	455	±12.5	±24	35	4	1500	
*CFWM455D	455	±10	±20	35	4	1500	
*CFWM455E	455	±7.5	±15	35	6	1500	
*CFWM455F	455	±6	±12.5	35	6	2000	
*CFWM455G	455	±4.5	±10	35	6	2000	
*CFWM455H	455	±3	±9	55	6	2000	
*CFWM455I	455	±2	±7.5	55	7	2000	

 CFWM455 series filters are 6-element ce ture versions of CFWS455^[] serie DIMENSIONS: mm







*Available as standard through authorized Murata Electronics Distributors.

6.3

3.5

2.8

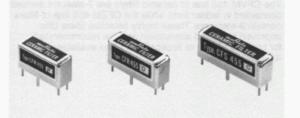
U 4

*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.

436

CG01-I

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 (BDT Improved) (NOT available in 450kHz.)

SPECIFICATIONS

IN

CFM/CFJ/CFR/CFS/CFL 455kHz

muRata

Part	Nominal Center	3dB		Ripple	Bandwi	and the second sec	Attenuation	Spurious	Insertion Loss	Input/Output Impedance (Ohms)	GroupDelay Time Dev. sec. max. (kHz)
Number	Frequency (kHz)	Bandwidth (kHz) min.	Bandwidth (kHz) min.	(dB) max.	(kHz) max.	At (dB)	455±100kHz (dB) min.	0.1 ~ 1MHz (dB) min.	(dB) max.		
CFM455A	455	±13	±17.5	3	±30		50	30	3	1000	
CFM455B	455	±10	±15	3	±25		50	30	3	1000	
CFM455C	455	±9	±13	3	±23		50	30	3	1000	
CFM455D	455	±7	±10	3	±20		50	30	3	1500	
CFM455E	455	±5.5	±8	3	±16	60	45	30	5	1500	
CFM455F	455	±4.2	±6	3	±12		45	30	6	2000	
CFM455G	455	_	±4	3	±10		45	30	6	2000	
CFM455H	455	_	±3	3	±7.5		45	30	6	2000	
CFM455I	455		±2	3	±5		45	30	7	2000	
CFR455A	455	±13	±17.5	3	±30		60	40	4	1000	
CFR4558	455	±10	±15	3	±25		60	40	4	1000	
CFR455C	455	±9	±13	3	±23		60	40	4	1000	
CFR455D	455	±7	±10	3	±20		60	40	4	1500	
CFR455E	455	±5.5	±8	3	±16	70	55	40	6	1500	
CFR455F	455	±4.2	±6	3	±12		55	40	6	2000	
CFR455G	455	_	±4	3	±10		55	40	6	2000	
CFR455H	455	_	±3	3	±7.5		55	40	7	2000	
CFR455I	455	_	±2	3	±5		55	40	8	2000	
CFR455J	455	_	±1.5	3	±4.5		55	40	8	2000	
CFJ455K5	455	_	2.4 (Total)	2	4.5 (Total)		_	60 40 at 600 ~ 700kHz	6	2000	
CFJ455K14	455	_	±1.1 ~ ±1.3	2	4.5 (Total)	60		60 40 at 600 ~ 700kHz	7	2000	
CFJ455K8	455	_	1.0 (Total)	1.5	3.0 (Total)		60	-	8	2000	
CFS455A	455	±13	±17.5	3	±30		70	50	4	1500	
CFS455B	455	±10	±15	3	±25		70	50	4	1500	
CFS455C	455	±9	±13	3	±23		70	50	4	1500	
CFS455D	455	±7	±10	3	±20		70	50	4	1500	
CFS455E	455	±5.5	±8	3	±15	80	70	50	6	1500	
CFS455F	455	±4.2	±6	3	±12		70	50	6	2000	
CFS455G	455		±4	3	±9		70	50	6	2000	
CFS455H	455		±3	3	±7.5		70	50	7	2000	
CFS455I	455		±2	3	±5		70	50	8	2000	
CFS455J	455		±1.5	3	±4.5		60	50	8	2000	
CFL455BG5	455	±10.5	±13.5	0.5	±27.5		60	30	10	1000	25µ (455±10.
CFL455CG1	455	±9.5	±12.0	0.5	±25.5	00	60	30	10	1000	35µ (455±9.5
CFL455DG2	455	±7.0	±9.0	0.5	±21.0	60	60	30	11	1000	35µ (455±7)
CFL455EG1	455	±5.0	±7.0	0.5	±18		60	30	13	1500	30µ (455±5)
DIMENSIO	NS: mm										
CFM	455	CFF	R 455		CFJ 455K		CF	S 455		CFL 4550	à
- 20.0		-	23.5	-	23.5	*	-	29.0	-	16.5	-
	† 10.5		-	0.5		¢ 10.5			10.5		† 10.5
4-10	4.0	L	4			4.0	- 4	↓ + 1.5	4.0		4.0

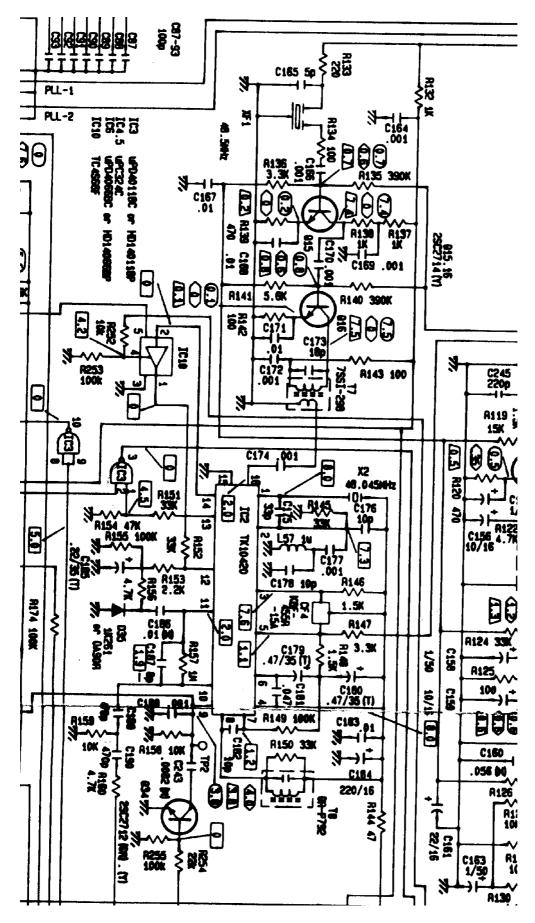
→ + 1.2 4.0 U ∐____ → | + 1.5 + + 1.2 4.0 -+ + 1.2 4.0 ų ų ų ų - 4-.5 Dia. 4-.8Dia. 4-.8 Dia. 4-.8Dia. 4-.8Dia. 2.00 4.0 7.5 ÷ 4.0 7.5 (3) (4). 4.2 7.5 4.0 (4) 0 (4)-0 (4)(2) तोत्र 14.2 20.0 20.0 . 26.0 4

*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.
438

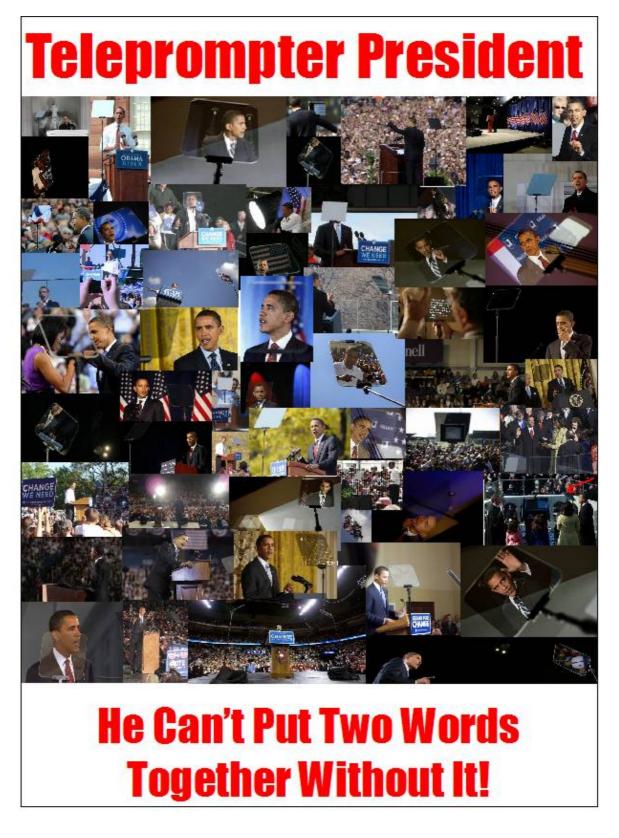
CG01-I

4.2 7.4

Radio Shack PRO-2006 NBFM IF Schematic







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, III. 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."

(www.politico.com/news/stories/0309/19663.html)

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."



Lovelle Mixon died before realizing his dream of becoming a community organizer for the DNC.

Note how they talk about these traffic cameras in terms of "public safety," then towards the article's end they mention how much money they bring in!

Welcome to the web of corruption known as Illinois Democrat politicians.

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.

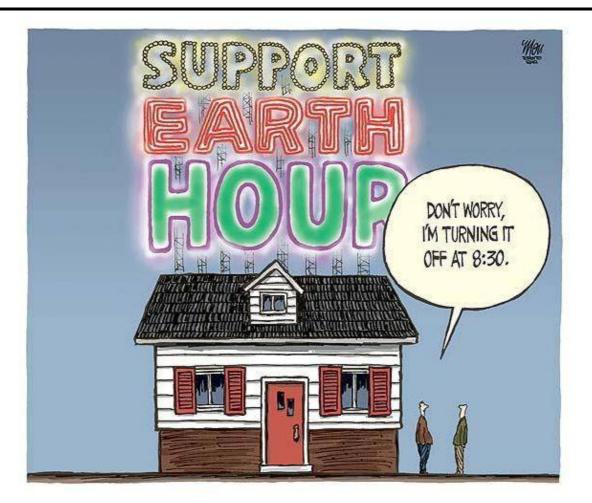
In addition to permitting municipalities and counties to install automated speed–enforcement cameras, the legislation would lift a ban on recorded images for speed enforcement unless a police officer is present, officials said.

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.

