

GBPPR 'Zine



Issue #55 / The Monthly Journal of the American Hacker / November 2008

"By now, there have been so many of these incidents, in so many places, in so many elections, that we must conclude that ACORN operates so as to encourage fraudulent registrations -- intentionally. That's a harsh charge, but I think the evidence supports it. They hire practically anyone, including people with criminal records. When they can, they pay their workers by the number of registrations they turn in. Not valid registrations, just registrations. Inevitably some of these workers cheat. And I am absolutely convinced that the people who run ACORN (and similar groups) know that some of their workers will cheat."

--- October 9, 2008 quote from Washington's (state) Secretary of State Sam Reed discussing ACORN voter fraud on the *Kirby Wilbur and Company* news talk show.

Table of Contents

- ◆ **Page 2 / Software System Introduction – Software Description / #1A ESS (Part 2)**
 - ◆ Detailed analysis of the software "generic" program which a #1A ESS operates under.
- ◆ **Page 28 / Rotary Coaxial Cable Connection Experiments**
 - ◆ Utilize the rotating drum head from an old VCR to make a low-frequency rotary coax connector.
- ◆ **Page 40 / DMS-100 Directory Number Route (DNROUTE)**
 - ◆ Directory number table entries which represent routes, rather than line equipment, in a DMS-100.
- ◆ **Page 44 / Horn Antenna for 2.45 GHz Microwave Oven Magnetrons**
 - ◆ Make a simple microwave horn antenna using only parts from a hobby store.
- ◆ **Page 57 / Bonus**
 - ◆ Fail
- ◆ **Page 58 / The End**
 - ◆ Editorial and rants.

a register to determine if the change is a true disconnect or a hit.

4.75 Universal and inband operator trunks are scanned to detect an off-hook change every 50 ms. The on-hook condition is reported via the trunk seizure and answer hopper. In the case of an on-hook change, program control is given to a routine which determines if the on-hook is a hit.

Directed Programs

4.76 The directed scan programs collect information about the state of a line or trunk at various stages of a call or maintenance procedure. The scan routines include:

- Digit collection
- Detection of abandons
- Detection of interdigital time-out
- Detection of hits, flashes, disconnect
- Origination (coin, test)
- Detection of signals for outpulsing
- Detection of revertive pulse signaling.

OUTPULSING SOFTWARE

A. General

4.77 The purpose of the outpulsing function is to control and coordinate the transmission of signals representing digits required to complete a call between two customers, served by different central offices.

4.78 The outgoing call control programs serve as the main controlling programs for the outpulsing process. These programs act as the interface between the client programs requiring outpulsing and the programs which perform the outpulsing functions.

B. Outpulsing Control

4.79 To communicate with other types of central offices, the ESS switch central office must be able to transmit digit information to each office in a

form which the distant office is prepared to accept. Since each type of office is different, several forms of signaling have been developed and used in the central office. The two most common signaling methods are MF signaling and DP signaling. Each of these outpulsing control programs perform the following functions:

- (a) Sets the outgoing trunk and transceiver into proper states for outpulsing
- (b) Tests the transmission path for continuity and polarity
- (c) Transmits trunk seizure signals
- (d) Detects start-pulsing signal
- (e) Causes the called number (and in some cases the calling number) to be transmitted.

The outpulsing function interfaces with five subsystems in order to handle all aspects of an interoffice call. Scanning, peripheral control, call processing, and translation subsystem interface are essential in completing any type call, while diagnostic programs are called upon in case of any malfunction. This interface is summarized in Fig. 8.

C. Outpulsing Description

4.80 Entry to the outpulsing subsystem is always made by the outgoing call control programs and may come from any one of the following client programs:

- Digit analysis programs
- Add-on and dial conference programs
- Coin control programs
- Trunk and line test panel programs
- Trunk and service circuit diagnostic programs
- Tandem connection program.

4.81 When outpulsing is requested by one of the above client programs, the client program initializes an outpulsing control register (OPCL). This register is used to store outgoing call control infor-

SECTION 231-045-005

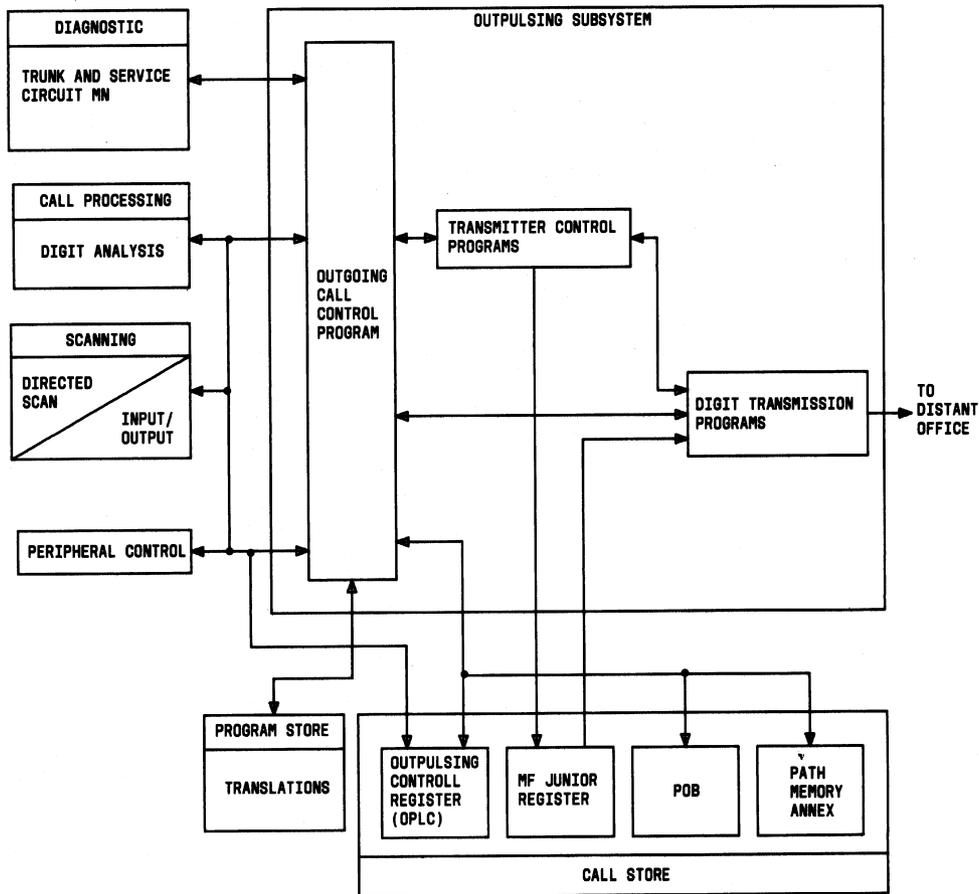


Fig. 8—Outpulsing Interfaces

information needed during outpulsing operations. The OPLC contains the following input information:

- Trunk hunt control information
- Path memory information
- Digit collected or derived from translations
- Outpulse prefix and delete digit information
- Receiver preemption control information
- Return information.

Trunk Hunt Control Information

4.82 The client program furnishes trunk hunt control information through the use of network control programs. This information includes selection of an idle outgoing trunk and transmitter. The selection of a transmitter depends on the type of receiver at the distant office (MF or DP). The network control programs are provided with the ability to find an idle trunk in a group and to make a second trial if all paths to the first selected trunk are busy. The trunk and transceiver locations and the type of outpulsing (type transmitter) are stored in the OPCL.

Path Memory Information

4.83 The network control programs are used to select paths through the network. The busy/idle record of all links in the network is kept in the network map; the network control programs can reserve a path from one terminal to another for use in the call. The paths selected are reserved by changing the busy-idle bits associated with the paths to busy. The path information is stored in the OPCL. The instructions to establish the paths are loaded into the peripheral order buffer.

Digits Collected or Derived from Translations

4.84 The digits collected or derived from translations are the digits to be outpulsed. The number of digits outpulsed and the stage when the outpulsing function will begin are determined by the translation programs. The translation programs may indicate overlap outpulsing, a method of speeding up call completion by starting to outpulse digits before all dialed digits are received. When overlap outpulsing is performed, the client program transfers control to the outpulsing control program prior to receiving all the digits to be outpulsed. This information, as well as the remaining digits to be outpulsed is stored in the outpulsing control register when received by the client.

Return Information

4.85 The OPCL contains the appropriate return addresses and coding to reflect the state of a call at the completion of outpulsing. This information indicates one of two conditions: outpulsing success or outpulsing failure.

D. MF Outpulsing

4.86 The MF outpulsing is a high-speed, long-range method of digit transmission using short-duration bursts of voiceband tones. Speed is gained over other pulsing methods because each digit is assigned a unique pair of tones and need not be built up in time by a series of like pulses, as is done in dial pulsing. Range is extended because voiceband signals can use any amplifying equipment in the trunk over which they travel without special bypassing or relaying circuits.

4.87 Before the MF outpulsing control program is entered, the client program to outpulsing has collected and analyzed the digits which identify the terminating office. The outgoing call control program has performed the following:

- (a) Seized an outpulsing annex register
- (b) Seized a peripheral order buffer
- (c) Called the peripheral order buffer loading programs for changes in network to hunt and busy an outgoing trunk, transmitter, network path, and to load the peripheral order buffer with orders to connect the path from the outgoing trunk to the transceiver.

4.88 An MF junior register (MFJR) is seized and initialized. The peripheral order buffer is loaded with orders to close cut-throughs to the transmitter, putting the outgoing trunk and transmitter into the proper state for outpulsing. The peripheral order buffer is activated, and the MFJR is initialized for digit transmission using the MF outpulsing control program.

4.89 After the peripheral order buffer is successfully executed, preparations are made for digit transmission. If an abandon occurs, the outpulsing process is terminated and control is returned to the client program to take down the connection.

E. Dial Pulse (DP) Outpulsing

4.90 The DP method of digit transmission consists of transmitting trains of direct current pulses representing the called number of the terminating customer line. Each train represents a digit. The numerical value of each digit is represented by the number of on-hook intervals in the train of pulses.

SECTION 231-045-005

The on-hook intervals of each pulse are separated by short off-hook intervals, while the digits themselves are separated by relatively long off-hook intervals called interdigital intervals.

PERIPHERAL CONTROL SOFTWARE

A. General

4.91 The peripheral control software (Fig. 9) is responsible for directing the central control in those actions necessary to control the switching network. Network actions include the interconnection of lines, trunks, service circuits, and tone sources associated with a central office. The peripheral control routines supply the following services:

- (a) Hunt for an idle path through the network
- (b) Administer the network map and path memory
- (c) Control the loading and execution of peripheral orders in the peripheral order buffers.

4.92 The network is comprised of line link networks and trunk link networks, providing 2-wire metallic paths through eight stages of switching. Of the many discrete paths through the network, there are only three types of paths. Between any two network terminals the paths are line-to-line (L-L), line-to-trunk (L-T), and trunk-to-trunk (T-T). Any network path or connection can be in one of the following path states:

- (a) **Connected:** If the path is in use and current flows, the path is said to be connected. In order for current to flow through a path, all the network crosspoints that comprise the path must be closed and any trunk or junctor circuit(s) in the path must be in the cut-through state. A record of this connected path is kept in duplicated call store (referred to as path memory). This record permits a determination of which link and junctors are used in the path. The in-use links and junctors are marked busy in the link map area of call store. This record prevents the links and junctors used in the busy path from being used in any other path.
- (b) **Reserved:** If the path is in use and no current is flowing, the path is said to be reserved. Link map and path memory records similar to those kept for a connected path are retained for

this path state. Any number of network crosspoints that comprise the reserved path may be closed.

- (c) **No-Path:** If the path is not in use, then the path is said to be in the no-path state (idle).

4.93 The three network states mentioned can be configured to form eight transitions for network path(s). These eight transitions and their associated states are listed in Fig. 10. Figure 10 also illustrates the terms and use of the network transitions as they relate to an idle path (no-path), memory, and path hardware. Two of the transitions (C and S) require network and signal distributor relay operations. Another two (T and A) require signal distributor relay release operations. In all cases, the network path crosspoints are closed prior to operating the appropriate signal distributor point. This is to ensure network crosspoints do not switch current.

B. Entry to Peripheral Control

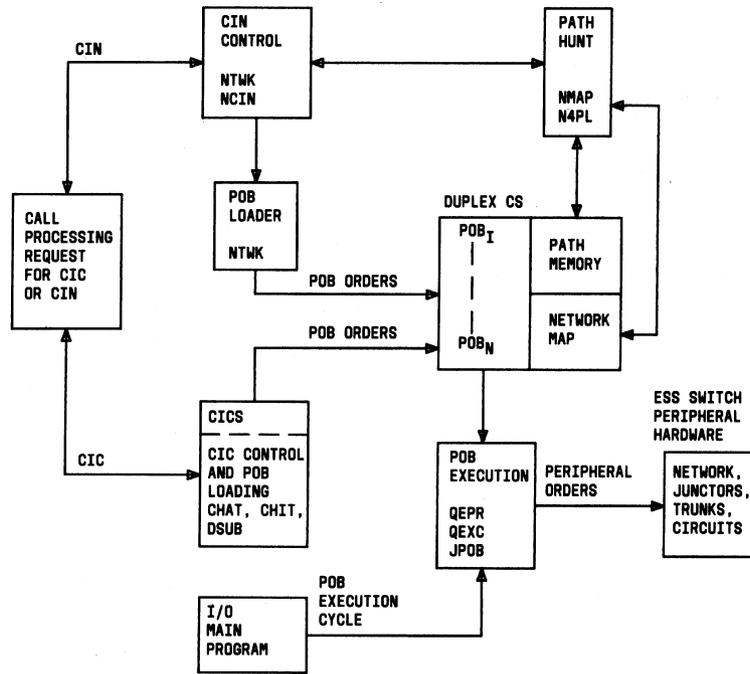
4.94 Entry to the peripheral control subsystem is made from any client program (call processing or maintenance) requiring a network path (or circuit) change, eg, dialing connection, ringing connection, incoming trunk test, etc. This network path change may require a single transition or several simultaneous transitions to complete the proper network path. Therefore, the client will perform one or both of the following functions:

- (a) Change in network (CIN)†
- (b) Change in circuit (CIC).

Change in Network (CIN)

4.95 For each transition involving two or more terminals, there is a unique subroutine called a CIN. This CIN is generally requested with the aid of a macro call. This macro consists of a series of instructions which sets up the calling sequence (or entrance requirements) for a CIN subroutine and subsequent transfer to a sub-CIN subroutine.

4.96 The CIN programs transfer program control to the network path hunt program when a request to find an idle path between two terminals is made. This hunt for an idle path is accomplished by interrogating the activity or busy/idle bits in the network link map. The network link map contains a



LEGEND:

- CHAT - CHANGE IN CIRCUIT HEAD AND RELAY ACTION TABLES
- CHIT - CHANGE IN CIRCUIT TABLES
- CIC - CHANGE IN CIRCUIT
- CICS - CHANGE IN CIRCUIT SUBROUTINES
- CIN - CHANGE IN NETWORK
- CS - CALL STORE
- DSUB - DISTRIBUTOR AND SCANNER SUBROUTINES
- JPOB - J-LEVEL PERIPHERAL ORDER BUFFER
- NCIN - CHANGE IN NETWORK PROGRAM
- NMAP - NETWORK MAP ADMINISTRATION PROGRAM
- NTWK - NETWORK TRANSITION CONTROL PROGRAM
- N4PL - NETWORK TABLES
- POB - PERIPHERAL ORDER BUFFER
- QEPR - POB ADMINISTRATION
- QEXC - QUEUE EXECUTION

♦ Fig. 9—Peripheral Control Subsystem ♦

SECTION 231-045-005

OLD STATE (FROM)	NEW STATE (TO)	TRANSITION
NO-PATH NO-PATH	RESERVED CONNECTED	RESERVE (R) CONNECT (C)
RESERVED RESERVED RESERVED	NO-PATH RESERVED CONNECTED	ERASE (E) HOLD (H) SET-UP (S)
CONNECTED CONNECTED CONNECTED	NO-PATH RESERVED CONNECTED	ABANDON (A) TO-HOLD (T) KEEP (K)

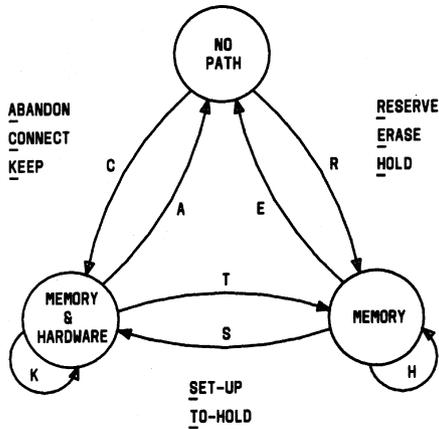


Fig. 10—Network Path Transitions

busy/idle indication for all links in the ESS switch network. A path from one terminal to another can be reserved by changing the busy/idle bits associated with the path to busy. This path selection process is performed by the network path hunt programs.

Change in Circuit (CIC)

4.97 For each transition involving the control of trunk(s) or service circuits via a signal distributor point, there is a unique subroutine called a CIC. The CIC expands program orders into the required data to effect the desired circuit state change. The CIC subroutine also provides distributor and scanner subroutines which load peripheral order buffers, and perform scanning and central pulse distributor operations without a peripheral order buffer.

4.98 After the appropriate peripheral orders are formed (by the CIN or CIC routines), they are placed into a peripheral order buffer. The peripheral order buffer holds the orders in the proper execution sequence until activation time; then the peripheral order buffer execution program unloads the orders and sends them to the peripheral system.

C. Peripheral Order Buffer Execution

4.99 The purpose of the peripheral order buffer execution programs is to control the rate at

which signal distributor, network, central pulse distributor, and scanner orders are sent to peripheral units. These orders perform the function of switching telephone traffic. Prior to peripheral order buffer execution, these orders are placed, at the request of a client program, in a peripheral order buffer. During every fifth J-level interrupt (approximately every 25 ms) the peripheral order buffer execution program sends the orders loaded in that peripheral order buffer to the appropriate peripheral units. However, with incoming step-by-step trunks this normal peripheral order buffer scheme is not fast enough to recognize an abandon and an immediate reseizure. Therefore, J-level peripheral order buffers are set up which rapidly handle orders associated with these bylink trunks. The speed increase is due to J-level peripheral order buffers being loaded in J-level rather than base level. The J-level peripheral order buffer execution precedes normal peripheral order buffer execution. The peripheral order buffer execution programs can be divided into the following functions:

- Memory usage
- Peripheral order buffer administration
- Peripheral order buffer execution
- J-level peripheral order buffer actions.

OPERATOR FUNCTIONS

A. General

4.100 The operator functions programs (Fig. 11) process all calls requiring operator assistance or intervention. The actions of the operator function programs are described in the following paragraphs.

B. No Test Calls

4.101 Operator no test calls are handled by pident OFNT. No test calls allow an operator to verify that a line is legitimately busy or break in on an existing call in an emergency. When an operator originates a no test call, OFNT sets up a connection from the operator to the desired line, if possible, or returns a signal to the operator if a connection is not possible. This may occur if there is trouble on the line, blocking due to traffic, or if the line has temporary transfer active.

C. Emergency Manual Line (EML) Service

4.102 The emergency manual line service program (OFML) allows the operator to provide service to critical customers manually by bypassing the switching machine in case the ESS switch is severely overloaded or inoperative.

D. Miscellaneous Outgoing to Switchboards and Desk

4.103 The miscellaneous outgoing to switchboards and desk program (OFGT) sets up connections to various trunks such as recording completing, intercept, business office, and repair service. It disconnects the digit receiver, seizes an operator register, and reserves a network path and a trunk of the appropriate type. When an operator answers, it gives the operator a class of service tone about the calling line and sets up the previously reserved path between line and operator.

E. Toll Switch Origination

4.104 Calls originated by an operator, for example, to complete a toll call, are handled by OFTR. It seizes an incoming register to store needed information. If the called line is not busy, OFTR seizes an operator register, transfers information from the incoming register to the operator register, and releases the incoming register. The OFTR then sets up a path to a ringing trunk, and establishes a talking path if the called party answers. The OFTR also processes coin collect and return, ringing control, and disconnects.

F. Traffic Service Position System (TSPS)

4.105 The TSPS program handles all calls routed to a traffic service position. This includes spe-

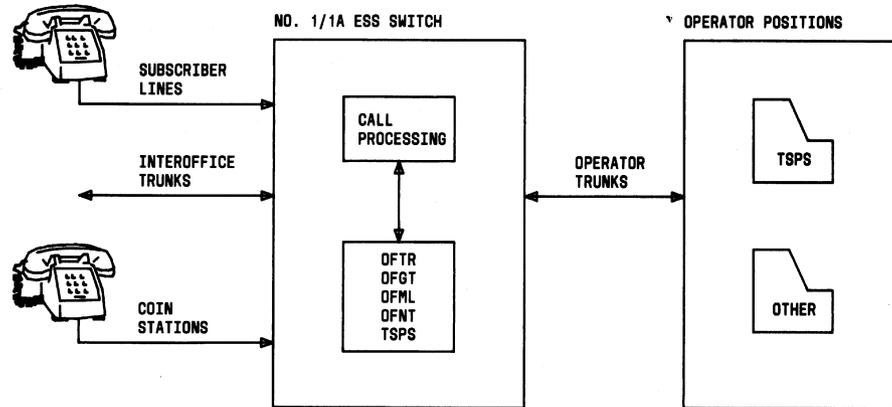


Fig. 11—Operator Functions Software Subsystem

SECTION 231-045-005

cial toll calls such as collect or person-to-person calls. It interfaces with OFTR to seize an operator register and connect a ringing circuit. The TSPS also has entries which place a 911 emergency trunk or a regular TSPS trunk which is on maintenance onto the high and wet list.

G. Toll Operator Signaling

4.106 The toll operator signaling program (TOPR) handles signaling to and from switchboards. These signals are sent over intertoll trunks and access toll switches include ringback, ring forward, TSPS double ring forward, and alerting wink. The TOPR is entered from the scan point change director program when a trunk ferrod changes states. It takes appropriate actions based on what type of signal is detected. It also handles answer and disconnect.

CHARGING SOFTWARE

A. General

4.107 The AMA facilities of the ESS switch (Fig. 12) provide a means of collecting certain billing data related to the charging of customer calls.

B. Collecting Local AMA Data

4.108 Once the call processing programs determine that a particular line origination is requesting a billable service, an attempt is made to put an AMA register on the call. Pident AMAC is entered from call processing to seize an AMA register. Once the AMA register has been linked to the call, billing information such as the calling directory number and called directory number is transferred from the originating register (OR) to the AMA register. Subsequent data entries to the AMA register include call type (CTYP), answer time, special service marks, disconnect time, etc. Time of day information is supplied

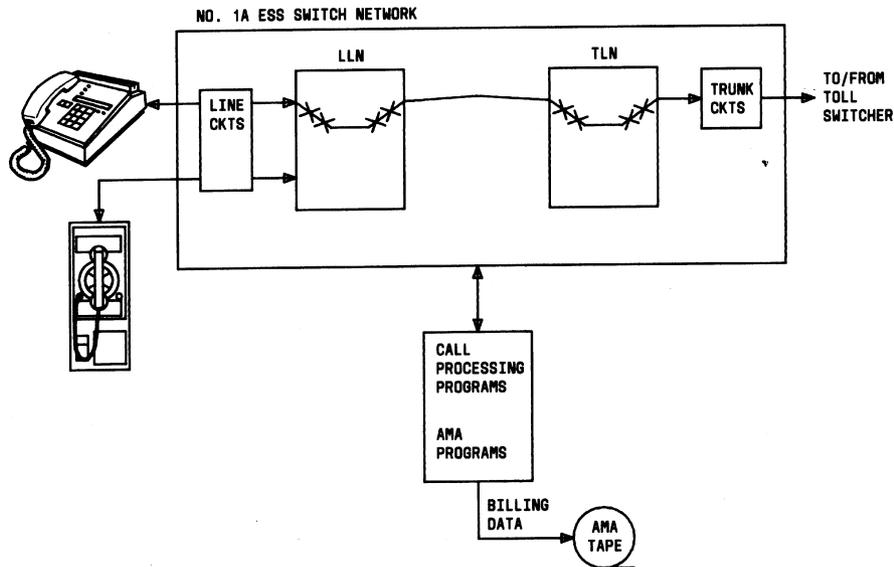


Fig. 12—AMA System

by the ESS switch system clock, in units of hours, minutes, seconds, and tenths of seconds.

4.109 After the call has terminated and all the required billing information is in the AMA register, the AMA data accumulation program (AMAC) organizes the AMA register data into a binary coded decimal (BCD) format for storage in the AMA buffer area of duplicated call store. The stored BCD formatted information for each call must be a multiple of 5 BCD digits. When required, the information is padded (extended) to a multiple of 5 BCD digits. Padding is done with noncheck dummy (NCD) characters. The AMA buffer, administered by AMAC, consists of a dedicated block of words for duplicated call store and additional words of call store are provided for overflow. After each AMA register is reformatted into the buffer, AMAC checks for a buffer-full condition. If full, AMAC sets up an entry to the AMA data transfer program (pident AMDX) which will unload the AMA buffer and transfer the data (which is reformatted again by AMDX) to another buffer for subsequent output to tape.

4.110 Normally only completed calls are recorded by the AMA system. However, the call processing programs are allowed to insert a number of special marks in the OR. These marks, when present, will cause incomplete calls to be recorded.

4.111 The AMA programs collect the charging data for all automatically billable calls. Charging and other call related data are initially collected into an OR by the call processing programs. Call processing then enters AMAC to associate an AMA register with the call. At the time of initial seizure of an AMA register, AMAC is called as a subroutine of the call processing client currently in control of the call. This is because AMA is concerned only with billing functions and not with such things as calling subscriber abandon or network path configuration. The AMAC is entered on outgoing calls just before the outgoing call control program is given control. On intraoffice calls, eg, message rate, AMAC is entered just before the ringing program gets control.

4.112 The AMA register initialization requires that an OR be on the link of registers serving the call, although any standard call register, eg, coin charge, may be master. Initial billing information, such as calling directory number, billing index or chart class (depending upon service marks in the OR), and called directory number, is copied from the

OR into the AMA register. Upon entry, AMAC attempts to associate a 13-word AMA register (depending on the type of billing) with the call.

4.113 If all AMA registers are busy because of high traffic, AMAC decides whether the call will be allowed to complete. Generally, if AMAC fails to seize an AMA register, then single unit message rate and coin calls are allowed to go with no billing data collected. All other calls are given reorder tone. If there is an AMA hardware failure (resulting in a lack of buffer space and AMA registers), then all calls are allowed to go with no billing data collected.

C. AMAC Interfacing

4.114 Once an AMA register is put on a call, returns to AMAC from the call processing programs (to report things such as answer, disconnect, abandon, etc) are via a program tag (PT) table. The PT is a pointer set by AMAC into the AMA register. The PT defines the proper return entry to AMAC from the calling program.

D. AMA Register-to-Buffer Processing

4.115 After all billing data for a particular type of call has been collected in an AMA register (note that this does not necessarily imply that the call has terminated), AMAC is reentered via the program tag table. For example, if billing data on a call has been completed with a disconnect time, the disconnect program (pident DISC) reenters AMAC via the program tag. Following checks for available space in the AMA buffer, AMAC builds the AMA register data into the required data formats and stores the formatted data in the buffer. The data formatting routines, based on the type of billing data collected, select other routines which in turn select the formats, ie, data groups, to be built for the particular call type. After the formatting and buffer loading is complete, the buffer input pointer (which points to the next available buffer entry slot) is updated and buffer full condition is checked. If full, AMA_GO sets a queuing condition to queue AMA registers waiting to be unloaded and sets up a call to the AMA data transfer program.

4.116 If AMA register queuing occurs and an AMA register does not clear the call register queue within 1 to 6 minutes, an audit will dump it off. Since only about 0.5 second is required for AMDX to copy a full buffer to tape, if an AMA register is stuck on

SECTION 231-045-005

the queue for longer than a couple of seconds, then a major AMA hardware failure has occurred and some AMA data will be lost.

E. AMA Midnight Routine

4.117 At midnight, AMAC is entered from ECMP, and all AMA registers in the office are checked for traffic for the day's tape. Registers which are busy, ie, collecting billing data on a call, are found by testing their PT values. The AMA buffer is readied for a new day's activity by doing a final write to tape. This end-of-day processing results in a clearing of the AMA buffer with the buffer input pointer reset to the first buffer slot.

4.118 For any in-process AMA call, other checks are made to determine the number of midnights passed since the call entered the system. If an AMA register is busy on a call at midnight, the register's midnight counter (AMA register item MNT) is scored. This counter has a maximum count of nine. Midnight counts of 2 to 9 generate an AMA output message on the maintenance TTY. The message includes the calling directory number, number of midnights passed since the call entered the system, and the address of the AMA register serving the call. If the MNT count reaches 10, the AMA register is taken off the call with a dummy code entered for disconnect time. The call is left up.

4.119 Other checks are made to catch disconnects which occur just before midnight but after the MNT counters are scored. This situation could occur if a call was in the "timing for final disconnect" state at midnight.

COIN FUNCTIONS

A. General

4.120 The coin programs handle charging for local and coin zone calls from coin stations. These stations may be dial-tone-first or coin first. The functions performed by coin programs include the following:

- (a) Seize and release coin charge registers and, if needed, AMA registers
- (b) Test for presence of initial and overtime deposits
- (c) Call either ringing and answer detection programs or outpulsing programs to connect the coin station to called line

- (d) Determine call rates and timing intervals
- (e) Collect initial and overtime deposits
- (f) Test for local overtime deposits
- (g) Connect and alert an operator when an additional deposit is required
- (h) Disable TOUCH-TONE service when connected to an operator to prevent fraudulent use of the TOUCH-TONE service signals
- (i) Time initial period on local and coin zone calls and overtime periods on local calls
- (j) Collect/return coins at disconnect
- (k) Provide disconnect on coin calls.

Coin calls are classified into three charging classes:

- (a) Local Coin Call: A call within the coin customers' local calling area.
- (b) Coin Zone Dialing: An arrangement whereby multiple message unit calls may be dialed from a coin telephone without the necessity of dialing a 0+ or 1+ prefix. Coin zone calls terminate outside the local calling area but generally do not qualify for toll rates.
- (c) Toll and Special Services: A toll call, is a call that is neither local nor coin zone and is usually routed through a TSPS office. Special service calls are directory assistance, repair service, 911, dial 0, etc.

Toll calls from coin lines require the assistance of either a toll operator or TSPS; hence, they are classified operator assisted.

LOCAL COIN

A. Operation

4.121 The origination stage of a local coin call is processed by the dialing connection program in a normal manner and the dialing stage of a local coin call is processed by the digit analysis program also in a normal manner. Immediately after the last digit is dialed, the digit analysis program transfers the call over to a coin charge program (COIN). The

COIN program first seizes a coin charge register (CNC) and initializes it. The CNC administers charging for the coin call. The COIN program then checks whether the coin station is dial tone first or coin first. If the station is dial tone first, COIN tests the coin station for presence of the initial deposit. The initial deposit test is not performed on coin first stations because coin first stations cannot produce an origination signal without the presence of an initial deposit. If the coin customer remitted the initial charge, COIN invokes either the ringing and answer detection programs or the outpulsing programs for connecting the coin line to the called line. If the coin customer did not satisfy the initial charge, the coin programs route the coin line to the insufficient deposit announcement which informs the coin customer to remit the proper rate and redial the call.

4.122 The COIN program regains control of the local coin call when the called party answers. After regaining control, COIN checks whether the call type is local coin with overtime or local coin without overtime. If the call type is local coin without overtime, COIN marks the call chargeable, releases the CNC and waits for the call to end. When the call ends, COIN collects the initial deposit and disconnects the call. The disconnect programs perform the bulk of the disconnect actions. If the coin customer hangs up before the called party answers, the initial deposit is returned and the coin line is disconnected.

4.123 Local coin with overtime call type consists of one initial period and zero or more overtime periods. Durations of both the initial and the overtime periods are specified by the operating company. Upon regaining control of a local coin with overtime call type following an answer, COIN marks the call chargeable and begins timing the initial period. If the call ends during the initial period, the coin program collects the initial deposit and disconnects the call. Disconnect of a local coin with overtime call type is handled entirely by COIN. The disconnect programs are not used.

4.124 Twenty-four seconds before the initial period expires, COIN collects the initial deposit. The COIN program then waits 24 seconds and tests the coin station for overtime deposit. If the overtime deposit is present, COIN begins timing the overtime period. If the overtime deposit is not present, COIN connects both parties to a local coin overtime announcement which prompts the coin customer for the overtime deposit. The COIN program waits another

24 seconds following the announcement and performs a second overtime deposit test. If the overtime deposit is present this time, COIN begins timing the overtime period. Otherwise, COIN disconnects the call.

4.125 Twenty-four seconds before the overtime period expires, COIN collects the overtime deposit, waits 24 seconds and tests the coin station for the next overtime deposit. If the overtime deposit is present, COIN program begins timing the next overtime period. Otherwise, COIN connects both parties to the local overtime announcement, waits 24 seconds following the announcement and performs the next overtime deposit test. If the test passes, COIN begins timing the next overtime period. If the test finds no deposit, COIN disconnects the call. The COIN program repeats this overtime period sequence for as many overtime periods (all overtime periods are of the same duration) as the call lasts.

4.126 When the call ends, COIN collects the overtime deposit and disconnects the call. This disconnect is also handled entirely by COIN.

4.127 An operating company may optionally use an overtime monitoring operator (OMO), in lieu of the local overtime announcement for overtime deposit prompting.

COIN ZONE

4.128 Origination and dialing stages of a coin zone call are handled in the normal manner. After the last digit, control of the call passes to the COIN program. The COIN program seizes and initializes a coin charge register, seizes a coin zone operator (CZO), reserves a network path between the CZO and the coin line, and connects the coin line to audible ringing tone. When the CZO answers, COIN program reswitches the coin line from the audible ringing tone to the CZO. The CZO informs the coin customer how much to deposit for the initial period and monitors the deposit. When the CZO is satisfied with the deposit, the operator disconnects from the call thus signaling COIN to complete the call. Upon receiving this signal, COIN disconnects the coin line from the CZO and invokes either the ringing program or the outpulsing program to connect the coin line to the dialed party.

4.129 The COIN program regains control of the coin zone call when the called party answers.

SECTION 231-045-005

If the coin customer hangs up before the called line answers, the initial deposit is returned and the call is disconnected.

4.130 Upon regaining control of the coin zone call following an answer, COIN marks the call chargeable and begins timing the initial period. Twenty-four seconds before the initial period ends, COIN collects the initial deposit. The COIN program then waits 24 seconds and connects both the coin customer and the terminating party to the CZO where the parties remain until the call ends. If the call ends during the initial period timing, COIN collects the initial deposit and disconnects the call.

4.131 The CZO measures the duration of the over-time period. When the coin zone call ends, COIN signals the CZO that the call ended. The CZO then computes the overtime charges, informs the coin customer (the CZO may have to ring the coin line) to deposit the computed charge and monitors the deposit. When the coin customer remits the correct amount, the CZO instructs the customer to hang up and disconnects. Upon detecting both the coin line and the CZO on-hook, COIN collects the overtime deposit and disconnects the call.

A. Toll and Special Services

4.132 This class of coin calls is not processed by the coin charge program but by programs providing special services. Toll calls from coin lines are routed either to toll operators or to TSPS with TSPS being the more common method. The TSPS/toll operators complete the call to the desired destination and perform all charging functions for the call. The operators determine the cost of the call, prompt the coin customer for the proper deposit, and collect the deposited coins.

4.133 A typical toll call flows as follows:

- (1) Following the last digit, the digit analysis program returns any deposited coins (this is an office option) and connects the coin line to TSPS.
- (2) The TSPS computes the cost of the initial period, informs the coin customer to deposit the computed charge, monitors the deposit for proper amount, and completes the call when the initial period charge is satisfied.
- (3) When the terminating party answers, TSPS begins timing the initial period.

- (4) Upon timeout, TSPS collects the initial deposit and informs the coin customer that the call has entered into overtime and that overtime period charges will be levied.

- (5) The TSPS then begins timing the overtime period.

- (6) When the call ends, TSPS computes the cost of the overtime period based on the measured duration, informs the coin customer to deposit the computed charge, monitors the deposit of coins for proper amount, collects the deposit and disconnects.

- (7) Upon detecting an on-hook from the TSPS, No. 1/1A ESS disconnect program disconnects the coin line from the TSPS.

4.134 The TSPS does not perform coin collects directly as the previous paragraph alluded to. Rather, the TSPS transmits a coin collect request signal to the No. 1/1A ESS switch. The TSPS service program in conjunction with the coin control (COCN) program in the No. 1/1A ESS switch performs the actual coin collect function. The TSPS has the capability to request coin collects, coin returns, and ringbacks. The coin return request is used to return the initial deposit when the terminating party does not answer. The ringback request is employed for recalling the coin customer back to the phone should the customer hang up at the end of a call instead of flashing.

4.135 Operator-assisted toll calls (such as person-to-person where only a "0" is dialed) are handled similarly to TSPS toll calls. The digit analysis program returns any deposited coins (an office may elect to retain the deposit and credit it toward initial period) at the end of dialing and connects the coin line to a general assistance operator. The operator obtains any further information required for completing and billing the coin customer and completes the call. The operator then times the duration of the call, computes the charges for the call, requests deposits from the coin customer and at appropriate stages of the call, requests the No. 1/1A operator programs to collect coins, return coins, or ringback the coin line. The operator programs in conjunction with coin control programs temporarily seize control of the call, perform the operator requested coin function, and return the control of the call back to the operator.

4.136 Special service calls such as directory assistance and business bureau calls from coin

lines are routed similarly to calls from noncoin lines with two exceptions. The first exception is at the completion of dialing. Any deposited coins may be returned, collected or retained at the telephone company option and on a per trunk group basis. The second exception (arises at the end of a coin call), occurs when any coins retained at the end of dialing are returned at the end of the call.¶

B. Coin Station Testing

4.137 Programs are also provided to enable maintenance personnel to test a coin station. When a special number is dialed from the coin station, the coin line is connected to a coin station test line which then tests the operation of the ground removal relay in dial tone first phones and may optionally also test:

- Loop resistance
- Coin collection
- Coin return
- Operating time of coin collect relay.

4.138 The programs provide coded rings to indicate the results of the tests. If the coin station is taken off-hook during these rings, the program prepares to receive a digit requesting another test. It also begins timing. If the craftperson does not request a test within 60 seconds, the call is disconnected. If a digit is received, the program initiates the test associated with that digit.

TRANSLATIONS SOFTWARE

A. General

4.139 The translations program and data package provide a means of translating information from one form to another. For example, a telephone is identified to a customer by a directory number. At the line side of the network, the line termination connecting this particular telephone is assigned a line equipment number. Translating from line equipment number-to-directory number or directory number-to-line equipment number is a primary task in many call processing activities. The translations system provides this and numerous other translating services.

4.140 The translation system consists of a data base and collection of control routines which

provide specific kinds of translated information on request from other programs. When the requesting (ie, client) program requires service, it provides the translation system with the necessary input information (Fig. 13). The translation data base, in simplex call store, is interrogated like a dictionary by the translation control routines. The translation output is found and delivered to the client in a predetermined fashion. The output may consist of the following:

- Addresses of equipment numbers
- Numerical quantities
- Generic translation data
- An output indicating a special situation.

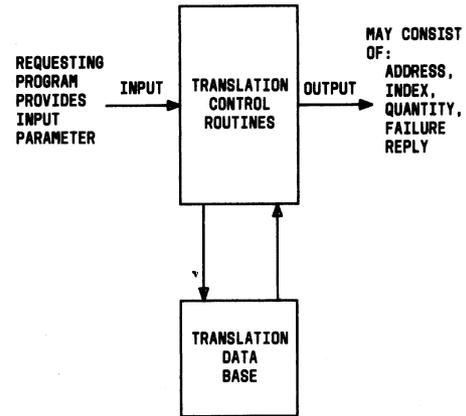


Fig. 13—Translation System

B. Addresses of Equipment Numbers

4.141 Addresses of equipment numbers are required by the client to do specific jobs—such as, operating a relay using a central pulse distributor or a signal distributor, reading a line or a trunk scan point, setting up network paths, etc.

SECTION 231-045-005

C. Numerical Quantities

4.142 Numerical quantities specify directory numbers, trunk group numbers, billing indexes, and route pattern numbers. These quantities have the same number of bits in all offices but may lead to different actions in each office.

D. Generic Translation Data

4.143 Generic translation data (translation data for determining call processing operations) has fixed numerical quantities; therefore, a particular binary number has the same meaning in all offices. For example, part of the output information from a line equipment number translation indicates if the line is for TOUCH-TONE service. A particular group of output data bits provides this information. These bits are interpreted in the same manner in every office by the call processing programs.

E. Special Output Situations

4.144 Special output situations are the result of finding a special condition when making the translation. For example, if the directory number is found unassigned, the call will be routed by the client program to intercept treatment. In this case, the return from the translation program is made to a different point in the client program than when an assigned entry is detected.

F. Translators

4.145 A translator consists of all of the data connected with a particular type of input. Because of growth considerations in any office, most translators are broken into subtranslators linked together by a head table. In most cases, the head table is referenced by the master head table. A subtranslator corresponds to a growth unit of the input. For instance, since the quantity of line equipment numbers in an office depends on the number of remreed line switch circuits, the line equipment number subtranslator contains translation information for 1024 line equipment numbers. In general, the translation input is divided into two parts: the subtranslator selector, and the subtranslator index. The selector is the number of the subtranslator in the head table, and the index is the number of the item (word number) in the subtranslator. In some cases, the translation information required may not fit into the limited subtranslator space (one translation

word) so auxiliary blocks and/or expansion tables may also be needed.

4.146 A full translator can consist of a head table, subtranslators, scattered auxiliary blocks, abbreviated codes, subauxiliary blocks, and associated expansion tables.

Locating Translators

4.147 A specific translator can be identified in memory by locating its address in the base translator block, ie, the master head table. The base translator block is located in simplex call store at a fixed Compool defined address and with a specified length. The master head table consists of the first 200 (octal) words of this fixed block. The master head table, containing addresses of other tables, serves as a directory for locating the translators.

Translator Types

4.148 Translation data exists in simplex call store as a structured collection of tables. The basic table entry is the 24-bit translation word. The set of tables devoted to the conversion of certain input data into certain output data is called a translator. Examples are line equipment translator, directory number translator, numbering plan area code and office code translators, trunk network number-to-trunk group number translator, trunk network number-to-peripheral equipment translator, etc. Each set of tables (a translator) is linked according to a hierarchic pattern. Tables high in the hierarchy contain pointers to (addresses of) lower tables. The lowest tables in the hierarchy (such as the subtranslator, the auxiliary block, or the abbreviated code expansion table) contain the actual translation data. Translators may be divided into two general types: multilevel and expansion.

Multilevel Translator

4.149 The multilevel translator consists of a maximum of six levels of information in a hierarchy. These levels are as follows:

Level	Name
1	Head table
2	Subtranslator
3	Primary translation word

- 4 Auxiliary block or expansion table
- 5 List
- 6 Subauxiliary block

4.150 The first level in the hierarchy is the head table. This head table serves as a directory for locating all translation data related to the input. This table usually contains an address of a subtranslator.

4.151 The second level in the hierarchy is the subtranslator. Each subtranslator corresponds to a growth unit of the telephone office. For instance, there is a subtranslator per line switch frame or circuit, per trunk switch frame or circuit, per 1000 directory numbers, etc.

4.152 The binary representation of the input to the multilevel translator is divided into two parts: the subtranslator selector and index. The selector and index can be visualized three different ways.

- (a) The selector identifies the unit and the index identifies the item within the unit.
- (b) The selector is the number of the subtranslator and the index is the number of the item within the subtranslator.
- (c) The selector is the word number in the head table which contains the address of the subtranslator; the index is the word number in the subtranslator which contains the primary translation word.

4.153 A subtranslator is a table which consists of one translation word per index. This word, the primary translation word, is the third level in the hierarchy and contains either the complete data associated with the input, or if one translation word is insufficient, an auxiliary address to an auxiliary block or expansion table (which is the fourth level). When the primary translation word contains an auxiliary address, the complete data associated with the input are contained in the auxiliary block words. To recognize an auxiliary address, the four most significant bits of the subtranslator word contain zero. Therefore, any subtranslator primary translation word for which the four leftmost bits are zero is interpreted as an auxiliary address.

4.154 The auxiliary address, in the primary translation word (PTW) is the address of the word

in the auxiliary block which contains the word number (WRDN). This WRDN either indicates the length of the auxiliary block if the block is 31 words or less or indicates that the block exceeds 31 words. The WRDN is located in the six most significant bits of the first word of the auxiliary block. When the block exceeds 31 words, the WRDN is 000000 and is located in the second word of the block. The true length of the block is then located in the word preceding the auxiliary block.

4.155 Another level, the fifth level of the translator, is the list (eg, speed calling list). The list is used when additional information other than the standard found in the auxiliary block is needed. The list, like the subtranslator, has an associated index. The address of the list may be found in one of the data words of the auxiliary block.

4.156 The last level is the subauxiliary block. The subauxiliary block is used if one word is insufficient to contain the information found in the list (eg, a 10-digit number for a speed call list is contained in a subauxiliary block). Therefore, like the subtranslator, the list may contain a subauxiliary address instead of the data usually found in the list.

Expansion Translator

4.157 The expansion translator has only one table, called an expansion table or simply a table. This type of translator has only an index as its input. The translator is no more than a list of words. Like the subtranslator in the multilevel translator, it may or may not have auxiliary blocks associated with it.

4.158 An example of the expansion translator is the unit type lengths table which is located in the base translator block. The unit type lengths table is a list of words containing the number of subtranslator translation words per unit type. The input is the unit type number. Therefore, to find the quantity of subtranslator words for any unit type, the unit type is used as the index. For example, for unit type 5 the quantity of units is found in the sixth word of the unit type lengths table.

4.159 The following translators are not pointed to by addresses located within the master head table or auxiliary master head table. Each of these translators has a fixed location in unduplicated call store which is defined in Compool:

- Master head table

SECTION 231-045-005

- Head table lengths table
- Unit type table
- Unit type lengths table
- Junctor circuit number-to-junctor network number head table
- Junctor network number-to-junctor circuit number head table
- Junctor network number-to-junctor network number/junctor circuit number.

Indexing into the master head table, unit type table, head table lengths table, and unit type lengths table translators is accomplished by utilizing fixed coding which is a function of the generic program algorithms.

Call Store and Program Store Memory Expansion Feature

4.160 The memory expansion program increases the memory spectrum to maximum size supported by the 1A processor to 20 million octal words. With the expanded memory size, call store is expanded to 32 K-codes and program store to 30 K-codes. Call store is split into three parts:

- (a) Duplicated call store (DCS): Contains transient information
- (b) Lower unduplicated call store (LUC): Contains translation information
- (c) Higher unduplicated call store (HUC): Contains translation information.¶

RECENT CHANGE (RC) SOFTWARE

A. General

4.161 The RC subsystem (RCSS) consists of a collection of recent change message interpretation programs designed to accept recent change (RC) input messages, check them for validity, and generate the proper translation data for storage in translation memory. The RCSS provides maintenance and service order with a means of modifying the translation data to reflect changes in customer service requirements and in the internal system. Figure 14 provides a generalized system diagram of the RCSS.

B. RCSS Functional Description

4.162 Inputs to the RCSS may occur simultaneously from up to four different sources:

- Customer lines
- The service order activation phone
- The office RC channel, and
- The service order TTY (includes paper tape or magnetic cartridge)
- Maintenance TTY.

These requests for RCSS service may originate for any number of reasons and, in the case of inputs from customer lines, are not under central office control. However, the end result in each case is that the translation data is dated to reflect requested changes.

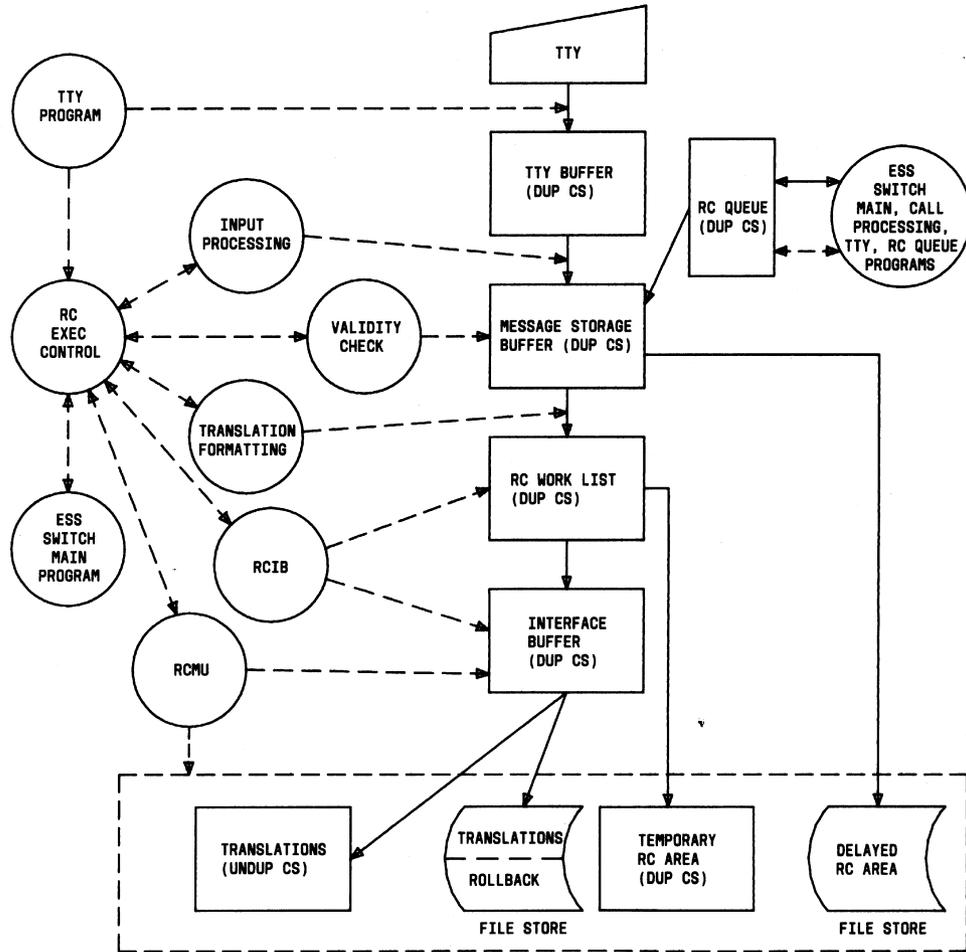
C. Translation Data Storage

4.163 Translation data is stored in unduplicated (ie, simplex), call store. Recent changes with permanent status are stored in the translations data area of unduplicated protected call store with two complete backup copies stored on disk in the file stores in No. 1A ESS switch. Recent changes that are temporary (eg, call forwarding changes) are stored in the temporary RC area in duplicated call store. File store also provides for storage of delayed recent changes and translation rollback data, ie, a copy of the original translation data (before modification) of every translation word affected by a permanent status RC.

D. Queuing of RC Inputs

4.164 Speed calling requires the RCSS to handle an increasingly large volume of inputs. The basic philosophy is that customer originated requests for changing a speed call list entry will be queued in a searchable queue until the RCSS is free to process the entries.

4.165 The RC queue consists of a dedicated block in duplicated call store whose size (length in call store words) is defined in parameters and, thus, is office engineered. The first word in the queue is reserved for TTY input requests. Should a request from the TTY be received while the RCSS is busy pro-



LEGEND:
 DATA ————
 PROGRAM CONTROL - - - -
 RCIB - RECENT CHANGE INTERFACE BLOCK BUILDER
 RCMU - RECENT CHANGE MESSAGE UPDATE

Fig. 14—No. 1A ESS Switch Recent Change Subsystem

SECTION 231-045-005

cessing a queue entry, the TTY channel number is loaded into the first queue word position. The RCSS scans the queue from top to bottom, thereby giving the highest priority to the TTY. Only the TTY channel number is held in the queue, as character buffering is accomplished in the TTY buffer. The remaining queue entries are processed in the order of their occurrence in the queue. Conversely, the queue is read from bottom to top in response to a request from the speed calling program. This ensures that the most recent information is returned for call processing purposes; eg, it is possible that two queue entries for the same speed calling list entry could exist at the same time. Reading from bottom to top yields the most recent information.

E. Customer Line Inputs

4.166 The queue control program in the RCSS interfaces directly with the call processing programs which interpret speed call list changes from customer lines. When a customer dials the proper access code to change a speed call list entry, the call processing program decodes the request and stores the change information in a holding register. The RCSS is then requested to load the change information into the RC queue. The RCSS then ensures that the request is valid and can be loaded into the queue. Once the queue entry is processed, the speed call list change is active and can be accessed as necessary by the speed calling program.

4.167 At the completion of processing a queue entry, the RC queue is checked for a waiting TTY request for service. If no TTY requests are present, the next queue entry is unloaded and processed. After processing an RC message from the TTY, the RCSS will always check the queue for any entries waiting to be processed, delete the entry that was just processed, and move the remaining entries up.

F. Inputs From the Office RC and Service Order TTYs

4.168 Most of the changes to the translation data in the No. 1A ESS switch memory will be the result of RC messages input from the TTYs, either manually or via paper or magnetic tape. RC messages entered via the TTY may have either immediate or delayed status. Unless entered as a delayed message, the RCSS will immediately process the message, format new or modified translation data, and then update the translation data in call store. If the RC is of a permanent type, the translation data in undupli-

cated call store is changed. The translation data image in file store is then brought up to date.

G. Rollback

4.169 There can be instances where an RC message or several RC messages affect the translation data in such a way as to degrade or disrupt system activities. To correct problems of this nature, it is necessary to "undo" the RCs suspected of causing the problems by reinstating the original translation data, as it appeared before the RC message was input. This operation is termed rollback.

4.170 Each time an RC message affecting the permanent translation data in unduplicated call store is entered, a rollback block (RBB) is built by the RCSS and stored in the rollback area on disk. The disk rollback area is a dedicated block of disk storage whose length is Compool defined. Each RBB, identified by an internally generated RC order number, contains the address and original data of each translation word affected by the associated RC message.

H. Rollback Area Management

Type I Rollback

4.171 Each RC message that is accepted by the RCSS and is to be inserted into the translation data base creates a single RBB in the rollback area. Each RBB contains the address and old data for each word of translations altered by the RC. The RBB is created and stored in the file store rollback area before any updating of the translation data takes place. This is done in case the update process is for any reason interrupted (eg, interrupt, disk return failure, etc). If the update should fail before the entire update is complete, the RBB data is used to remove the partially completed RC. This kind of rollback is referred to as Type I Rollback. Type I rollback is automatically initiated by the RCSS to remove a partially completed RC.

Type II Rollback

4.172 Type II rollback is the process of removing or canceling RCs which successfully update translation data and are effective with respect to call processing. Type II rollback must be manually requested either from the TTY or via the master control center (MCC).

Rollback Data Storage Concepts

4.173 The rollback area is logically circular; ie, the physical beginning and end of the rollback area act as though they were adjacent memory locations in a ring buffer. Thus, as RBBs are stored and the rollback area fills up, additional RBBs overlap the first RBBs stored causing that particular rollback data to be lost. To avoid this overlap, office personnel regularly write a new translation tape. This tape is a "snapshot" of translations data at that particular point in time. When this tape dump is performed, all RBBs which were created prior to the old latest system tape are discarded, freeing up a large (approximately 40 percent) portion of the rollback area. Associated with the rollback area are three indexes or pointers: TAPE1, TAPE2, and NEXT. The NEXT index points to where the next RBB will be stored when a new RC message is successfully processed. The TAPE2 index shows where the NEXT pointer was when the latest translation tape was written. The TAPE1 index, likewise, marks the position of the NEXT index when the next to the latest translation tape was written.

4.174 If the latest translation tape dump had been followed by RC messages 13, 14, 15, 16, and 17, then to restore translations to the state that existed when the latest tape was written, it is necessary to rollback RCs 17, 16, 15, 14, and 13. This is a rollback to TAPE2. If RC 7 was the first RC following the TAPE1 index, then to restore translations to the state that existed when the next to the latest tape was written, RCs 12, 11, 10, 9, 8, and 7 must also be rolled back. This is a rollback to TAPE1. It is not possible to rollback beyond the time of the writing of the next to the latest tape; ie, RBBs 6, 5, 4, 3, 2, and 1 (those preceding the TAPE1 index) cannot be rolled back and should be considered nonexistent.

4.175 When a new translation tape is written, the rollback area tape indexes are repositioned to reflect the existence of a new translation tape and that the old latest translation tape is now the next to the latest translation tape. Assume that a new translation tape has just been written and that no RCs have been input since. The TAPE2 index is now pointing to the same position as is the NEXT index and the TAPE1 index is moved up to the old TAPE2 position. At this point, a rollback to TAPE2 (the latest translation tape) would have no effect since no RCs have been input since the latest translation was written. A rollback to TAPE1, however, could be done. This

would rollback the RCs existing between the TAPE1 and TAPE2 indexes.

4.176 It is necessary to write new translation tapes on a regular basis to avoid filling up the rollback area and overtaking the TAPE1 or TAPE2 indexes. The frequency of writing translation tapes depends on the RC activity level in a particular office. Tape dumps may have to be performed every other day for some offices and once a week for others. The rule is that when the portion of the rollback area used since the latest tape dump reaches 40 percent, a tape dump should be performed as soon as practical, eg, that same evening. The percentage of the rollback area used since the latest tape dump is reported by an "RC:WARNING" output message when the percentage exceeds 40 percent and by the REPT:RC CENSUS output message as an hourly printout or in response to the "OP:RCCENSUS" input message. Since no indication is given of how much of the rollback area has been used since the next to the latest tape was written, the 40 percent rule must be strictly adhered to or the TAPE1 index can be overtaken without warning. Should the TAPE1 index be overtaken, it is moved up to coincide with the TAPE2 index with a resulting "REPT:RC WARNING" output message stating that a rollback to the next to the latest tape (TAPE1) is no longer possible. If additional RCs are entered beyond the 90 percent mark until the TAPE2 index is overtaken, the TAPE1 and TAPE2 indexes will be moved forward as many RBBs as necessary to store the new RBB. A "REPT:RC WARNING" output message will state that complete rollback to the latest system tape (TAPE2) is no longer possible. A partial rollback to TAPE2 can be performed, but the rollback is limited by the TAPE2 index which has been moved and no longer represents the state of translations that existed when the latest translation tape was written.

4.177 Type II rollback can be performed via the MCC with a Phase 4, 5, or 6 or by means of a TTY input message without going through a phase. There are four modes of Type II rollback by MCC request:

- Rollback to a particular order number
- Rollback to a specified number of orders
- Rollback to the TAPE2 index, and
- Rollback to the TAPE1 index.

4.178 Type II rollback via the TTY is done without a phase; of course the ESS switch must be

SECTION 231-045-005

sane enough to allow input/output via a TTY channel for this method to work. TTY rollback is restricted to 20 orders per rollback request, and only as far back as the TAPE2 index; ie, TTY rollback is limited to a rollback to the latest translation tape.

4.179 Each RBB has an internal order number assigned to it before being stored in the rollback area. This RCSS assigned order number is completely independent of the user specified order number entered with the ORD keyword in the RC message. The internal order number is of the following form:

MMDDSSSS

where MM = month (1-14s); DD = day of month; SSSS = daily sequence number (0-7777s). The internal order number, in addition to providing a unique identifier for each RBB, also gives information on when the RBB was created and stored in the rollback area.

4.180 Each day at midnight, the date is updated and the sequence number is reset to zero. This numbering scheme is useful in that it defines what can be called "safe rollback points." Assuming that all RC activity is halted around the hour of midnight, due to the working hours shift change and that translations is in a valid state so far as can be determined, it should always be safe to rollback to midnight of any particular day. This is done by rolling back through order number MMDD0000.

4.181 For example, assume that all RC activity for June 26 ended at 10:00 p.m. and the system seemed to be in good shape. The next day RC activity resumed and at 3:20 p.m. that afternoon serious system problems began to appear with indications of translation data mutilation due to RC errors. The decision is made to rollback RCs to clear the problem. Rollback of an arbitrary number of RCs is dangerous because the resulting state of translations at the completion of the rollback is not known. Rollback to TAPE2 or TAPE1 is maybe too drastic, possibly causing disruption of service to a large number of customers. Rollback to midnight of the previous day, a known safe point, is recommended in this situation. An MCC rollback in a phase 4, 5, or 6 through internal order number 06330000 (octal) is performed without delay resulting in the removal of the bad RC message(s). When the bad RC is isolated and corrected, the day's RC messages can be reentered using the

paper tape or magnetic cartridge backup. But, as a result of the rollback, speed calling list changes will be lost and each affected customer will have to reestablish his list. Call forwarding RCs are kept in the temporary RC area, produce no rollback data, and consequently, are not affected by rollback. The example illustrates the importance of the rollback data to extricate an office from service disrupting problems caused by faulty translation data. It should emphasize the importance of keeping the rollback data clean.

4.182 An easy way to generate messy rollback data and run the risk of generating translation data errors is to try to fix a bad RC with additional RCs. The use of "RC:PSWD" is often abused in this way. The proper way to correct a bad RC is to rollback the order and then reenter it correctly.

I. Delayed RCs

4.183 A delayed RC message may be entered by specifying DELAY in the message format. The entire message is processed normally, including syntax and data checks, but no translation data updating is done. Instead, the message is compressed and stored in the delayed RC area in file store. The message is not effective until the RC message is activated. Activation is accomplished by another RC message from the TTY or via the service order activation phone.

J. Inputs From the Service Order Activation Phone

4.184 A central office telephone set whose major class of service is activate service order may be used to activate delayed RCs. The delayed RC is activated by dialing the order number, nnnnn, filled out to five digits with leading zeros. If no delayed message with the dialed order number is found, reorder tone is heard. If the RCSS is busy and cannot process the activation code, busy tone is heard and the user may try again. If the delayed message associated with the dialed order number is found in memory, dial tone is returned and other order numbers may be dialed as appropriate.

Note: Receipt of dial tone does not guarantee that the delayed message was activated.

The TTY should be checked for an ACPT RC18 output message indicating successful activation.

QUEUE AND GENERAL PURPOSE SOFTWARE

A. Queue Administration Program (WQUE)

General

4.185 While processing a call, the system may encounter various busy conditions. When these conditions occur, the call is delayed by having it wait in a queue or list until the register or hardware necessary to continue the processing of the call becomes available. The queue administration program (WQUE) loads the various queues when requested by client programs due to busy conditions, and unloads

the same queues when the equipment or registers are available. If a call is abandoned or a time-out has occurred, the item can be removed from a queue at the request of the client program.

4.186 Subsystems interfacing with the queue administration program are shown in Fig. 15.

Functional Description

4.187 This program administers two types of queues: the fixed length queue and the variable length queue. A fixed length queue is a call store block of memory engineered to a length dependent

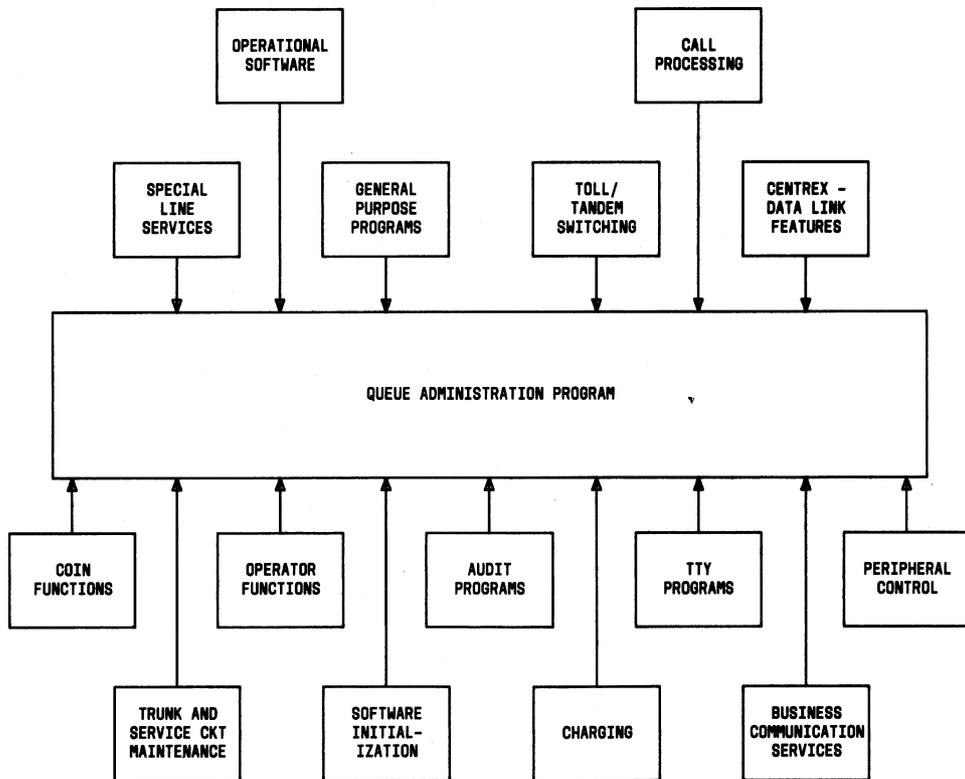


Fig. 15—Subsystem Interfacing With Queue Administration

SECTION 231-045-005

upon the office size and traffic requirements. This block contains information on customers unable to be served due to the unavailability of equipment. Each fixed length queue has a 4-word head cell: two words for loading information (load pointer and load bottom pointer), and two words for unloading information (unload pointer and unload bottom pointer).

4.188 A variable length queue is a 1-way list of call registers which have been linked together waiting for release of equipment, release of call store memory, a time-out, etc. This type of queue requires a 2-word head cell. The first word contains the address of the first client register on the queue, and the second word contains the address of the last client register on the queue. The linkage is completed via the queue word of each register which contains the address of the next register on the variable length queue.

4.189 When an item is loaded on either a fixed length queue or a variable length queue, the associated executive control main program flag for the specific queue is set, indicating that an item is now on a specific queue. With the flag bit set, the executive control main program returns periodically to the unload entry for the queue until the queue is empty, at which time the flag bit is turned off.

4.190 As a result of an abandon, a time-out, etc, a client can be removed from a queue before the executive control main program unloads the entry. In such cases, an entry on a fixed length queue is removed by zeroing the location in the queue. Linkage to the location is established via the client register's queue word which contains the address of the client register entry on the fixed length queue. An entry on a variable length queue is removed from the queue by searching from the beginning of the list until the client register address to be removed matches the contents of the queue word of one of the registers on the list. At this point, the list is updated by substituting the next client register address on the list for the one just removed from the queue word.

B. General Purpose Programs

General

4.191 General purpose programs perform preliminary work and particular functions for client programs. Some of the general purpose programs provide service routines which return control to the

client programs when execution is complete. Others perform jobs of considerable magnitude and remain in control of a call during real-time breaks.

Functional Description

4.192 Following is a brief description of the major functions performed by the general purpose programs.

◆ Supervision Modernization (SUPERV)

4.193 The SUPERV programs deal with the analysis and delivery of call control signal to client programs. Signaling information and input/output structures are controlled by these small group programs, resulting in simplification of call processing functions. These programs also allow the introduction of a nonscanner controlled signaling system, such as CCIS, without requiring a dedicated software interface.

4.194 The SUPERV programs are concerned with:

- (a) Control and reporting of trunk seizures and answers
- (b) Control and reporting of line and trunk disconnects
- (c) Control and reporting of line and trunk reanswers
- (d) Processing of operator wink, flash, and other information signals.◆

Incoming Trunk to Busy, Overflow, or Special Service Circuit Program (YFTO)

4.195 Other programs use the YFTO program to connect an incoming trunk to busy tone, regular overflow tone, common overflow tone, or a special service circuit trunk such as high tone, low tone, announcement, or milliwatt test circuit. Upon entry to YFTO, a network path involving the incoming trunk may or may not exist. If a path does exist, the YFTO program will abandon and erase the path involving the incoming trunk and then connect the trunk to a tone or special service circuit trunk. Upon successful completion of the connection, supervisory scans for detecting the disconnect of the incoming trunk are initiated.

4.196 If the requested connection is to a busy tone trunk and a blocked or busy condition is en-

countered, the YFTO program attempts a connection to common overflow tone. If the requested connection is not to a busy or an overflow tone trunk, an attempt is made to connect to common overflow tone if a blocked condition is encountered, or to a busy tone if a busy condition is encountered. If a blocked or busy condition is found during an attempt to connect to common overflow tone or regular overflow tone, the incoming trunk is placed on High and Wet until it disconnects.

Originating Line to Busy, Overflow, or Special Service Circuit Program (YTTO)

4.197 The YTTO program is used by other programs to connect an originating line to busy tone, regular overflow tone, common overflow tone, or a special service circuit trunk such as high tone, low tone, announcement, or milliwatt test circuit. Upon entry to YTTO, a network path involving the originating line may or may not exist. If a path does exist, the YTTO program will abandon and erase the path involving the originating line and then connect the line to a tone or special service circuit trunk. Upon successful completion of the connection, supervisory scans for detecting the disconnect of the originating line are initiated.

4.198 If the requested connection is not to an overflow tone trunk and a blocked or busy condition is encountered, the YTTO program attempts a second connection—this time, to common overflow tone. If a blocked or busy condition is found when an attempt is made to connect to common or regular overflow tone, the call is abandoned and all memory associated with the call is idled. This will result in the originating line receiving dial tone if it remains off-hook.

Scan of Single Master Scanner Point Program (YFDS)

4.199 Client programs use the YFDS program to determine if a scan point on a master scanner is busy or idle. The client program provides the master scanner number of the scan point to be scanned at entry to YFDS. The scanner number is expanded to determine the enable address, the row, and the answer bit position of the scan point. Orders are sent on the peripheral bus to read the scan point. Upon completion of the scan, the YFDS program notifies the client program of the scan point condition (busy or idle).

Report and Miscellaneous Subroutines (COPR)

4.200 The COPR program consists of numerous subroutines which perform various functions in connection with call register administration. Services provided for client programs include the following:

- (a) Link, in a circular linked list, a specified register which is not already linked to the list to a specified register which may or may not be already linked.
- (b) Unlink a register from a circular linked list, release the register and restore it to its idle linked list, and pass on a report of a change in state to the next register on the linked list.
- (c) Accomplish charge delay timing for master registers. This timing is required on all coin charge, hotel-motel charge, and AMA message rate-without-overtime calls.
- (d) Grant or deny clearance to a requesting nonmaster register.
- (e) Set supervisory scan points upon return from a successful peripheral order buffer execution.
- (f) Set disconnect supervision on a line-to-line, line-to-trunk, or trunk-to-trunk connection.

Seize and Release Routines and L-, J-, and T-Bit Administration (YAHA)

4.201 This program seizes a register from its idle linked list or releases a register and restores it to its idle linked list for client programs. The kinds of registers provided for include call registers (originating, disconnect, etc), junior registers (MF junior, etc), and auxiliary memory blocks (path memory annex registers, etc).

Register Linking Routine (YCLK)

4.202 This program links a given register to a single register or to a circular linked list of call registers on a single call.

Miscellaneous Register Subroutines and Tables (YMRG)

4.203 General purpose program YMRG consists primarily of linked list register hunt rou-

SECTION 231-045-005

tines, register identification-program tag (RI-PT) routines, and general release register routines.

4.204 The linked list register hunt routines refer to the parameter area data in program store to deliver to a client program, one after another on repeated entries, the addresses of all registers of a specified type. These routines are used primarily by audit programs in their checking or rebuilding of register linked lists but are also used by call processing programs.

4.205 The RI-PT routines are used to select a return address to a call processing program on the basis of information stored in the state word of a call register. The reason for the return, that is, the type of information conveyed, determines which of four methods of return is used:

- (a) The disconnect method is used to report a disconnect (ie, scan point change from off-hook to on-hook).
- (b) The timing method is used to report the removal of a call register from a timing linked list upon time-out.
- (c) The queue method is used to report the removal of a register from a queue.
- (d) The standard method is used to report any expected event except the three preceding cases. For example, a change from on-hook to off-hook is reported by the standard method.

4.206 The general release register routines are designed to release a single register or all registers in a circular linked list of call registers on a single call, or all registers but one in such a circular linked list.

Call Store Zeroing Program (ZERO)

4.207 This program is used to rapidly zero a block of adjacent call store locations. Client programs may enter program ZERO at any point to zero the appropriate number of call store words.

C. Call Trace Program (TRCE)

General

4.208 The call trace program uses information in call store to find the terminal (line, trunk, or

service circuit), if any, which is connected to a known terminal. On failure to find the sought terminal in a connected path, the trace program gives an indication of the state of the known terminal in the system.

4.209 Call trace may be requested via various software functions or via TTY input messages to identify a line equipment number or trunk network number if it is connected to either a known directory number or a known trunk.

Functional Description

4.210 This program is functionally divided into two routines, one of which is used to find the terminal connected to a line, and the other to find the terminal connected to or reserved for a trunk.

Call Trace for Lines

4.211 The input to call trace for lines is a valid line equipment number or remote equipment number (for remote switching system). After determining that the given line is marked busy, the trace program searches for the line equipment number in the path memory associated with the given line link network.

4.212 When the line equipment number cannot be found and the line bit is busy, flags are set for audits to investigate and, if possible, correct the state of the line bit and path memory.

Call Trace for Trunks

4.213 The input to call trace for trunks is a trunk network number. If the trunk is in a stable and valid path, the program hunts for the other terminal.

4.214 Failure to find the trunk network number on a list or in a busy state implies that the path memory for trunks is not correct. A flag is set requesting an audit which restores the trunk to the trunk idle linked list, if necessary, and updates path memory.

D. Queuing Calls to Multiline Hunt Groups

General

4.215 An incoming call directed to a multiline hunt group is immediately serviced if there are

any idle hunting group terminals. If all terminals are busy, the call is placed on a queue and waits its turn. While waiting, the calling party receives audible ringing tone. The call that has been on the queue the longest will be the first call served when a line becomes available. (The number of calls that can be queued for a multiline hunt group depends on the number of queuing registers provided for that group.) Up to three call waiting lamps may be provided on the customer's premises for the purpose of informing the customer of the length of time calls are on the queue before being answered. If an incoming call cannot be placed on the queue, the calling party will receive busy tone unless the incoming call is a foreign exchange call. In this case, the new call is placed on the high and wet list (not connected to busy tone) to prevent charging.

Functional Description

4.216 Following is a brief description of programs functional in the queuing of calls to multiline hunt groups.

- (a) Queue and Administration Processing (QAPR): This program contains basic routines responsible for determining if a call should go on queue and if there is room on the queue for a call. In addition, these routines start the process to give tone to the customer. The QAPR program also contains general queue unloading routines. Other routines, which are entered from ECMP every flagged class D, perform on-queue processing functions, determining if there is an idle agent and, if so, taking a call off the queue and terminating the call to the idle agent. Additional routines detect time-outs and disconnects.
- (b) Give Audible, Disconnect, and Line Termination Routines (QTAL): This program contains routines which, depending on the type of origination, call the proper routines to perform switching, determine what type of tone or announcement the customer gets, and finish putting the call on queue. For a call that fails to get the necessary facilities when coming off the queue, this program contains routines to give reorder to the customer, given the type of incoming facility. Other routines prepare the call coming off the queue terminating to an intraoffice call to flow into the normal legs as if the call were never queued. There are a number of general routines that perform such functions as unlinking and releasing ringing registers and selecting audible route.

- (c) Customer Interface and Special Auditing Routines (QCIA): This program contains a routine to determine the oldest call on the queue. It also performs special auditing functions.

- (d) Queue State Information Features (QSIF): This program contains a routine that places into the queuing register the time that a call was placed on queue. In addition, there are routines connected with the call waiting lamp feature that determine states and send lamp orders.

SPECIAL SERVICES SOFTWARE

A. General

4.217 With the advent of stored program control to provide telecommunication services and control of switching networks, many special services have become available to the telephone customer. As with any programmable system, the services available are determined by the particular programs provided. Thus, the services available for any particular ESS switch central office are primarily dependent upon the software and vary from office to office. (See Fig. 16.)

4.218 The special service software can be separated into three general groups:

- Centrex-Data Link Services
- Business Communication Services
- Special Features.

B. Program Description

Centrex-Data Link Services

4.219 Centrex Console Lamp and Key Program (CXIO, CXKY): Provide all the normal (nonmaintenance) input/output functions for the centrex data link. The programs also interrogate key signals to determine their effect on the centrex console and any call on the console. These programs also investigate supervision reports.

4.220 Centrex Attendant Line and Trunk Seizure Program (CXLO): Provides control for seizing lines and trunks to establish a connection between the calling party and the centrex attendant.

4.221 Centrex Trunk Code-Call Answer Program (CXTA): Entered when digit analy-

SECTION 231-045-005

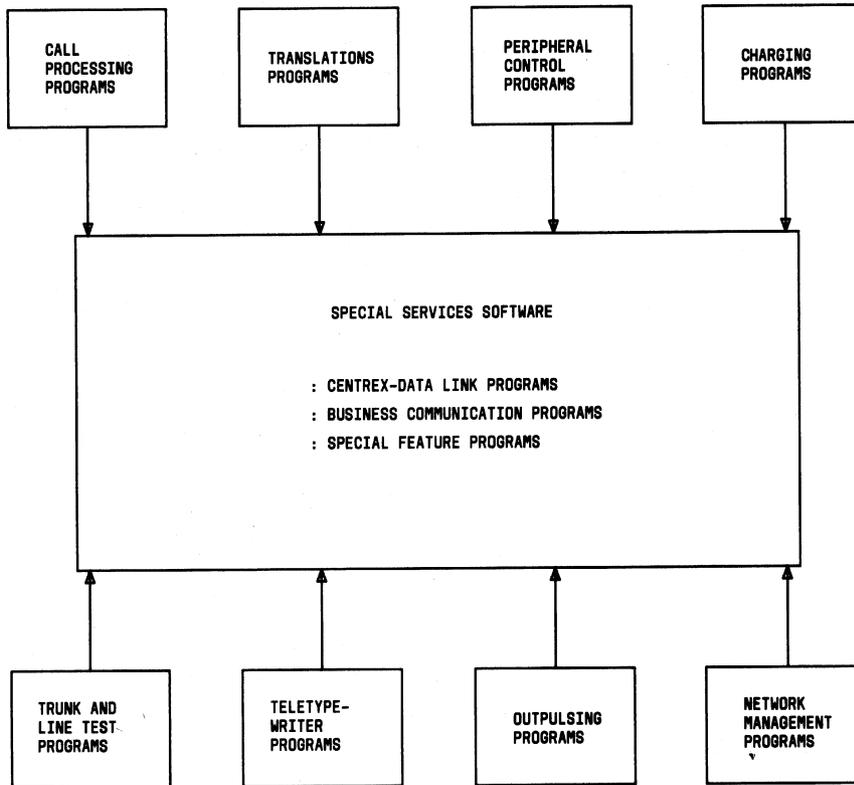


Fig. 16—Special Services Program Interface

sis programs recognize that a code call answer code has been dialed. The program then activates signaling devices (bells, horns, etc) to sound the called party code.

4.222 Centrex Console Lamp Control Program (CNLP): Provides an interface between the central office and the attendant console. As the state of attendant calls progress, this program transmits orders to and from the central office to administer particular console lamps and activate switching functions.

Business Communication Services

4.223 Centrex Simulated Facilities Program (CXSF): Administers usage counts and associated centrex console trunk-busy lamps after the program has determined that a call has reached the maximum allowable number of active calls that are being monitored for the specific simulated group.

4.224 Hotel-Motel Program (HMTL): Records the number of message units to be charged to a guest each time they place a call.

Rotary Coaxial Cable Connection Experiments

Overview

From time-to-time the amateur SIGINT enthusiast may require an antenna system that is fully rotatable and immune to tangled coaxial feed lines. The usual solution to this problem is to purchase a commercial rotary in-line coaxial cable adapter, which actually do exist. The problem with these adapters is that they are expensive, very hard-to-find, and may also be frequency limited.

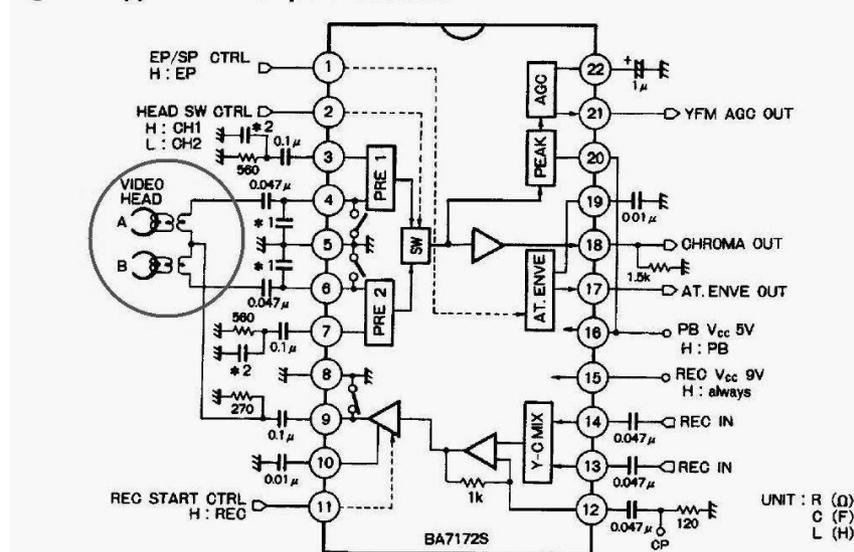
Our solution is to build our own using only commonly available parts. Our rotating coaxial adapter will be built around the spinning ferrite transformer inside the rotating drum head in all VHS VCRs. Every VHS VCR in existence has this type of rotating head, so supplies should be no problem. The rotating VCR drum head's construction and operation is quite ingenious. A small coil of wire, which makes up the actual playback/record head, is located inside the VCR's drum. This head (2 or 4 per drum) is used to read or write the low-frequency signal on the VCR cassette tape itself. Since the drum is rotating at around 1,800 Hz, and in order to transfer the signal from the heads to the rest of the VCR circuitry, the drum itself is made up of two rotating ferrite-coupled transformers. The head's schematic looks just like any 1:1 coupling transformer, just that the core is capable of being rotated a continuous 360 degrees.

The main drawback to using rotating VCR drums is their poor high frequency response and high insertion loss. Testing showed a limited frequency range below 10 MHz. This should be O.K. when used in downconverted Intermediate Frequency (IF) applications. Also, the drum's coupling loop windings were not impedance matched to 50 ohms, which probably accounted for most of the signal loss. With a little tweaking, a series-shunt, dual-capacitor impedance matching network could be used to match the transformer to 50 ohms over a limited frequency range.

In the below application example schematic for the Rohm BA7172 2-Channel Video Signal Amplifier IC, you can see that the VCR heads are actually little transformers with their core being rotating ferrite material. Note that the two VCR heads share a common output connection. These wires will need to be unsoldered and isolated for this experiment.

BA7172S, BA7172FS Pre-amplifier and recording amplifier

Figure 1 Application example 1—BA7172S



Construction Notes & Pictures



Several different types of VHS VCR rotating drum heads. These are from VCR models made between 1993 and 2000. They'll all probably be very electrically and mechanically similar.

Get a whole bunch of old VCRs to play with and take apart. They are filled with lots of nice, salvageable components.



Bottom view of the drum.

The drum's motor windings and electronics can be removed. They are usually mounted to the spindle with two screws.



Top view of the drum.

The bottom drum section is meant to be stationary while the top drum half, which contains the actual playback/record heads, spins at a rate of 1,800 Hz.

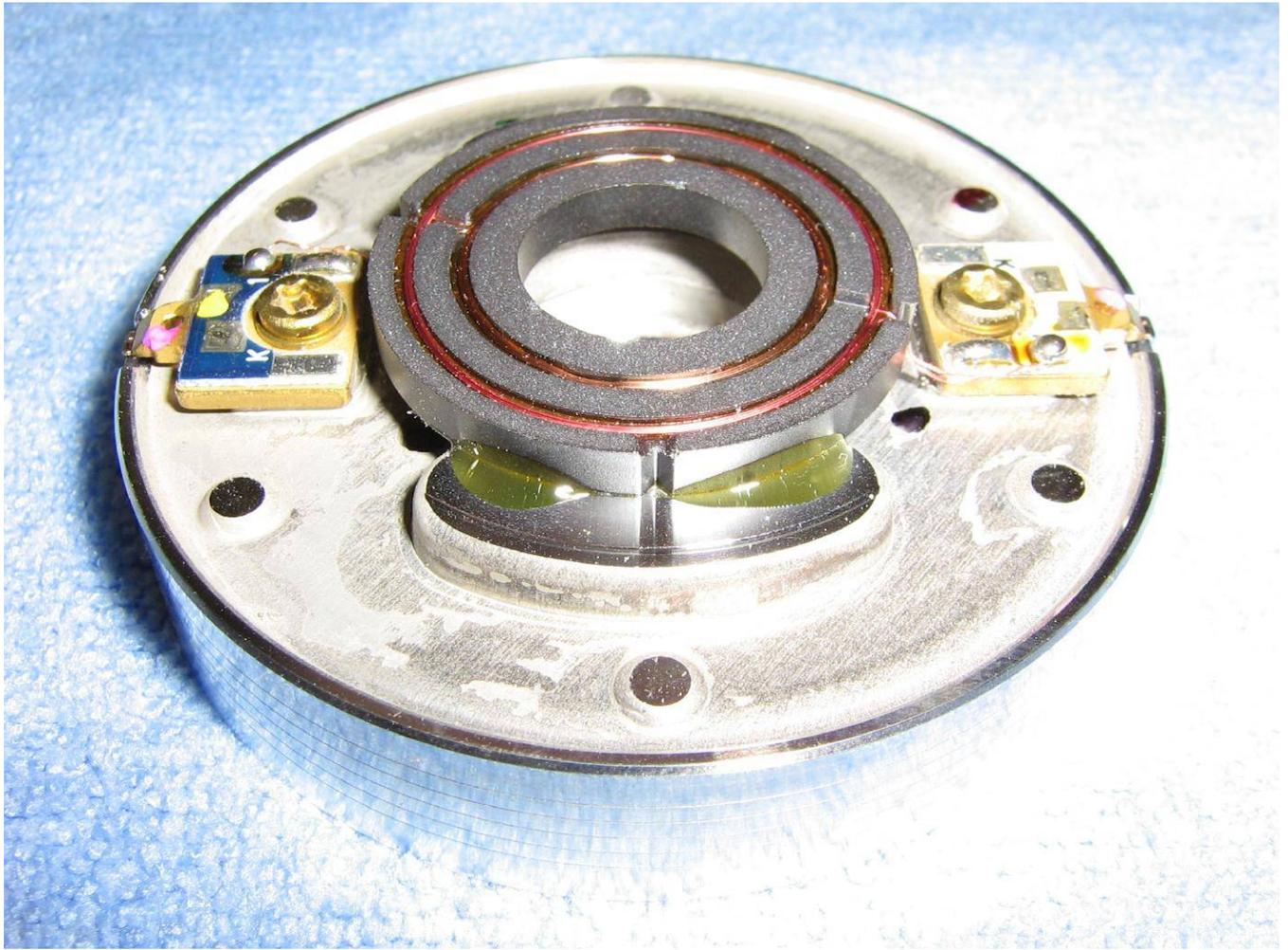


Separate the two halves.

Note that each half contains a loop of wire mounted in a piece of ferrite material. This is what makes up the actual rotary transformer.

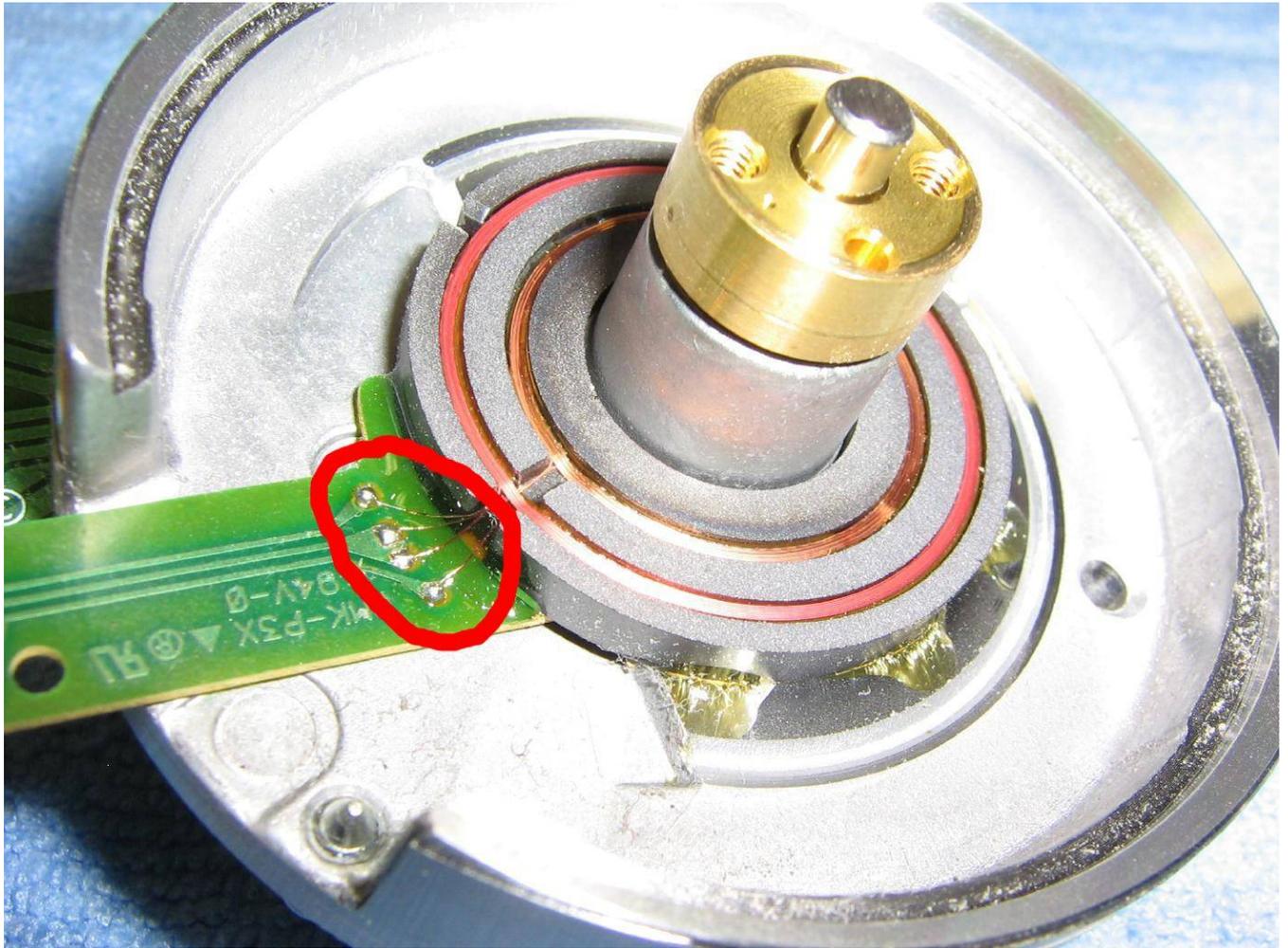


There is usually an Allen screw and clamp on the spindle that will need to be removed.



Closeup of the top rotating half.

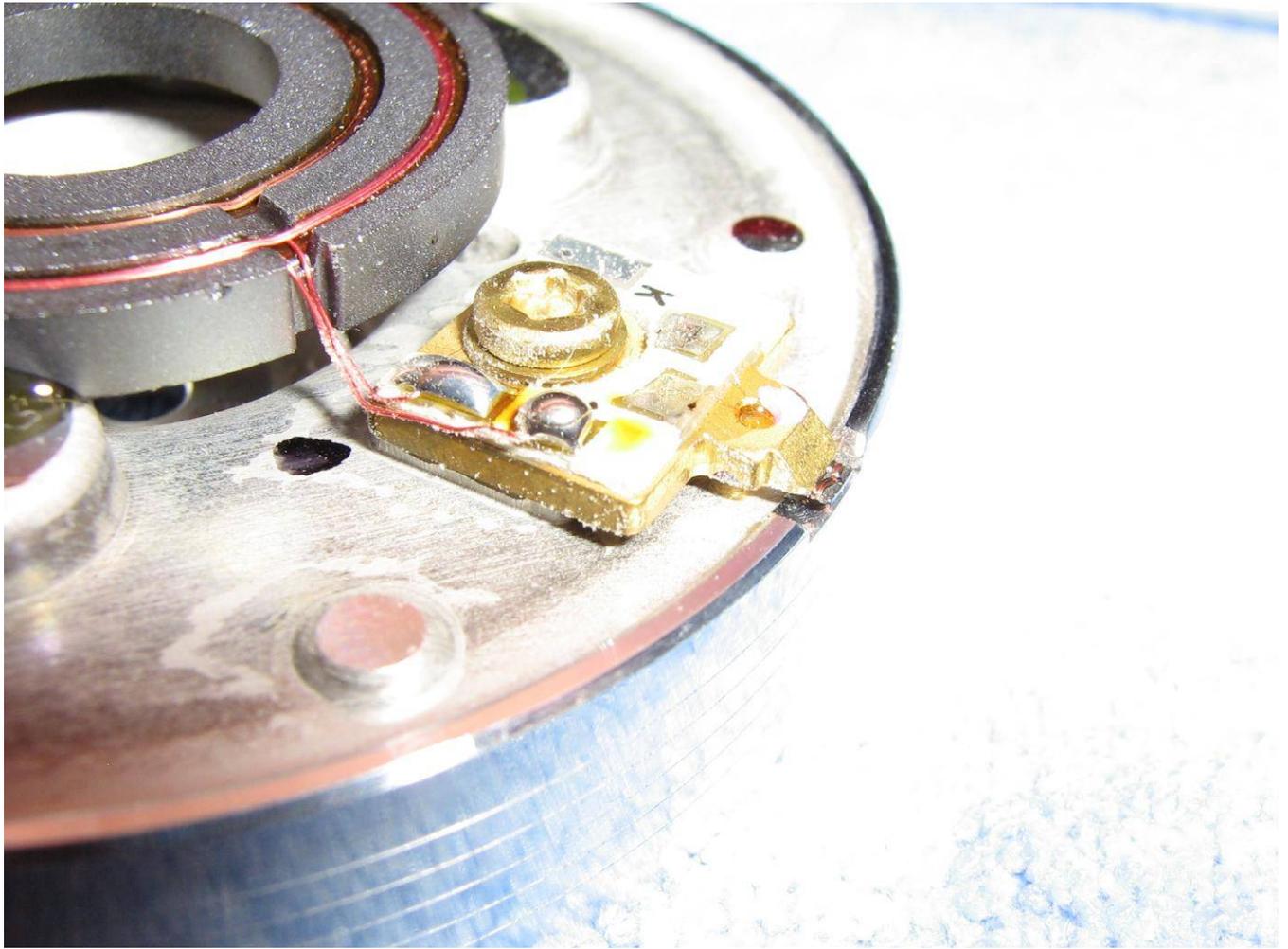
The actual playback and recording head windings are soldered to the little brass plates on each side. This is a two-head VCR design.



Closeup of the bottom stationary half.

Note how the two separate windings share a common connection.

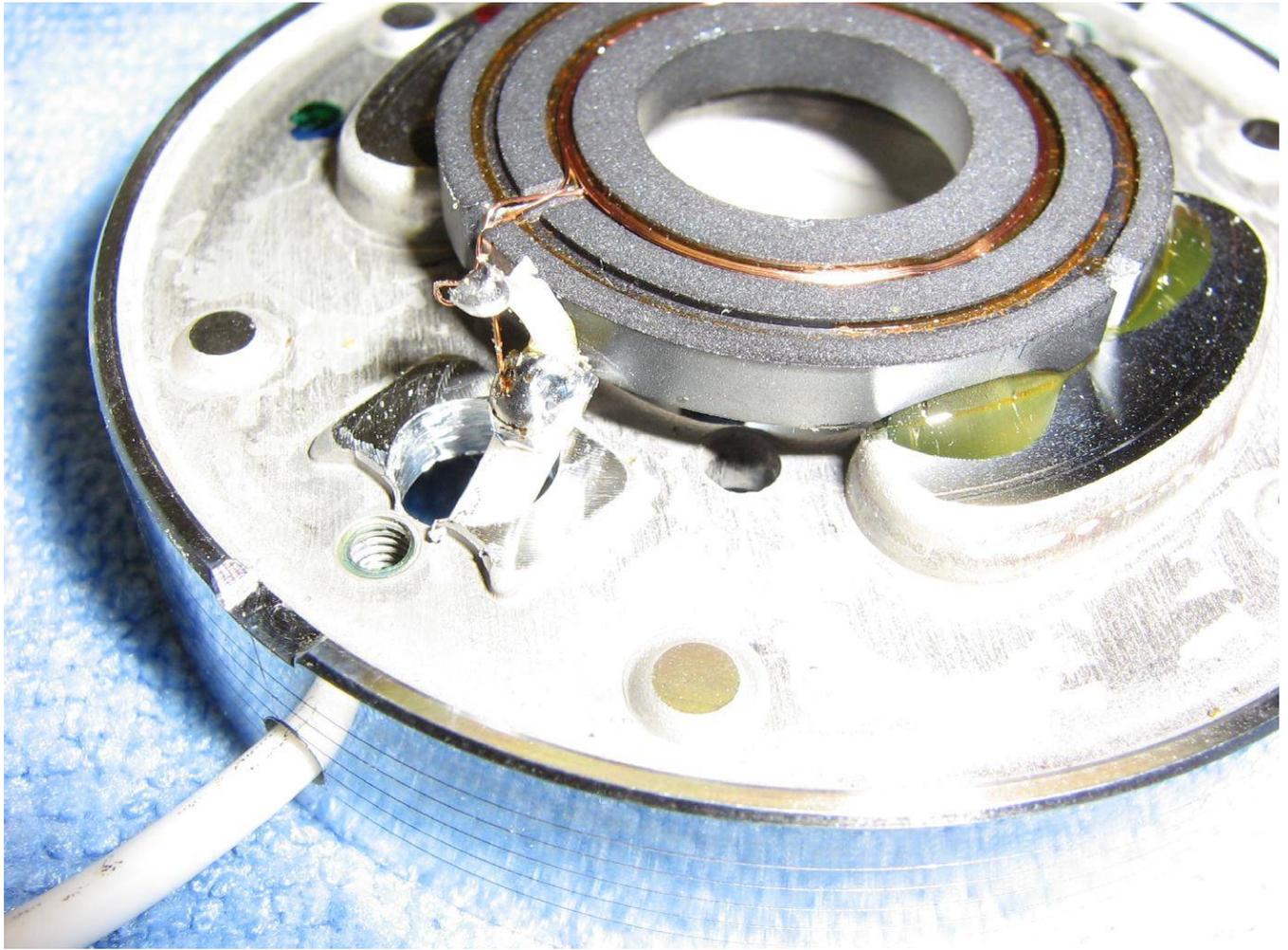
One of the coils should be removed completely from the head to avoid any unnecessary circuit loading.



Closeup of the actual playback/recording head's two solder connections.



Unsolder the two wire connections and remove the brass plate.



Solder some coaxial cable to the wire loops on each of the drum sections.

No impedance matching will be used for this experiment, but you should match each of the loops to 50 ohms to lower the insertion loss.

Also, note that one of the coil windings (outer) on the ferrite core has been removed.



Completed rotating coaxial connection using an old VCR drum head.

The insertion loss was quite high, and it only works up to around 10 MHz.

Test the connections by attaching the rotating connector to a HF receiver and monitoring the NIST time standard signals at 2.5, 5, 10, 15, and 20 MHz.

Proper 50 ohm impedance matching to the loops would probably increase the overall performance significantly. The operating frequency range will probably still stay low though.

A video of the adapter's rotation in action is available here:

http://zine.gbppr.org/Rotary_Coaxial_Cable_Connection.wmv

Sharp Four-Head VCR Schematic

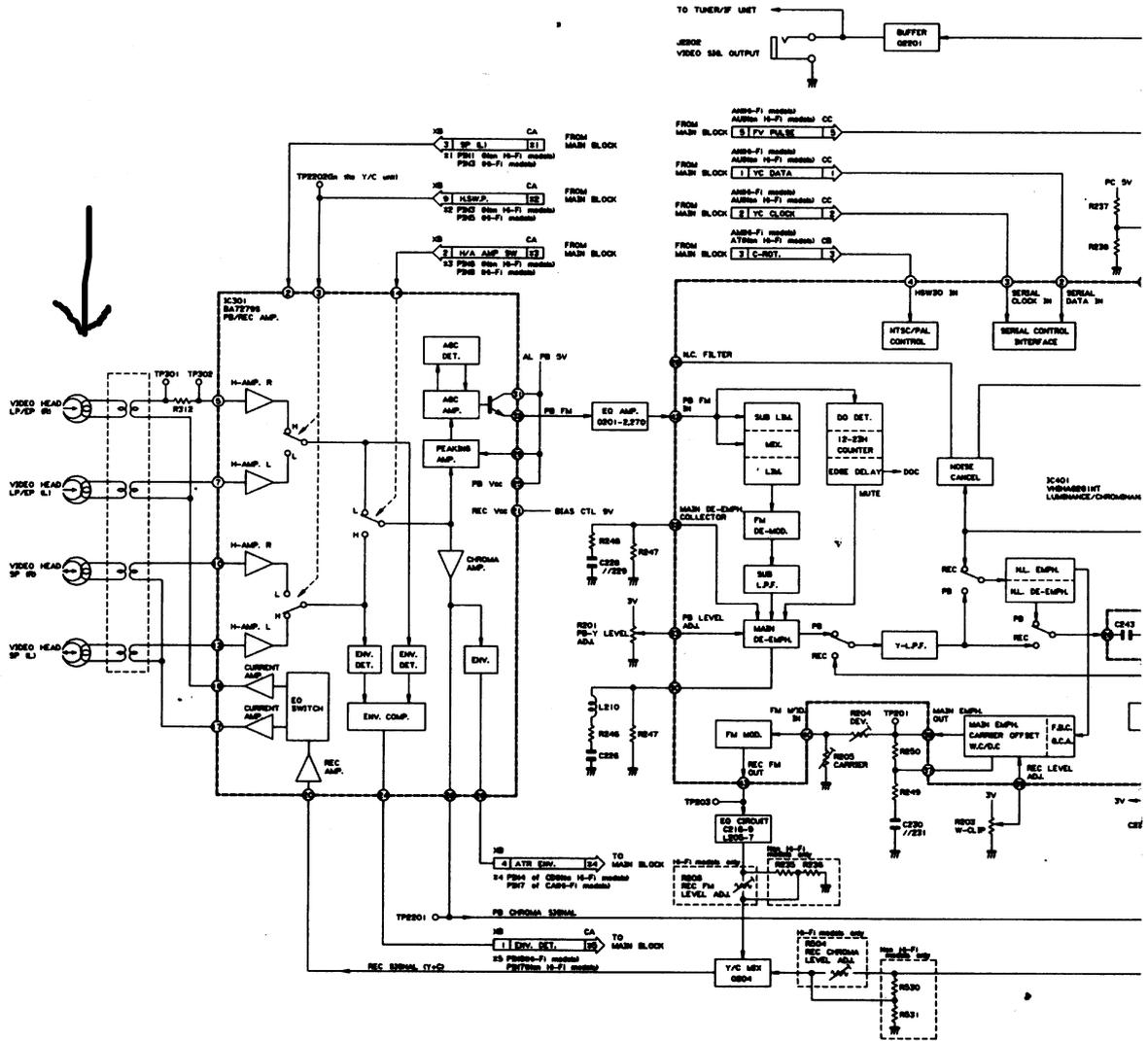


PROFAX Manufacturers' Schematics
 JANUARY 1995

SIGNAL FLOW BLOCK DIAGRAM

Product safety should be considered when component replacement is made in any area of an electronics product. A star next to a component symbol number designates components in which safety is of special significance. It is recommended that only exact cataloged parts be used for replacement of these components.

Use of substitute replacement parts that do not have the same safety characteristics as recommended in factory service information may create shock, fire, excessive x-radiation or other hazards.



Reprinted by permission of Sharp Copyright 1995,
 CQ Communications, 76 N. Broadway, Hicksville, NY 11801

DMS-100 Directory Number Route (DNROUTE)

Table Name

Directory Number Route

Functional Description of Table DNROUTE

Table DNROUTE lists information for programmable Directory Numbers (DN) in the switch (such as a DN that identifies a route), rather than Line Equipment Numbers (LEN). Table DNROUTE replaces table WRDN (Write Directory Number).

Table DNROUTE associates a DN with a specific trunk group member. This DN specifies the lowest numbered DS-0 on the Customer Premises Equipment (CPE) on which a wideband call must terminate. Given the lowest numbered DS-0 and the bandwidth, the incoming wideband call is connected to the CPE.

There are many variations of input for table DNROUTE, as shown in the table below. Selectors not described in the table that follows are not valid entries for table DNROUTE.

DNROUTE Features

DN Selector	Usage
D	Treatment
DSVC	Default Service
M	Listed Directory Number
MM	Meet-Me Conference
RSMT	Restricted Dial Tone
SYN	Synonym Directory Number
T	Route List
FEAT	Automatic Call Distribution (ACD)
FEAT	Advanced Intelligent Network (AIN)
FEAT	Automatic Set Relocation (ASR)
FEAT	Call Forward/Interface Busy
FEAT	Direct Inward System Access (DISA)
FEAT	Message Center DN (MCDN)
FEAT	Message Storage and Retrieval (MSR)
FEAT	Meet-Me Conference (MEETME)
FEAT	Meridian Offnet Access (MONA)
FEAT	Mobile Directory Number Trigger (MTXDNTRIG)
FEAT	Simultaneous Ringing (SIMRING) Virtual DN
FEAT	Subscriber Programmable Ringing for CFDA (SPRING) - Option RCTL (Ring Control)
FEAT	Suppressed Ringing Access (SRA)
FEAT	Uniform Call Distribution (UCD)

Note 1: The selectors C, H, L, P, HC, LC, A, ILC, IHC, MDN, IMC, SDN, SC, ACDTK, and SCM are listed in field DN_SEL. These selectors are not valid entries for table DNROUTE.

Note 2: Selector MTXDNTRIG only applies to the DMS-100 Wireless switch.

Tuples for feature Meet-Me Conference are no longer applicable to table DNROUTE. The Meet-Me Conference data is datafilled in table MMCONF (Meet-Me Conference), and the Meet-Me DN information is stored in table DNINV (Directory Number Inventory).

Datafill Sequence & Implications

The following tables must be datafilled before table DNROUTE:

- TOFCNAME (Terminating Office Name)
- ACDSGRP (Automatic Call Distribution Subgroup)
- AVRTDATA (AUTOVON Route Data)
- BCDEF (Bearer Capability Definition)
- BROADCAST (Broadcast Call)
- CLLI (Common Language Location Identifier)
- COSMAP (Network Class of Service Mapping)
- CUSTHEAD (Customer Group Head)
- HNPACONT (List of Home NPA Code Subtables)
- IBNRTE (Integrated Business Network Route)
- XLAPLAN (Translation Plan)
- RATEAREA (Rate Area)
- MSRTAB (Message Storage and Retrieval)
- NARDATA (Network Access Registers Data)
- NCOS (Network Class of Service)
- OFRT (Office Route)
- PRECONF (Preset Conference)
- TMTCNTL.TREAT (Treatments Subtable)
- TRIGDIG (Trigger Digits)
- TRIGINFO (Trigger Information)
- TRIGGRP (Trigger Group)
- KSETFEAT (Business Set and Data Unit Feature)
- UCDGRP (Uniform Call Distribution Group)

During the One Night Process (ONP), table TABXFR (Table Transfer) detects DNTRIGGER datafill. The detection does not allow transfer of DNTRIGGER datafill to the inactive side.

The following restrictions apply to the Default Service (DSVC) selector, which identifies a DN in table DNROUTE as a Default Service DN:

- Before you can provision a Default Service DN in table DNROUTE, you must execute the Command Interpreter (CI) `DEFSVCCI` tool `SETUP` command. This command provisions Default Service data on all ISDN interfaces that support Default Service.
- The `DEFSVCCI` tool `REMOVE` command removes all provisioning established by the `SETUP` command. Before using this command, you must remove the Default Service DN from table DNROUTE.

Table Size

0 to 640,000 tuples

If office parameter `ACTIVE_DN_SYSTEM` in table OFCENG is set to North American, the maximum size is 640,000 tuples.

If office parameter `ACTIVE_DN_SYSTEM` in table OFCENG is set to Universal, the maximum size is 500,000 tuples.

Note: Tables DNROUTE and DNINV use the same physical store. The maximum size of table DNROUTE is less than or equal to table DNINV.

The maximum number of Advanced Intelligent Network (AIN) tuples that can be datafilled is 2,048.

Datafill

The following table describes datafill for table DNROUTE:

Table DNROUTE Field Descriptions

Field	Subfield	Entry	Explanation and Action
AREACODE		0 to 9999999 (1 to 7 digits)	<i>Area Code</i> The area code identifies a major geographical area served by the switch. If office parameter ACTIVE_DN_SYSTEM in table OFCENG is set to "North American," the area code must be three digits long.
OFCCODE		0 to 9999999 (0 to 7 digits)	<i>Office Code Digit Register</i> The office code is a subregion of the area code. If office parameter ACTIVE_DN_SYSTEM in table OFCENG is set to "North American," the area code must be three digits long. The office code must be specified in table TOFCNAME. Tuples can be added if the value of field OFCCODE is "\$" as specified in table TOFCNAME. They cannot contain STNCODEs whose leading digits are an OFCCODE in the same area code.
STNCODE		0 to 99999999 (up to 8 digits)	<i>Station Code</i> The station code identifies a unique station within the terminating office (TOFC). If office parameter ACTIVE_DN_SYSTEM in table OFCENG is set to "North American," the STNCODE code must be one or four digits in length. If one digit is entered, it is treated as a D-digit, where the D-digit represents the fourth digit in the format ABC-DEFG. A D-digit is then given the appropriate DN result. For example, if area code 613 and office code 226 are entered, and if "5" is entered as the station code, any calls to a number beginning with 6132265 are automatically routed to the specified treatment. If "5" is the D-digit, and DNROUTE is datafilled as "613 226 5 D OPRT", any numbers for 6132265 are routed to the operator treatment as set in table TMTCNTL. Ambiguity within the STNCODEs of one TOFC is not permitted. For example, 8594 and 859 cannot be specified as STNCODEs within one TOFC.
XLAPLAN		Alphanumeric (up to 16 characters)	<i>Translation Plan Index</i> Enter the index into the XLAPLAN table.
RATEAREA		Alphanumeric (up to 16 characters)	<i>Area Code</i> Enter the index into the RATEAREA table.

-End-

Error Messages

The following table explains error messages that can occur when you attempt to datafill table DNROUTE.

Note: You cannot assign a Public Office Directory Number (PODN) when Software Optionality Control option LNP00200 is IDLE.

Message	Explanation	User Action
ERROR: PODN may not be assigned when Software Optionality Control option LNP00200 is IDLE.	SOC option LNP00200 is in the IDLE state.	Activate SOC option LNP00200. Assign the PODN again.

Note: When an "Unable To Write New Data" error message is displayed at the MAP terminal, an add or change operation to table DNROUTE was unsuccessful. When the add operation fails, no tuple is written to table DNROUTE. When a change operation fails, the existing tuple in table DNROUTE remains unchanged.

Horn Antenna for 2.45 GHz Microwave Oven Magnetrons

Overview

When constructing EMP, HERF, or radar devices based around microwave oven magnetrons, it is often useful to have a horn antenna on the antenna's feed to provide the proper free-space impedance matching and parabolic dish illumination. Not having a good antenna impedance match or dish illumination will often result in lower overall antenna gain and wierd sidelobes in the antenna's radiation pattern. Now, the chances of finding a commercial feed horn at a hamfest are going to be very low, so your best bet is to try and make one yourself.

A fairly high-quality, medium-gain, pyramidal horn antenna operating at 2.45 GHz isn't as hard to construct as you may think. All the materials are available at most hardware or hobby stores. The horn antenna's layout dimensions, equations, and construction techniques are available in Chapter 18 of the *ARRL Antenna Handbook*. Since the ARRL can go fuck themselves, a copy of the handbook's section entitled "A Horn Antenna for 10 GHz" will be included here. The design is easily scaled to other frequencies.

Read the article first to get a good idea on what you'll need to do for laying out and marking the four main pieces of the horn. The article's instructions are kinda confusing, but it will start to make sense if you layout everything onto a large piece of cardboard. Two of the horn's sides will need to be "a little longer" to allow soldering on the *outside* seams of the horn. This is to help improve the horn's electrical specifications when operating at microwave frequencies as solder is actually a poor conductor.

This horn antenna will be made from a 4" x 10" piece of 0.008" thick tin sheet stock (K&S Engineering #254). While this particular tin sheet is a little too thin for proper mechanical stability, the ease of soldering will more than make up for it. You will also need to salvage the little "magnetron-to-waveguide" assembly often found in most older microwave ovens. This will be used to mount the magnetron and to also allow for some custom L-brackets which the horn will attach to.

When mounting the horn on a parabolic dish, the horn should be adjusted so that the focal point is just a little bit *inside* the horn's main opening. To reduce the possibility of sidelobes, the focal length should also be a whole number of 1/2 wavelengths. This keeps the proper phase relationship with the "direct" and "reflected" signal coming from the feed. Horn antennas are naturally wideband, and should work over the entire frequency range of the waveguide feed. It should also be able to handle all the power you can throw at it. A piece of polystrene from the hobby store can be glued to the horn's opening to make it somewhat water resistant.

2.45 GHz Horn Dimensions

One Wavelength at 2.45 GHz : 4.82 inches
Target Horn Gain : 14 dB
Gain as a Power Ratio : 25

"L" Length : 7.88 inches
"A" Length : 5.34 inches
"B" Length : 4.33 inches
"S" Length : 2.88 inches

Waveguide's H-Plane : 3.50 inches
Waveguide's E-Plane : 1.70 inches

Construction Notes & Pictures



Magnetron waveguide assembly and aluminum L-brackets.

The waveguide assembly was salvaged from an old microwave oven. Use a chisel and a hammer to break the spot welds holding the waveguide to the microwave oven's chassis.

The waveguide's internal dimensions are 3.5 inches wide (H-plane) and 1.70 inches high (E-plane).

The L-brackets are made from 1/2" aluminum angle stock. The brackets will be used for mounting the horn antenna and can also "square up" the opening of the waveguide in case it was not cut straight.



Temporarily attach the four L-brackets to the waveguide as shown.

This part is quite critical, as the waveguide opening will need to be square, without any gaps to prevent dangerous RF leakage. Drill a series of evenly-spaced holes for the mounting screws. Be sure the mounting screws to interfere with each other.



Alignment of the L-brackets for the final horn mount.

Note how the top bracket was used to square the waveguide opening as it was cut a little crooked.

Use a belt sander to further square up and debur the L-brackets. Mark each bracket and waveguide side with an engraver when disassembling. This will help keep everything in alignment when construction is finished.



Layout and cut the tin stock.

Layout the dimensions per the *ARRL Handbook* article. You should layout and mark everything ahead of time on a piece of cardboard to practice with. Move onto cutting the actual tin when all the dimensions appear O.K.

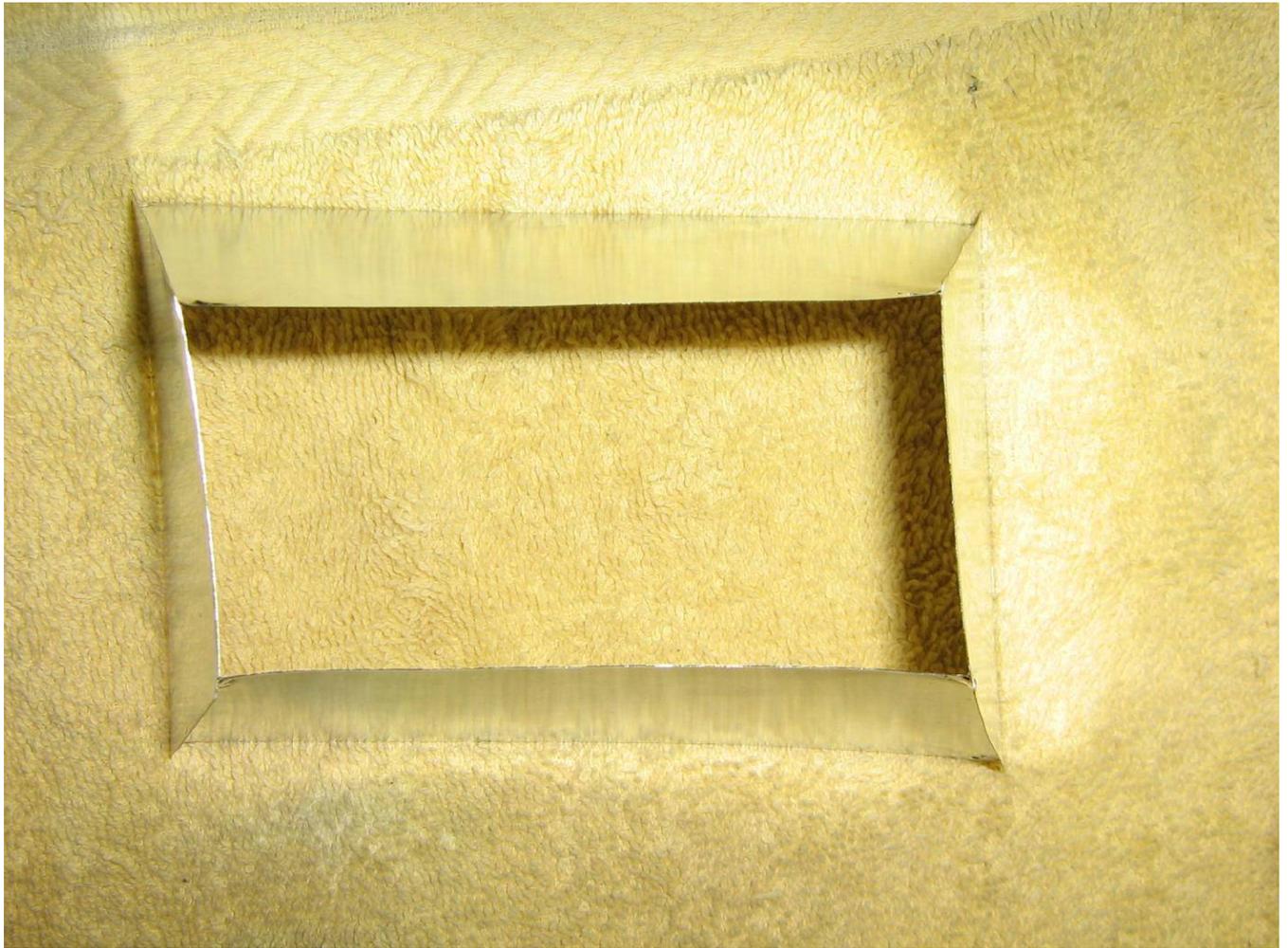
A digital calipers and carbide scribe will be really useful for doing the layout marking. A pair of heavy-duty scissors can be used to cut the tin as it's very thin.



Align the four main pieces that make up the horn itself, being sure the side that attaches to the waveguide is mechanically solid and square.

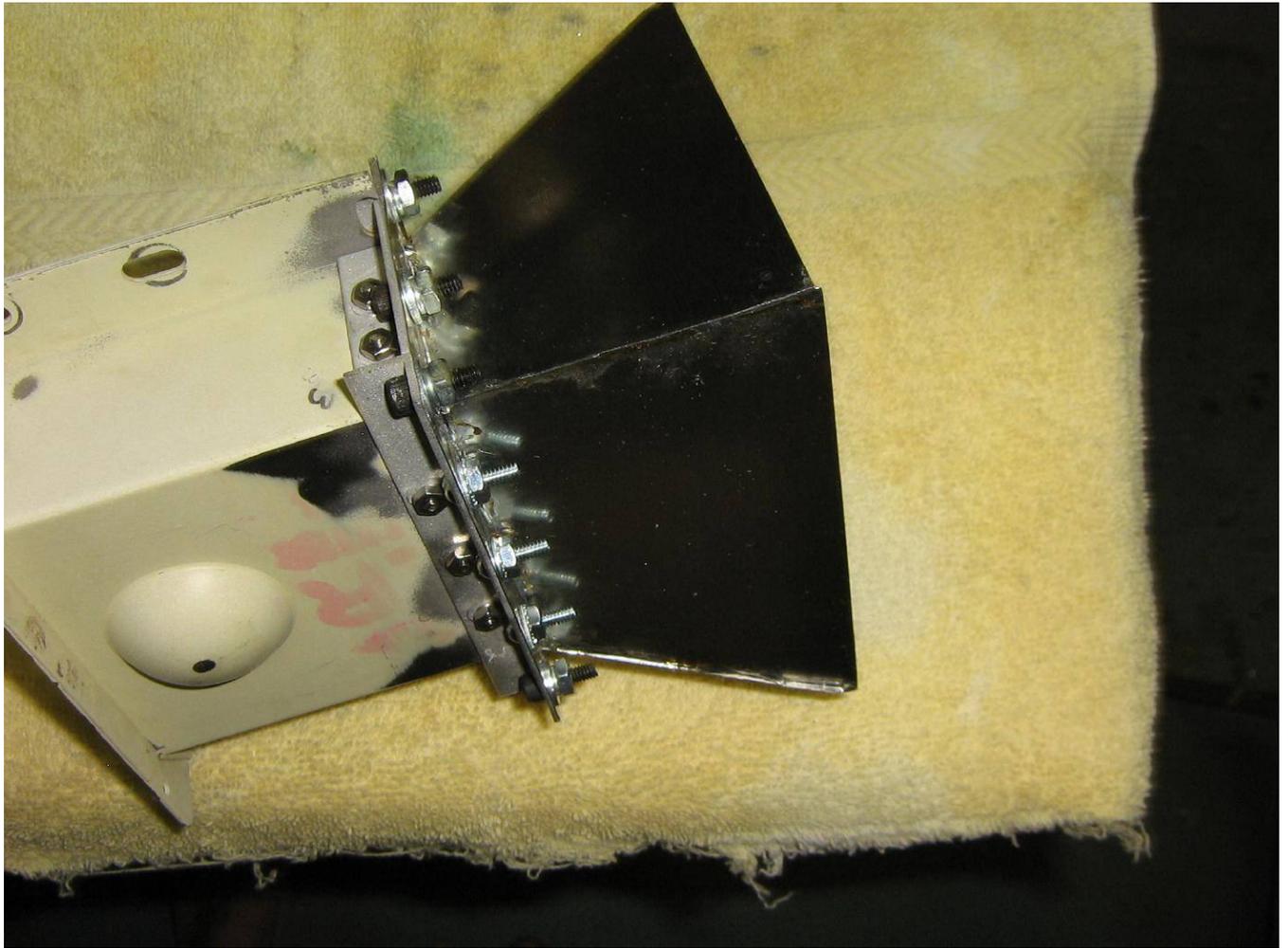
Tack solder the corners to hold everything in place.

Note how crooked the horn's opening is. This can be trimmed down by gently placing the horn on a belt sander.



Cut a piece of tin for mounting the horn to the L-brackets.

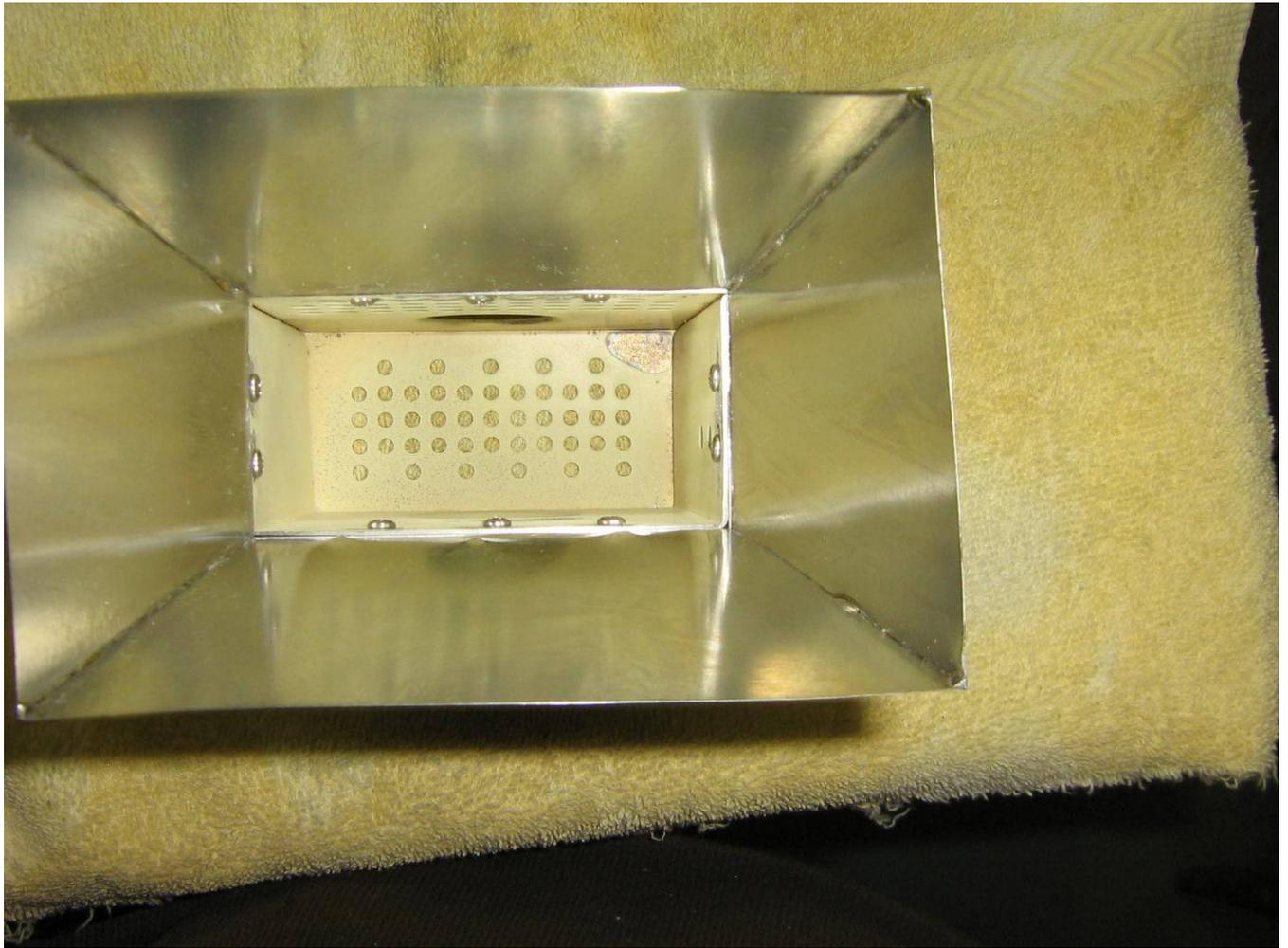
Use a nibbling tool to cut out the center opening so it matches the internal dimensions of the waveguide.



Using a combination of attaching the mounting screws and soldering, solder all the **outside** seams around the horn antenna.

The mounting hardware will be used to "square up" and secure the thin pieces of tin while you solder them in place.

Again, don't worry too much about the large opening of the horn, just concentrate on making a good, RF-proof seal between the waveguide and the horn's mounting plate. There should be **no** gaps in the seam and no protrusions sticking up into the waveguide's opening.



Horn and waveguide internal view.

The L-brackets were secured to the waveguide assembly using #4 stainless steel hardware.

Use a small file to trim the inside of the horn and the lip of the horn-to-waveguide transition.

The right-panel came out a little lopsided...



Finished view.

Painted black as the tin is very shiny and can rust.

The finished tin horn isn't very mechanically rigid and will easily bend or dent, so you'll need to be careful handling it. Brass or copper sheets can be used for the horn's construction instead. This will increase in horn's strength and maybe even the electrical performance, but with an increase in cost.

When used with a 800 watt, 2.4 GHz magnetron, the effective radiated output power with just this horn will be around 20 kilowatts.

A Horn Antenna for 10 GHz

The horn antenna is the easiest antenna for the beginner on 10 GHz to construct. It can be made out of readily available flat sheet brass. Because it is inherently a broadband structure, minor constructional errors can be tolerated. The one drawback is that horn antennas become physically cumbersome at gains over about 25 dB, but for most line-of-sight work this much gain is rarely necessary. This antenna was designed by Bob Atkins, KA1GT, and appeared in *QST* for April and May 1987.

Horn antennas are usually fed by waveguide. When operating in its normal frequency range, waveguide propagation is in the TE_{10} mode. This means that the electric (E) field is across the short dimension of the guide and the magnetic (H) field is across the wide dimension. This is the reason for the E-plane and H-plane terminology shown in Fig 65.

There are many varieties of horn antennas. If the waveguide is flared out only in the H-plane, the horn is called an H-plane sectoral horn. Similarly, if the flare is only in the E-plane, an E-plane sectoral horn results. If the flare is in both planes, the antenna is called a pyramidal horn.

For a horn of any given aperture, directivity (gain along the axis) is maximum when the field distribution across the aperture is uniform in magnitude and phase. When the fields are not uniform, side lobes that reduce the directivity of the antenna are formed. To obtain a uniform distribution, the horn should be as long as possible with minimum flare angle. From a practical point of view, however, the horn should be as short as possible, so there is an obvious conflict between performance and convenience.

Fig 66 illustrates this problem. For a given flare angle and a given side length, there is a path-length difference from the apex of the horn to the center of the aperture (L), and from the apex of the horn to the edge of the aperture (L'). This causes a phase difference in the field across the aperture, which in turn causes formation of side lobes, degrading directivity (gain along the axis) of the antenna. If L is large this difference is small, and the field is almost uniform. As L decreases however, the phase difference increases and directivity suffers. An optimum (shortest possible) horn is constructed so that this phase difference is the maximum allowable before side lobes become excessive and axial gain markedly decreases.

The magnitude of this permissible phase difference is different for E-plane and H-plane horns. For the E-plane horn, the field intensity is quite constant across the aperture. For the H-plane horn, the field tapers to zero at the edge. Consequently, the phase difference at the edge of the aperture in the E-plane horn is more critical and should be held to less than 90° ($\frac{1}{4} \lambda$). In an H-plane horn, the allowable phase difference is 144° (0.4λ). If the aperture of a pyramidal horn exceeds one wavelength in both planes, the E-plane and H-plane patterns are essentially independent and can be analyzed separately.

The usual direction for orienting the waveguide feed is

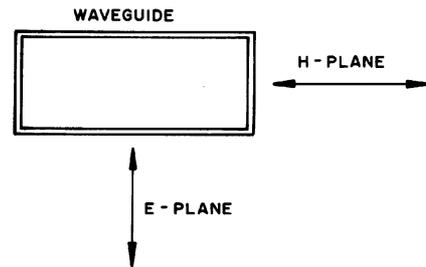


Fig 65—10-GHz antennas are usually fed with waveguide. See text for a discussion of waveguide propagation characteristics.

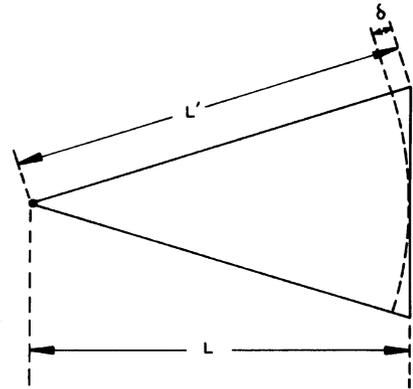


Fig 66—The path-length (phase) difference between the center and edge of a horn antenna is δ .

with the broad face horizontal, giving vertical polarization. If this is the case, the H-plane sectoral horn has a narrow horizontal beamwidth and a very wide vertical beamwidth. This is not a very useful beam pattern for most amateur applications. The E-plane sectoral horn has a narrow vertical beamwidth and a wide horizontal beamwidth. Such a radiation pattern could be useful in a beacon system where wide coverage is desired.

The most useful form of the horn for general applications is the optimum pyramidal horn. In this configuration the two beamwidths are almost the same. The E-plane (vertical)

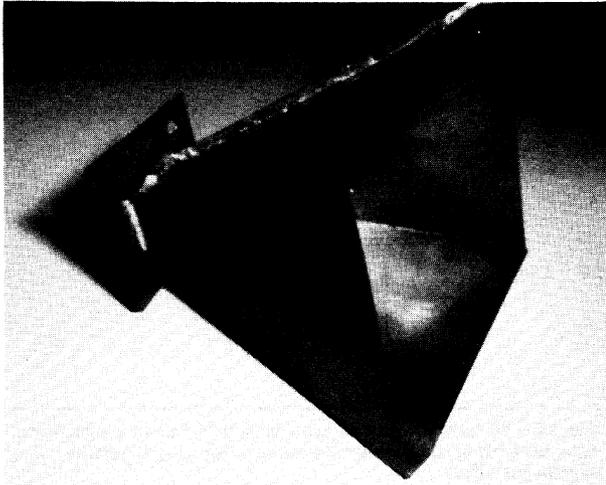


Fig 67—This pyramidal horn has 18.5 dBi gain at 10 GHz. Construction details are given in the text.

beamwidth is slightly less than the H-plane (horizontal), and also has greater side lobe intensity.

Building the Antenna

A 10-GHz pyramidal horn with 18.5 dBi gain is shown in Fig 67. The first design parameter is usually the required gain, or the maximum antenna size. These are of course related, and the relationships can be approximated by the following:

$$L = \text{H-plane length } (\lambda) = 0.0654 \times \text{gain} \quad (\text{Eq 1})$$

$$A = \text{H-plane aperture } (\lambda) = 0.0443 \times \text{gain} \quad (\text{Eq 2})$$

$$B = \text{E-plane aperture } (\lambda) = 0.81 A \quad (\text{Eq 3})$$

where

gain is expressed as a *ratio*; 20 dB gain = 100

L, A and B are dimensions shown in Fig 68

From these equations, the dimensions for a 20 dB gain horn for 10.368 GHz can be determined. One wavelength at 10.368 GHz is 1.138 inches. The length (L) of such a horn is $0.0654 \times 100 = 6.54 \lambda$. At 10.368 GHz, this is 7.44 inches. The corresponding H-plane aperture (A) is 4.43λ (5.04 inches), and the E-plane aperture (B), 4.08 inches.

The easiest way to make such a horn is to cut pieces from brass sheet stock and solder them together. Fig 68 shows the dimensions of the triangular pieces for the sides and a square piece for the waveguide flange. (A standard commercial waveguide flange could also be used.) Because the E-plane and H-plane apertures are different, the horn opening is not square. Sheet thickness is unimportant; 0.02 to 0.03 inch works well. Brass sheet is often available from hardware or hobby shops.

Note that the triangular pieces are trimmed at the apex

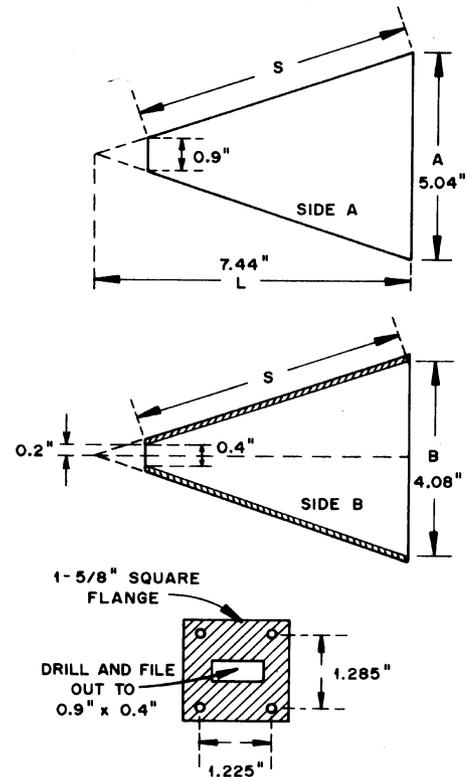


Fig 68—Dimensions of the brass pieces used to make the 10-GHz horn antenna. Construction requires two of each of the triangular pieces (side A and side B).

to fit the waveguide aperture (0.9 × 0.4 inch). This necessitates that the length, from base to apex, of the smaller triangle (side B) is shorter than that of the larger (side A). Note that the length, S, of the two different sides of the horn must be the same if the horn is to fit together! For such a simple looking object, getting the parts to fit together properly requires careful fabrication.

The dimensions of the sides can be calculated with simple geometry, but it is easier to draw out templates on a sheet of cardboard first. The templates can be used to build a mock antenna to make sure everything fits together properly before cutting the sheet brass.

First, mark out the larger triangle (side A) on cardboard. Determine at what point its width is 0.9 inch and draw a line parallel to the base as shown in Fig 68. Measure the length of the side S; this is also the length of the sides of the smaller (side B) pieces.

Mark out the shape of the smaller pieces by first drawing

a line of length B and then constructing a second line of length S. One end of line S is an end of line B, and the other is 0.2 inch above a line perpendicular to the center of line B as shown in Fig 68. (This procedure is much more easily followed than described.) These smaller pieces are made slightly oversize (shaded area in Fig 68) so you can construct the horn with solder seams on the outside of the horn during assembly.

Cut out two cardboard pieces for side A and two for side B and tape them together in the shape of the horn. The aperture at the waveguide end should measure 0.9×0.4 inch and the aperture at the other end should measure 5.04×4.08 inches.

If these dimensions are correct, use the cardboard templates to mark out pieces of brass sheet. The brass sheet should be cut with a bench shear if one is available, because scissors type shears tend to bend the metal. Jig the pieces together and solder them on the *outside* of the seams. It is important to keep both solder and rosin from contaminating the inside of the horn; they can absorb RF and reduce gain at these frequencies.

Assembly is shown in Fig 69. When the horn is completed, it can be soldered to a standard waveguide flange, or one cut out of sheet metal as shown in Fig 68. The transition between the flange and the horn must be smooth. This antenna provides an excellent performance to cost ratio (about 20 dB gain for about five dollars in parts).

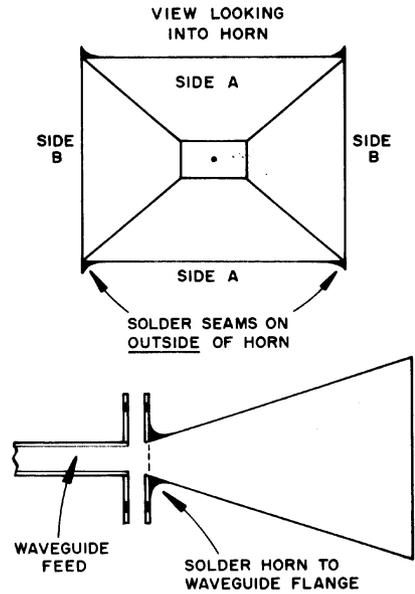


Fig 69—Assembly of the 10-GHz horn antenna.

Bonus

(specifically excluding all of the more violent suggestions) regarding what to do with racists:

- o Teach them to read.
- o Make them travel. Mark Twain said, "Travel cures prejudice." Maybe getting them out of those little white towns and trailer parks would do them some good.
- o Deport them to the Caucasus mountains.
- o Force them to take a bath every day until they renounce racism.
- o Put them on an island. Then they could be happy hating each other forever.
- o Require them to attend college classes at state universities -- their failures will instill humility since non-whites will always outperform them.
- o Euthenize all mentally deficient racists (per their own Nazi doctrine) -- this will eliminate over half of them.
- o Lock them in a football stadium with lots of machine guns and alcohol.
- o Send the Aryans back to India.
- o Put them in the boxing ring and let them prove their superiority over blacks.
- o Sterilize all racists them so they don't breed back into the population and pollute the human gene pool.
- o Fine their mothers for littering society with white trash. If their fathers can be identified and caught, jail them.
- o Make them take an IQ test and euthenize those who flunk.
- o Make them wear yellow swastikas on their clothes, confiscate their property, and confine them to walled-in ghettos or send them to work camps.
- o Make them slaves for 400 years so that they understand the evil that they have done.

Send additional suggestions to Students and Youth Against Racism. All images on this page are public domain and may be used without permission.

End
Racism

RETURN

End
Racism

From: www.clubs.psu.edu/up/sayar/solution.htm

The above is a Penn State Students & Youth Against Racism webpage entitled "What Should Be Done With Racists?"

Note all the spelling errors. They even spell "done" wrong in the HTML title header:

```
-----  
<html>  
<head>  
<title>The Final Solution: What Should Be Doine With Racists?</title>  
</head>  
<body link="#0066FF" vlink="#805580" BGCOLOR="#ffffff">  
-----
```

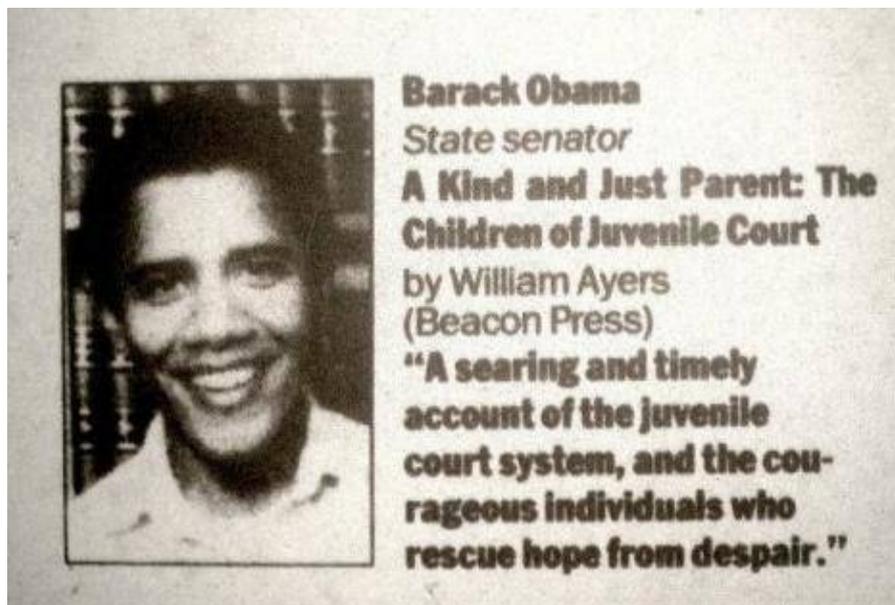
Change!

End of Issue #55



Any Questions?

Editorial and Rants



Still think Obongo doesn't know who Bill Ayers is?

From the *Chicago Tribune*, December 21, 1997. (www.zombietime.com/zomblog/?p=64)



CONFIDENTIAL

Do not distribute

-----for internal use only-----

Strategy Group - Critical Contacts:

Clear Peak Colorado New West Fellowship CD4-Coloradans for Life CoDA ⇔ DCCC ⇔ Fawcett ⇔ Perlmutter ⇔ Paccione ⇔ Gordon ⇔ Kennedy ⇔ O'Brien ⇔ Ritter ⇔ CoDA ⇔ DCCC	Dominic Del Papa – Principal Jessica Baker – Emergency Response	303.888.8556 ddp@ikoninc.net 303.861.0223 ext.2 Jessica@ikoninc.net
Colorado Voter Project CVP – Gubernatorial Field Ops	Tyler Chaffee Leticia Martinez	303.832.2444 tyler@rbistrategies.com 303.7177593 leticiamartinez@gmail.com
Senate 527 – Moving Colorado Forward	Ray Rivera Mary Alice Mandarich	720.984.5439 rrivera@afscmecolorado.org 303.594.0504 mamandaric@aol.com
House 527 – Mainstreet Colorado Minimum Wage – wedge issue management - increasing CO union power: “Educate the Idiots” campaign. Target: minorities, GED’s, drop-outs..... Legal maneuvering and navigation, distraction lawsuits against Gigi, Trailhead, etc.	Kelly Nordini Carolyn Siegel – AFL-CIO	303.437.5277 knordini@gmail.com 303.710.2856 csiegel@coaficio.org
CoDA Infrastructure Committee, Linda Shoemaker, (current) Chair	Macghan Collins – LS Chief of Staff, Head of Family Foundation	303.442.1200 mcollins@sprucestreetmansion.com
America Votes	Rich Pelletier	rich@americavotes.org
Colorado Democracy Alliance – CoDA	AlYates – Interim Executive Director – Advises Pat Stryker	970.481.2784 albertyates@comcast.net
Colorado Democracy Alliance – CoDA, \$Exchange, Leave-No-Trace Strategy-ACTIVATED Individuals ⇔ CoDA\$ ⇔ Campaigns ⇔ CoDA\$ ⇔ 527 \$ ⇔ Individuals/Campaigns ⇔ CoDA\$ ⇔ 527\$	Courtney A.E. Messenbaugh Director of Planning and Assessment - CoDA primary contact	courtneyessary@yahoo.com 303.830.6655 – office

CoDA

Next Board Meeting: Linda Shoemaker’s home – Boulder

Board Members Attending: Ruff Bridges, Pat Stryker, Tom Congdon, Doug Phelps, Tim Gill, Linda Shoemaker, Bruce Oreck.

Advisories:

- 1-Secrecy Level: **HIGH-COVERT:** take usual precautions.
- 2-Entry: **Password Protected:** Notification on day of rendezvous.
- 3-Media: **Activated:** Imbedded Media Ops.

Schedule:

- 527’s, c3’s & campaign ops – long and short strategy – End of October – TBD – (**Advisories 1 & 2**).
- Stryker Victory Party & “debrief”, at or near residence, November 8, 2006 (**1&2**).

From: www2.facethestate.com

The above sheet is from a Colorado Democrat "strategy" handout.

Still think Democrats want to help the "little people?" *Hah!* They just use "idiots" for the votes!

This is marked confidential, so don't let anyone know!

Rashid Khalidi
 From Wikipedia, the free encyclopedia
 (Difference between revisions)

Revision as of 22:24, 26 October 2008 (edit)
 71.195.238.59 (Talk)
 (->Obama relationship)
 - Previous edit

Revision as of 23:02, 26 October 2008 (edit) (undo)
 203.14.108.18 (Talk)
 (->Obama relationship)
 Next edit

Line 225: ===Obama relationship===
Obama had no relationship with Khalidi except in the minds of right wing bloggers. They will say anything to win the white house for McCain].

Line 225: ===Obama relationship===
At Khalidi's going-away party in 2003, the scholar lavished praise on Obama, telling the mostly Palestinian American crowd that the state senator deserved their help in winning a U.S. Senate seat. "You will not have a better senator under any circumstances," Khalidi said. (LA TIMES)

=== NYC teacher training program ===

Revision as of 23:02, 26 October 2008

Rashid Khalidi (born 1950), an American historian of the Middle East, is the Edward Said Professor of Arab Studies at Columbia University, and director of the Middle East Institute of Columbia's School of International and Public Affairs.

Rashid Khalidi	
Born	1950
	New York, New York
Residence	New York, New York

From: en.wikipedia.org

Wikipedia is fair and balanced source of accurate information!

I read that, on Wikipedia!



why do black

why do black people stink	441,000 results
why do black men like white women	3,790,000 results
why do black guys like white girls	1,310,000 results
why do black people have white hands	23,900,000 results
why do black people have white palms	828,000 results
why do black men cheat	852,000 results
why do black women have big bums	448,000 results
why do black people say aks	474,000 results
why do black people like watermelon	475,000 results
why do black people love fried chicken	381,000 results

close

Advanced Search
 Preferences
 Language Tools

Advertisi

ogle UK

Blatant ACORN Voter Fraud

The following are fraudulent U.S. voter registrations created by the radical left-wing "community organizer" group ACORN. Barack Hussein Obama used to be a lawyer for ACORN...

Creating false voter registrations allows people with false identities (or illegal aliens) the chance to vote multiple times or to vote in different districts.

You **WILL NOT** be hearing about this from *\$2600 Magazine*, Slashdot, Digg, or the mainstream liberal media!!!!

Note how some of the voter registration forms all suspiciously have the same handwriting!

Also note several of the "voters" signed their name – last name *first!*

When was the last time you did *that*? When you copied names from a phone book, that's when! LOL!

PETITION FOR DIRECT LEGISLATION BY VOTER INITIATIVE **2068**

The undersigned acknowledge that they are voters currently registered to vote within the municipal limits of the City of Albuquerque and sign this petition voluntarily in support of the measure to be presented for approval to the City Council of the City of Albuquerque which set forth in full on the attachment to this page. If the measure below is not approved by the City Council, the measure shall be placed on the ballot at a City election pursuant to City Charter Article III, Section 3.

Pursuant to City Charter Article III, the undersigned, support this proposed measure to enact a Living Wage Ordinance that established a Minimum Wage in the City of Albuquerque of \$7.50 per hour for regular employees and \$4.50 per hour for tipped employees. The ordinance will not cover small businesses with 10 or fewer employees, work-study students, interns working for academic credit, or certain other employees who are exempt from the New Mexico minimum wage law. Each year in the future the minimum wage rates will be increased to keep pace with inflation. The ordinance provides for enforcement, including double damages for unpaid workers and forbids retaliation against employees for exercising their rights.

Signature of Voter	Print Name of Voter as listed on Voter Registration Rolls	Print Address of Voter as listed on Voter Registration Rolls
<i>Edmondson Christopher</i>	Edmondson Christopher	3708 Camino Capistrano
<i>St John Judith A</i>	St John Judith A	3708 Camino Capistrano
<i>St John Valeria J</i>	Valeria J St John	3708 Camino Capistrano
<i>Wakley Christine</i>	Christine Wakley	3800 Camino Capistrano
<i>Eugene R Henley</i>	Eugene R Henley	3826 Camino Capistrano
<i>Tyler Henley R</i>	Henley Tyler R	3826 Camino Capistrano
<i>Pace Elizabeth Ann</i>	Elizabeth A Pace	3805 Camino Capistrano
<i>Fossett Ashlee</i>	Fossett Ashlee L	3813 Camino Capistrano
<i>Hoffman Jeffrey F</i>	Hoffman Jeffrey F	3829 Camino Capistrano
<i>Saenz Jacob C</i>	Jacob Saenz	3708 Camino Don Diego



RECEIVED
ALBUQUERQUE CITY CLERK

The form of this petition has been approved by *Judy N. Chavez*, City Clerk Date 05.19.05

Page 2 of 2 Return to 411 Bellamah NW, Albuquerque, NM 87102, 242-7411

PETITION FOR DIRECT LEGISLATION BY VOTER INITIATIVE

S15 2069

The undersigned acknowledge that they are voters currently registered to vote within the municipal limits of the City of Albuquerque and sign this petition voluntarily in support of the measure to be presented for approval to the City Council of the City of Albuquerque which set forth in full on the attachment to this page. If the measure below is not approved by the City Council, the measure shall be placed on the ballot at a City election pursuant to City Charter Article III, Section 3.

Pursuant to City Charter Article III, the undersigned, support this proposed measure to enact a Living Wage Ordinance that established a Minimum Wage in the City of Albuquerque of \$7.50 per hour for regular employees and \$4.50 per hour for tipped employees. The ordinance will not cover small businesses with 10 or fewer employees, work-study students, interns working for academic credit, or certain other employees who are exempt from the New Mexico minimum wage law. Each year in the future the minimum wage rates will be increased to keep pace with inflation. The ordinance provides for enforcement, including double damages for unpaid workers and forbids retaliation against employees for exercising their rights.

Signature of Voter	Print Name of Voter as listed on Voter Registration Rolls	Print Address of Voter as listed on Voter Registration Rolls
1. Steve Greenway	Steve Greenway	3705 Camino Don Diego
2. Sherri Celarik	Sherri G Celarik	3713 Camino Don Diego
3. Melissa Garcia	Melissa Garcia	1402 Gt Street
4. Alex William Mallion	Alex William Mallion	1610 Gt St NW
5. Lucino Max	MAX Lucero	609 Aspen Ave
6. Maria Chavez	Maria Chavez	621 Aspen Ave
7. Thelma Chavez	Thelma Chavez	621 Aspen Ave
8. Antonio Armito	Antonio Armito	807 Aspen Ave
9. Michael Armito	Michael Armito J	807 Aspen Ave
10. Andrew Chavez	Chavez Andrew	611 Bellamah Ave



The form of this petition has been approved by Judy N. Chavez, City Clerk

05.19.05
Date

PETITION FOR DIRECT LEGISLATION BY VOTER INITIATIVE

1511

The undersigned acknowledge that they are voters currently registered to vote within the municipal limits of the City of Albuquerque and sign this petition voluntarily in support of the measure to be presented for approval to the City Council of the City of Albuquerque which set forth in full on the attachment to this page. If the measure below is not approved by the City Council, the measure shall be placed on the ballot at a City election pursuant to City Charter Article III, Section 3.

Pursuant to City Charter Article III, the undersigned, support this proposed measure to enact a Living Wage Ordinance that established a Minimum Wage in the City of Albuquerque of \$7.50 per hour for regular employees and \$4.50 per hour for tipped employees. The ordinance will not cover small businesses with 10 or fewer employees, work-study students, interns working for academic credit, or certain other employees who are exempt from the New Mexico minimum wage law. Each year in the future the minimum wage rates will be increased to keep pace with inflation. The ordinance provides for enforcement, including double damages for unpaid workers and forbids retaliation against employees for exercising their rights.

Signature of Voter

Print Name of Voter as listed on Voter Registration Rolls

Print Address of Voter as listed on Voter Registration Rolls

1.	<i>James Cardo</i>	James Cardo	228 Genson Dr SW
2.	<i>William R. Rice</i>	William R. Rice	3021 Payne Rr SW
3.	<i>Sen Rodgers</i>	Sen Rodgers	1301 SunSet RD SW
4.	<i>Jean Rodgers</i>	Jean Rodgers	1301 SunSet RD SW
5.	<i>James Taylor</i>	James Taylor	9800 Rough Rider RD
6.	<i>Deborah Taylor</i>	Deborah Taylor	9800 Rough Rider RD
7.	<i>Daniel Stone</i>	Daniel Stone	1716 Shady Side Dr SW
8.	<i>Marianne Linnin</i>	Marianne Linnin	1325 Park Ave SW
9.	<i>Gary Stewart</i>	Gary Stewart	13116 Delta Longa
10.	<i>Evelyn Stewart</i>	Evelyn Stewart	13116 Della Longa



07:4 PM 28 JUN 05

The form of this petition has been approved by *Judy N. Chavez*, City Clerk

05-19-05
Date

PETITION FOR DIRECT LEGISLATION BY VOTER INITIATIVE

1881

The undersigned acknowledge that they are voters currently registered to vote within the municipal limits of the City of Albuquerque and sign this petition voluntarily in support of the measure to be presented for approval to the City Council of the City of Albuquerque which set forth in full on the attachment to this page. If the measure below is not approved by the City Council, the measure shall be placed on the ballot at a City election pursuant to City Charter Article III, Section 3.

Pursuant to City Charter Article III, the undersigned, support this proposed measure to enact a Living Wage Ordinance that established a Minimum Wage in the City of Albuquerque of \$7.50 per hour for regular employees and \$4.50 per hour for tipped employees. The ordinance will not cover small businesses with 10 or fewer employees, work-study students, interns working for academic credit, or certain other employees who are exempt from the New Mexico minimum wage law. Each year in the future the minimum wage rates will be increased to keep pace with inflation. The ordinance provides for enforcement, including double damages for unpaid workers and forbids retaliation against employees for exercising their rights.

Signature of Voter	Print Name of Voter as listed on Voter Registration Rolls	Print Address of Voter as listed on Voter Registration Rolls
1. <i>Dischweinitz, George</i>	Dischweinitz, George L.	1813 Dorothy St. N.E. 87122 - 533
2. <i>Saba, Amanda N.</i>	Saba, Amanda N.	1817 Dorothy St. N.E. 87122 (533)
3. <i>Saba, Antony W.</i>	Saba, Antony W.	1817 Dorothy St. N.E. 87122 (533)
4. <i>Jaramilla, Charlene</i>	Jaramilla, Charlene	1809 Dorothy St. N.E. 87122
5. <i>Jaramilla, Ray</i>	Jaramilla, Ray	1809 Dorothy St. N.E. 87122 (533)
6. <i>Oxentine, Sarah E.</i>	Oxentine, Sarah E.	1821 Dorothy St. N.E. (533)
7. <i>Coker, Mickey L.</i>	Coker, Mickey L.	1825 Dorothy St. N.E. 87122 (533)
8. <i>Coker, Teressa D.</i>	Coker, Teressa D.	1825 Dorothy St. N.E. 87122
9. <i>Bennett, Doe</i>	Bennett, Doe	1829 Dorothy St. N.E. 87122 (533)
10. <i>Weeden, Roth O.</i>	Weeden, Roth O.	1833 Dorothy St. N.E. (533) 87122



RECEIVED
ALBUQUERQUE CITY CLERK

JUN 28 2005 PM 4:31

Judy N. Chavez
The form of this petition has been approved by Judy N. Chavez, City Clerk

05.19.05
Date

3972

2799

PETITION FOR DIRECT LEGISLATION BY VOTER INITIATIVE

The undersigned acknowledge that they are voters currently registered to vote within the municipal limits of the City of Albuquerque and sign this petition voluntarily in support of the measure to be presented for approval to the City Council of the City of Albuquerque which set forth in full on the attachment to this page. If the measure below is not approved by the City Council, the measure shall be placed on the ballot at a City election pursuant to City Charter Article III, Section 3.

Pursuant to City Charter Article III, the undersigned, support this proposed measure to enact a Living Wage Ordinance that established a Minimum Wage in the City of Albuquerque of \$7.50 per hour for regular employees and \$4.50 per hour for tipped employees. The ordinance will not cover small businesses with 10 or fewer employees, work-study students, interns working for academic credit, or certain other employees who are exempt from the New Mexico minimum wage law. Each year in the future the minimum wage rates will be increased to keep pace with inflation. The ordinance provides for enforcement, including double damages for unpaid workers and forbids retaliation against employees for exercising their rights.

Signature of Voter	Print Name of Voter as listed on Voter Registration Rolls	Print Address of Voter as listed on Voter Registration Rolls
1. Jardine Nicole Lynn	Jardine Nicole Lynn	712 Sumbraa Drive SW
2. Mara Carl L.	Mara Carl L.	820 Sumbraa Drive SW
3. Romero Anthony C.	Romero Anthony C.	704 Sumbraa Drive SW
4. Romero Danda ^{Renee}	Romero Danda Renee	704 Sumbraa Drive SW
5. Jardine Brent W.	Jardine Brent W.	712 Sumbraa Drive SW
6. Vela, Vanessa A.	Vela Vanessa A.	823 Sunbird Court SW
7. Mara Beverly A.	Mara Beverly A.	835 Sunbird Court SW
8. Mendez Joe S.	Mendez Joe S.	831 Sunbird Court SW
9. Louis, Odell	Louis, Odell	800 Sunbird Court SW
10. Porter, Hilda F.	Porter, Hilda F.	900 Sunbird Road SW

FAIL!



The form of this petition has been approved by Judy N. Chavez, City Clerk

Date 05-19-05

was critical in the U.S. Senate race. On March 16th, State Senator Barack Obama won the right to represent the Democratic

National Voting Rights Act, also known as motor voter. Allied only with the state of Mississippi, Illinois had been refusing

Photo courtesy Association of Community Organizations for Reform Now



ACORN members meet with Illinois Senate candidate Barack Obama.

Orga.

Not Found | Change.gov

And it starts...

Censored from Barack Hussein Obama's official "change" website were the details on Hitler's, *err...* "Obama's Youth."

Also, the Wikipedia entry for Benjamin M. Emanuel, Rahm Emanuel's father, has been deleted. Wouldn't want any shady connections or quotes getting out now, would we? LOL!

Change!



Archive from the [Google Cache](#).

A HTML version is available here: www.bizzyblog.com/ChangeDotGovService110808.html

A screenshot of a Mozilla Firefox browser window displaying the Change.gov website. The browser's address bar shows a URL from a Google cache. The website header includes the Change.gov logo and a quote from President-Elect Barack Obama: "Today we begin in earnest the work of making sure that the world we leave our children is just a little bit better than the one we inhabit today." Below the header is a navigation menu with links for HOME, NEWSROOM, BLOG, LEARN, AGENDA, AMERICAN MOMENT, AMERICA SERVES, JOBS, and ABOUT. The main content area is divided into two columns. The left column is titled "AMERICA SERVES" and contains a sub-section "America Serves" with a quote: "When you choose to serve -- whether it's your nation, your community or simply your neighborhood -- you are connected to that fundamental American ideal that we want life, liberty and the pursuit of happiness not just for ourselves, but for all Americans. That's why it's called the American dream." This is followed by a paragraph about the Obama Administration's call for service. The right column features "UPCOMING EVENTS" with a box for "THE INAUGURATION" on January 20, 2009, in Washington, DC. Below this is "THE AGENDA" and "YOUR ADMINISTRATION", which includes portraits and names for President-Elect Barack Obama and Vice President-Elect Joe Biden. At the bottom of the right column is a "TRANSITION DIRECTORY" link. The browser's status bar at the bottom shows "Done".

Benjamin M. Emanuel - Wikipedia, the free encyclopedia - Mozilla Firefox

File Edit View History Bookmarks Tools Help

W http://en.wikipedia.org/wiki/Benjamin_M._Emanuel

Log in / create account

article discussion

Wikipedia: Making Life Easier. [Collapse]

Our Goal: \$6 million

Donate Now »

Benjamin M. Emanuel

From Wikipedia, the free encyclopedia

This page has been deleted. The deletion log for the page is provided below for reference.

- 01:12, 11 November 2008 Jayjg (Talk | contribs) deleted "**Benjamin M. Emanuel**" (*G4: Recreation of a page which was deleted per a deletion discussion: Wikipedia:Articles for deletion/Benjamin M. Emanuel (2nd nomination)*)
- 15:52, 19 January 2007 Avraham (Talk | contribs) deleted "**Benjamin M. Emanuel**" (*per Wikipedia:Articles for deletion/Benjamin M. Emanuel*)

Wikipedia does not have an article with this exact name. Please *search for Benjamin M. Emanuel* in Wikipedia to check for alternative titles or spellings.

- Start the *Benjamin M. Emanuel* article or add a request for it.
- Search for "*Benjamin M. Emanuel*" in existing articles.
- Look for pages within Wikipedia that link to

Look for **Benjamin M. Emanuel** on one of Wikipedia's sister projects:

- Wiktionary (free dictionary)
- Wikibooks (free textbooks)
- Wikiquote (quotations)
- Wikisource (free library)
- Wikiversity (free learning resources)
- Commons (images and media)

Done



Change!

Benjamin M. Emanuel

From Wikipedia, the free encyclopedia

Jump to: [navigation](#), [search](#)

This article is being considered for deletion in accordance with Wikipedia's [deletion policy](#). Please share your thoughts on the matter at [this article's entry](#) on the [Articles for deletion](#) page. Feel free to edit the article, but the article **must not** be blanked, and this notice **must not** be removed, until the discussion is closed. For more information, particularly on merging or moving the article during the discussion, read the [guide to deletion](#).

Steps to list an article for deletion: 1. {{subst:afd}} 2. {{subst:afd2|pg=**Benjamin M. Emanuel**|cat=|text=|}} ~~~~ (*categories*) 3. {{subst:afd3|pg=**Benjamin M. Emanuel** (2nd nomination)}} (add to **top** of *list*) 4. Please consider notifying the *author(s)* by placing {{subst:adw/**Benjamin M. Emanuel**/**Benjamin M. Emanuel** (2nd nomination)}} ~~~~ on their talk page(s).

Benjamin M. Emanuel (nã© Auerbach) is a [Chicago](#) pediatrician and former member of [Irgun](#)^{[1][2]}. He is the father of [NIH](#) bioethicist [Ezekiel J. Emanuel](#), U.S. Congressman and White House Chief of Staff–designate [Rahm Emanuel](#), talent agent [Ari Emanuel](#), and adopted daughter Shoshana.^[1] **Benjamin** Auerbach was born in [Jerusalem](#) in 1927,^[1] the son of pharmacists who had fled Russian pogroms.^[3] The family adopted the surname **Emanuel** in 1933, after **Benjamin**’s brother, **Emanuel** Auerbach, was killed in a skirmish with Arabs in Jerusalem.^[1] In the 1940s, **Benjamin Emanuel** interrupted his medical school training in [Switzerland](#) to take part in an unsuccessful scheme to smuggle guns from [Czechoslovakia](#) to the Israeli underground.^[3] He later served as a medic in the 1948 Israeli war of independence.^[3] In 1953, his medical training brought him to Chicago's Mount Sinai hospital, where he met X-ray technician Marsha Smulevitz, the daughter of a Moldavian immigrant and union organizer.^{[3][4]} The couple married and briefly lived in Israel before returning to [Chicago](#).^[3] They had three sons within four years, and according to **Benjamin Emanuel**, named their second son in honor of Rahamim, a [Lehi](#) combatant who was killed.^[2] They later moved to [Wilmette](#) and adopted their daughter.^[1] **Benjamin** sent his sons to summer camp in Israel, and Marsha insisted they take ballet lessons and accompany her to civil rights protests, where she was arrested three times.^[3] Dr. **Emanuel's** pediatrics practice grew to one of the largest in Chicago.^[3]

[edit] Quotes

"Obviously he [[Rahm Emanuel](#)] will influence the president to be pro–Israel. Why wouldn't he be? What is he, an [Arab](#)? He's not going to clean the floors of the [White House](#)." ^{[4][5][6][7]}

[edit] References

- ↑ *a b c d e* "Elisabeth Bumiller (1997–06–15). "The Brothers **Emanuel**", *New York Times*. Retrieved on Nov. 6, 2008
- ↑ *a b* Anshel Pfeffer and Shlomo Shamir (November 6, 2006). "Obama's first pick: Israeli **Rahm Emanuel** as chief of staff", *Haaretz*. Retrieved on November 6, 2008.
- ↑ *a b c d e f g* Nina Easton (2006–9–25). "Rahm **Emanuel**, Pitbull politician", *Fortune Magazine*. Retrieved on Nov. 8, 2008

