



(22) Date de dépôt/Filing Date: 1993/11/16

(41) Mise à la disp. pub./Open to Public Insp.: 1994/05/18

(45) Date de délivrance/Issue Date: 2002/11/12

(30) Priorité/Priority: 1992/11/17 (977,779) US

(51) Cl.Int.⁵/Int.Cl.⁵ H04Q 3/64

(72) Inventeurs/Inventors:

Baker, Daniel F., US;
Steinlicht, Joseph C., US;
Holtaway, Thomas S., US

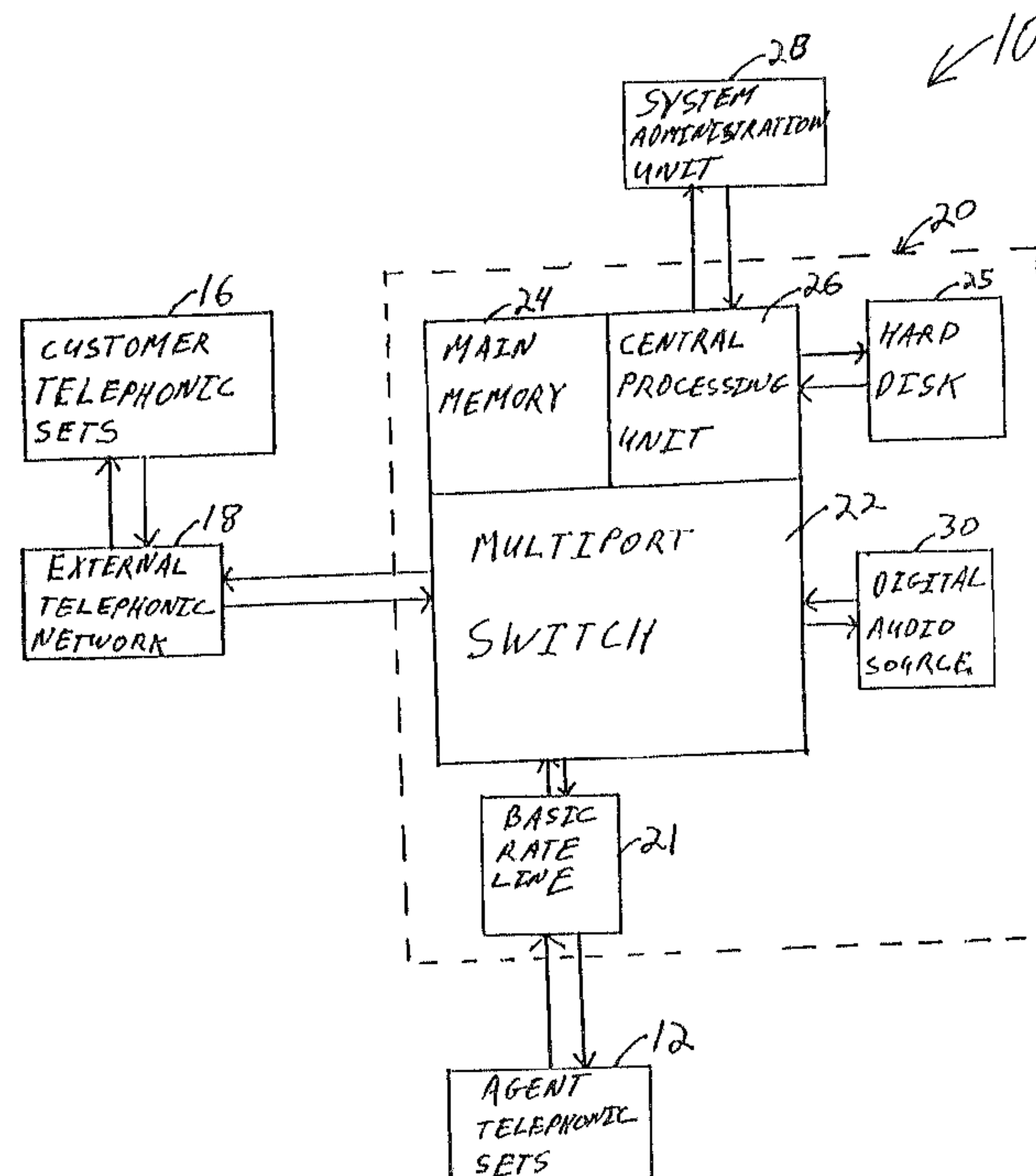
(73) Propriétaire/Owner:

ROCKWELL INTERNATIONAL CORPORATION, US

(74) Agent: SHAPIRO, COHEN

(54) Titre : DISTRIBUTEUR D'APPELS A MESSAGES AUTOMATIQUES

(54) Title: CALL DISTRIBUTOR WITH AUTOMATIC PREANNOUNCEMENT SYSTEM AND METHOD



(57) Abrégé/Abstract:

An automatic call distributor (20) connected with a plurality of agent telephonic sets (12) for audio communication with customers at customer telephonic sets (16) of an external telephonic switching system (18), visual displays (32) on the agent telephonic sets (12) for display of information concerning customers with an automated preannouncement system (10) having a hard disk (25) for storing preannouncement voice messages in a voice of a particular agent, a sign-in code stored in a main memory (24) of the automatic call distributor (20) associated with the particular agent, a key pad (34) at the agent sets (12) for entering in a sign-in code, and a digital audio source card circuit (30) for playing a preannouncement voice message received from the hard disk (25) in the voice of the particular agent associated with the sign-in code to a customer at a customer set (16) connected by the automatic call distributor (20) to the agent set (12) at which the sign-in code of the particular agent has been entered.



2103204

ABSTRACT OF THE DISCLOSURE

An automatic call distributor (20) connected with a plurality of agent telephonic sets (12) for audio communication with customers at customer telephonic sets (16) of an external telephonic switching system (18), visual displays (32) on the agent telephonic sets (12) for display of information concerning customers with an automated preannouncement system (10) having a hard disk (25) for storing preannouncement voice messages in a voice of a particular agent, a sign-in code stored in a main memory (24) of the automatic call distributor (20) associated with the particular agent, a key pad (34) at the agent sets (12) for entering in a sign-in code, and a digital audio source card circuit (30) for playing a preannouncement voice message received from the hard disk (25) in the voice of the particular agent associated with the sign-in code to a customer at a customer set (16) connected by the automatic call distributor (20) to the agent set (12) at which the sign-in code of the particular agent has been entered.

BACKGROUND OF THE INVENTION

Field of the Invention

This invention relates generally to the field of automatic telephone call distributors and, more particularly, to an automatic call distributor with an automatic preannouncement system.

Automatic call distributors employing a multiport switch controlled by a central processing unit for selectively interconnecting a plurality of agent sets, each having a telephone, with customer telephonic sets of an external telephonic switching system are well known. Examples of such call distributor systems are shown in United States Patent No. 5,268,903 of Jones et al. entitled "Multichannel Telephonic Switching Network With Different Signaling Formats and Connect/PBX Treatment Selectable For Each Channel", U.S. Patent No. 5,140,611 of Jones et al. entitled "Pulse Modulated Self-Clocking and Self-Synchronizing Data Transmission and Method for a Telephonic Communication Switching System", issued August 18, 1992 and U.S. patent No. 5,127,004 of Lenihan et al. entitled "Tone and Announcement Message Code Generator for a Telephonic Switching System and Method", issued June 30, 1992.

Typically an agent stationed at an agent set in known distributor systems answers many different types of calls from customers during a work day. In these known systems the agent receives information about the type of customer call on a visual display at the agent set when a call is distributed to the agent. The agent reads the information on the display to determine what type of call (i.e. sales, inventory, customer service, billing etc.) he is receiving. When answering a call

the agent greets the caller with a standard greeting which is usually specific to the application being served. For example, an agent may answer a sales call with the greeting, "Welcome to sales. May I help you?" Furthermore, the same agent may also answer calls of a different application such as customer service and present the caller with the standard greeting of "Welcome to customer service. May I help you?"

To help relieve the burden on agents of repeating the same standard greetings hundreds of times during a work shift, the known call distributor systems provide external hardware preannouncers which answer customer calls from customer units of an external network. In one known system, the voice message is stored at the agent set. Each time the agent moves to a different unit, a new voice message must be recorded at the new set which reduces agent efficiency.

A solution to this problem in system in systems with individual preannouncement at each agent set is shown in U.S.

Patent No. 5,539,818 of Zdenek et al., issued July 23, 1996, entitled "Telephonic Console With

Prerecorded Voice Message and Method" in which prerecorded voice messages are stored on a memory disk or card which is carried by the agent who prerecorded the message. The memory disk is releasably connectable to any play back unit of any of the agent sets to eliminate the need to re-record the message when there is a change of agents between different agent sets. Since these preannouncers are external and are not internally linked within the system, the agent must remove these disks and carry them when changing shifts or moving from one telephonic agent set to another. This creates a possible risk of an agent misplacing, losing or damaging the external preannouncer disk, the replacement of which is time consuming and costly. Additionally, if the agent forgets to bring the preannouncer memory disk to work or when they shift between agent sets, time must be taken for new messages to be recorded on another disk or for the original disk to be retrieved.

SUMMARY OF THE INVENTION

It is therefore a principal object of the present invention to provide an automatic call distributor with an automated preannouncement system and method in which the disadvantage of known call distributors noted above are overcome by providing means and methods for automatically playing to the callers at each agent set preannouncement messages in the voice of the particular agent selected for each agent set regardless of at which agent set the particular agent is located.

This object is achieved in part by providing an automatic call distributor, having a plurality of agent sets each with a telephone to communicate with customers at customer sets of an external telephonic switching system and a visual display for receipt of information concerning the customers, with an automated preannouncement system comprising means for storing preannouncement voice messages in the voice of a particular agent, means for associating a sign-in code with the particular agent, means for entering in the sign-in code at the agent set, and means responsive to the sign in code to enable the playing of a voice message in the voice of the particular agent associated with the sign-in code to customers connected by the automatic call distributor to the particular agent at the agent set.

The object is also achieved by providing an automatic call distributor, having a plurality of agent sets each with a telephone to communicate with customers at customer telephonic sets of an external telephonic network, with an automated preannouncement system comprising a voice message memory accessible to the plurality of agent sets, means for internally storing in the voice message memory a plurality of different preannouncement voice messages respectively associated with different types of customers, means for determining the type of customer calling from a customer set and means responsive to the determining means for selectively playing to the customer one of the plurality of different

preannouncement voice messages associated with the type of customer making the incoming call.

The object is further achieved in part by providing an automatic call distributor, having a plurality of agent sets each with a telephone to communicate with customers and a display screen to obtain visual information relating to customer types at customer telephone sets of an external telephonic network, with an automated preannouncement system comprising means for identifying one of a plurality of different customer types, means for storing a plurality of voice messages and a plurality of associated display screen messages respectively associated with the plurality of different customer types, and means responsive to the customer type identifying means for providing one of the plurality of display screen messages to the display screen of the agent unit associated with the type of customer connected thereto for playing one of the voice messages to the associated customer type.

The object is partly obtained by providing in such an automatic call distributor a method of automatically playing preannouncement messages comprising the steps of (1) storing preannouncement voice messages in a voice of a particular agent, (2) associating a sign-in code with the particular agent, (3) entering a sign-in code at one of the plurality of agent sets, and (4) playing the voice message of the particular agent associated with a sign-in code entered at any one of the plurality of agent sets to customers who have been connected by the automatic call distributor to the one agent set at which the sign-in code of the one particular agent has been entered.

The object is also acquired by providing in such an automatic call distributor a method of automatically playing preannouncement voice messages comprising the steps of (1) internally storing in a memory accessible to a plurality of agent sets a plurality of different preannouncement voice messages respectively associated with different types of customers, (2) determining the type of customer call from one

of the plurality of customer telephone sets, and (3) selectively playing to a customer the one of the plurality of different preannouncement voice messages associated with the type of customer determined to be making the incoming call.

The object is further achieved by providing in such a call distributor a method of automatically playing preannouncement messages comprising the steps of (1) identifying one of a plurality of different customer types, (2) storing a plurality of voice messages and corresponding display screen messages respectively associated with the plurality of different customer types, (3) automatically providing a display screen message to the display screen of the agent set associated with the type of customer connected thereto and (4) playing to the connected customer one of the plurality of voice messages which corresponds to the display screen message being provided to the agent set which is associated with the identified customer type of the customer connected thereto.

In a first aspect, the present invention provides an automatic call distributor having a plurality of agent sets each with a telephone to communicate with incoming telephone calls from customers at customer sets of an external telephonic switching system and a visual display for displaying information concerning customers, the improvement being an automated preannouncement system, comprising:

means for storing preannouncement voice messages in different voices of a plurality of different agents respectively;

means for associating a sign-in code with a particular agent of the plurality of different agents;

means at the plurality of agent sets for entering in the sign-in code; and

5A

means responsive to the sign-in code entered at any one of the plurality of agent sets to enable at least one of the preannouncement voice messages in the voice of the particular agent associated with the sign-in code to BE automatically played to incoming telephone calls from customers which have been connected by the automatic call distributor to the one of the agent sets at which the sign-in code of the particular agent has been entered and in which the at least one of the preannouncement voice messages is played in the voice of the particular agent connected to the incoming telephone calls upon the initial connection of the incoming telephone calls by the automatic call distributor to the one of the agent sets.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing objects and advantageous features of the invention will be explained in greater detail and others will be made apparent from the detailed description of the preferred embodiment of the present invention which is given with reference to the several figures of the drawing, in which:

Fig. 1 is a functional block diagram of the preferred embodiment of the automatic call distributor of the present invention as interconnected with a known external telephonic network of customer telephonic units;

Fig. 2 is a simplified plan view of an agent telephonic set of the automatic call distributor of Fig. 1;

Fig. 3A is a flow control diagram of the interrelationship between the software components of the automatic call distributor of Fig. 1 employed in the automated preannouncement system and as described and listed in the

software listings and associated notes in Appendix A attached hereto;

Fig. 3B is a legend for the computer program software component symbols used in Fig. 3A; and

Figs. 4A and 4B are flow charts of the preferred method of recording and playing back the voice messages through interaction between an agent at an agent set and the central processing unit of Fig. 1 and as described and listed in Appendix A attached hereto.

DESCRIPTION OF THE PREFERRED EMBODIMENT

Referring to Figs. 1 and 2, the automatic call distributor 20 is connected with a plurality of agent telephonic sets 12 which communicate with customers, i.e. any caller of the external network 18, at customer, or external, telephonic sets 16 of an external telephonic network 18. The agent sets 12 have a telephone 36, Fig. 2, for audio communication with the customers of the customer sets 16, Fig. 1. Additionally, the agents sets 12 have a visual display 32, Fig. 2, for receipt of information concerning the customers such as the location of the calling customer or the type of products the customer desires to purchase.

The automated preannouncement system 10, Fig. 1, has means for storing preannouncement voice messages in the voice of the particular agent, i.e. any operator or user of an agent set 12. An agent receives a call from a customer set 16 and is connected to the call at an agent set 12 via a multiport switch 22 of the automatic call distributor 20. The agent telephonic sets 12 are connected to the switch 22 through a basic rate line (BRL) interface card circuit 21. The basic rate line card 21 provides for the termination of Integrated Services Digital Network (ISDN) basic rate links from the switch 22. The links are used in the automatic call distributor 20 of the preferred embodiment to connect the ISDN compatible agent sets 12 into the system 10. A Motorola 68000 microprocessor provides for the overall control of the basic

rate line circuit card 21. Generally, while the automated preannouncement system of the present invention can be implemented in numerous types and sizes of call distribution networks, it is preferably employed in a call distribution system of the type shown in United States Patent 5,268,903 of Jones, et al. entitled "Multichannel Telephonic Switching Network with Different Signaling Formats and Connect/PBX Treatment Selectable for Each Channel", United States Patent No. 5,140,611 of Jones, et al. entitled "Pulse Modulated Self-Clocking and Self-Synchronizing Data Transmission and Method for a Telephonic Communication Switching System", issued August 18, 1992; and U.S. Patent No. 5,127,004 of Lenihan, et al. entitled "Toner and Announcement Message Code Generator for a Telephonic Switching System and Method", issued June 30, 1992.

To store a voice message, the agent enters an assigned sign-in code by actuating appropriate numbered keys 38, Fig. 2, on the keypad 34 located on the agent set 12. The agent is assigned an associated sign-in code which is stored in the main memory 24, Fig. 1, of the automatic call distributor 20 at initialization of the system. A system administrator enters in a sign-in code number for each agent at a system administration unit 28 connected with the central processing unit 26 of the call distributor 20 (ACD). The system administration unit 28 is preferably a cathode ray tube (CRT) terminal or a personal computer (PC) and has a keyboard (not shown) for entry of information to the system main memory 24. Using keyboard at the administrator unit 28, the number of and the type of preannouncement voice messages that are assigned to a particular agent is entered into the CPU 26. For example, a particular agent is selectively assigned two preannouncement voice messages, one for sales calls, and another for subscription calls. Alternatively, another agent may also service billing calls and is assigned an additional billing preannouncement voice message. In the preferred embodiment, an agent records up to six voice messages which have a maximum duration of six seconds each.

The system administrator programs the central processing unit 26 by assigning each agent sign-in code number a set of preannouncement voice messages allowed for recording that correspond to the types of customer calls which the particular agent has connected through the ACD 20. This information is entered into the system administration unit 28 and is stored in the memory 24 associated with the central processing unit 26. The central processing unit (CPU) 26 is the primary controller for the automatic call distributor system 20. In the preferred embodiment, the processing power for the CPU 26 is provided by a 32 bit Motorola 68030 microprocessor.

In response to the agent signing in on the key pad 34 of the agent set 12, a menu appears on the visual display 32, Fig. 2, of the set. The agent accesses the voice message menu by actuating the messages key 40 on the key pad 34 of the agent set 12. The agent receives information through the visual display 32 pertaining to his assigned announcements. Only those messages allowed to be recorded appear on the agent set 12 visual display 32 for a particular agent in response to receipt of the agent sign-in code number. The agent records a preannouncement voice message by actuating a RECORD key 42 on the agent set and announces the message into a telephone 36 handset or headset at the agent set. An alternative means of initiating the recording and storing of the preannouncement voice messages is done through actuation of soft keys 41 in which the executed function for the key 41 varies as the menu changes and appears on the LCD display 32. The preannouncement voice message is recorded and digitally stored at the system hard disk 25 using binary pulse coded modulation.

The agent accesses submenus on the LCD visual display 32 through the keypad 34 to record new preannouncement voice messages or to listen to previously recorded messages. In response to receipt of a particular agent sign-in code at the central processing unit 26, Fig. 1, the existing recorded voice announcements for the particular agent are down loaded from the system hard disk 25 to a digital audio source card

(DAS) 30. The hard disk 25 is a magnetic recording medium which digitally stores all the recorded preannouncement voice messages received from the DAS card 30. The storage capacity of the hard disk 25 in the preferred embodiment ranges from 300 mega byte to 1.2 giga-byte of memory. Alternative memory capacities can be utilized for storage without departing from the scope of the invention. Each recorded preannouncement voice message is stored as a file at the hard disk 25. The digital audio source card 30 allows the recording of messages for the purpose of preannouncement voice message features, DTMF and MF digit transmission, message transmission and tone transmission. The received analog audio voice from the agent is converted into a digital signal by pulse code modulation. A message for recording, announced by an agent at an agent set 12 is converted to a digital signal and transferred to the DAS 30 via the switch 12. The digital voice message is up loaded to the hard disk 25 for accessible storage. The digital audio source circuit 30 includes a Motorola 68000 microprocessor operating from a 10 MHz clock to provide the overall controlling functions of the DAS circuit. The DAS card circuit 30 additionally includes an eight megabyte RAM for storage of preannouncement voice messages down loaded from the hard disk 25 upon receipt of an agent sign-in code at the ACD 20. A description of the general operation of the DAS card circuit 30 can be found at with reference to U.S. patent No. 5,127,004 of Lenihan et al. entitled "Tone and Announcement Message Code Generator For a Telephonic Switching System and Method", issued June 30, 1992, and the software listings and associated notes attached in Appendix A.

Once a preannouncement voice message is recorded at an agent set 12, the message is up loaded from the DAS card 30 to the system hard disk 25. A preannouncement voice message number is stored in the main memory 24 of the CPU 26 which corresponds to the recorded and up loaded preannouncement voice message. The preannouncement voice message number represents the particular agent making the recording and the type of customer call to which the preannouncement message is

102103204

responsive (i.e. a sales message for a sales call). In response to the agent sign-in code being entered at one of the plurality of agent sets 12, a preannouncement voice message of the particular agent associated with the sign-in code is played to a customer connected by the automatic call distributor 20 to the agent set 12 at which the signing code was entered. Therefore, upon a customer call being distributed to a particular agent the down loaded preannouncement voice message at the DAS card 30 corresponding to the call is played at the connected customer set 16 and the agent set 12. For example, a customer calling a sales department to make a purchase will receive a stored preannouncement voice message in the voice of the agent at the particular agent set 12 to which the call is connected stating "Welcome to the sales department. How many items would you like to purchase?".

The automated preannouncement system 10 correlates the different preannouncement voice messages with the different type of customers upon receipt of an incoming call through a Dialed Number Identification Service (DNIS) number. It is common practice for many long distance telephone carriers to assign a DNIS number to various 1-800 or toll free telephonic calls coming in through the external network 18, such as American Telephone and Telegraph Co., MCI, Sprint and Northern Telecom. Different 1-800 customer calls are assigned a different corresponding DNIS number which indicates or identifies the type of call, or type of customer, which has reached the multiport switch 22. For example, a customer dialing a New York toll-free number for airline ticket sales, has an assigned DNIS number corresponding to the called dialed number. The DNIS number and the customer call are carried concurrently to the switch 22. In this ACD 20, the receipt of the DNIS number at the automatic call distributor 20 would provide the identification of the dialed number being a sales call for airline tickets from a New York customer and thus would identify the type of customer. Various DNIS numbers are assigned to certain voice messages in order to provide

responsive preannouncement messages. Therefore, receipt of a New York airline ticket sales DNIS number will trigger a preannouncement voice message such as: "Welcome to airline ticket sales. From which airport in New York would you like to depart?". Preferably, the preannouncement voice message is played at both the customer set 16 and the agent set 12, so that the preannouncement not only automatically greets the customer but also informs the agent as to the type of customer call being connected at the agent telephonic set 12. This reduces the hold time a customer waits for an agent, since the agent does not have to read identification information on the visual display 32 in order to determine the type of customer before answering the connected customer call and enables the employment of blind agents.

The preannouncement voice messages are correlated with information concerning the customer that is displayed on the visual display 32 of the agent sets 12. This information concerning the customer (i.e. sales, customer service, billing, etc.) is visually displayed at the agent set while the preannouncement voice message is automatically played at the connected customer set 16 and the agent set 12 servicing the call. A system administrator enters the visually displayed information messages at the system administration unit 28 which are stored in the main memory 24 via the central processing unit 26. The system administrator programs the central processing unit to display at the agent set 12 the information message corresponding to a particular customer call in response to the call being distributed at the agent set. The information message informs and identifies to the agent the type of customer call being connected at the agent set by identifying the customer type. The displayed information aids the agent in responding to the connected call.

An alternative means for associating different stored preannouncement voice messages with different types of customer calls is done by a customer inserting identification information through the customer set 16 based on prompts

received from the ACD 20. The customer inserts information, such as a credit card number, through the numbered key pad on the customer telephonic set 16 in response to a prompt message from a tape player or voice response unit (not shown) upon connection to the automatic call distributor 20. This information received at the call distributor 20 identifies the type of call or customer, such as a gold card customer. Thus, upon receipt of this information, the central processing unit 26 initiates the playing of an appropriate preannouncement voice message in the voice of the particular agent receiving the call. Alternatively, a customer call is identified by designating certain trunks or trunk groups on the switch to represent certain types of calls during the installation of the automatic call distributor 20. A call received on a particular trunk indicates the call type or geographical origin. The central processing unit 26 is programmed to initiate the playing of a responsive preannouncement voice message in the voice of the agent receiving the call in response to connection of the call.

There are different preannouncement voice messages stored at the hard disk 25 which are respectively associated with the different types of customers or customer calls. A plurality of preannouncement voice messages are recorded to be responsive to the different types of calls received, such as sales calls, billing calls, payroll calls, subscription calls, customer service calls, etc. The multiple preannouncement voice messages are stored in a plurality of different agent voices. Different agents record and store a plurality of different preannouncement voice messages in their own voice. The preannouncement voice message played to the customer at a customer set 16 is in the same voice of the particular agent located at an agent set 12 to which the call is distributed and connected. The hard disk 25 stores many identically worded message announcements recorded from different agents in the different agent voices. For example, two different agents each record an identical message announcement for a sales call, each announcing "Welcome to sales. May I help you?".

The identically worded message is stored in the particular voice of each agent making the recording.

The automatic call distributor 20 directs a call received from a customer set 16 to a particular agent at an agent set 12 assigned to service certain customer calls. The assigning of certain calls to particular agents is done by a system administrator through the system administration unit 28. One of the plurality of preannouncement voice messages is selected from those stored at the hard disk 25 in the voice of the particular agent with the entered sign-in code upon the connection of the customer call to the particular agent. The customer set 16 connected through the automatic call distributor 20 is played the one selected preannouncement voice message in the voice of the particular agent associated with the sign-in code entered at the agent set 12. The preannouncement voice message played corresponds to the type of customer call; a sales preannouncement voice message is played at a customer set in response to receipt of a sales call at the automatic call distributor.

The hard disk 25 internally stores in the ACD 20 the different preannouncement voice messages which are respectively associated with the different types of customers. The hard disk memory 25 is accessible to all the agent sets 12. Thus, an agent accesses his assigned stored messages by signing-in at any of the plurality of agent sets. In response to the central processing unit 26 determining or identifying the type of customer calling from a customer telephone set 16 the DAS card 30 plays to the customer one of the different preannouncement voice messages associated with the particular type of customer making the call. Information concerning the customer making the incoming call is visually displayed at the display screen 32 of the agent set 12 while the preannouncement voice message is played to the customer. This stored customer information is in the form of a display screen message respectively associated with the plurality of different customer types. In response to the central processing unit 26 identifying the type of customer making the

call, one of the display screen messages stored in the main memory 24 and associated with the type of customer is provided to the display screen 32 of the agent unit 12. A voice message associated with the customer type of call and in the voice of the agent receiving the call is also played to the customer set 16 through a speaker (not shown). The automated preannouncement system 10 correlates and plays a different preannouncement voice message to a customer set 16 and to the connected agent set 12 in response to receipt of a different type of customer call.

The agent personal preannouncement voice message is recorded on the DAS 30, and stored on the hard disk 25 by a message loader program, Figs. 4A and 4B. On every sign-in, any recorded announcements for the current group assignment for the agent are downloaded to the DAS 30 from the hard disk 25 by the message loader. When an agent records a new announcement, or modifies an existing one, that announcement is stored on the hard disk 25.

Referring to Fig. 3A, the Call Services 60 library state machines cooperate to request the recording of a personal announcement. Call Services 60 requests that the DAS 30, Fig. 1, allocate memory for the preannouncement feature upon sign-in. That request is sent through DAS_CP_MBOX 62, Fig. 3A, which is read by the req_handler 64 process on the DAS.

The req_handler 64 allocates DAS memory as requested, and send a response back to Call Services through the CALL_SERVICES_MBOX 66. The req_handler 64 process allocates DAS memory by using DAS RAM primitives 68.

The requests to start recording and stop recording are also sent through the DAS_CP_MBOX 62 to the req_handler 64, and the responses are also returned through the CALL_SERVICES_MBOX 66. The DAS RAM primitives 68 subroutine are also used to start and stop the recording.

Call Services up loads the recorded announcement to the hard disk 25, Fig, 1, by sending a request to the CP_DAS_ULDR_MBOX 70. Requests are read from this mail box by the bmu_uldr 72 process, which controls the uploading. The

bmu_uldr 72 process cooperates with the das_msg_uldr 74 process which reads from the DAS_ULD_MBOX 76.

The das_msg_uldr 74 process accesses an announcement on the DAS memory 78 by using the DAS RAM primitives 68 subroutine and the global DAS tables 80. The das_msg_uldr 74 sends the message back the to bmu_uldr 72 piece by piece in messages sent through the CP_DAS_ULDR_MBOX 70 on the central processing unit 26.

The bmu_uldr 72 writes the message to the hard disk 25 and stores information about the message in a message control shared data table 82.

Movement of messages from the disk 25 to the DAS 30 is controlled by the Call Service state machine 60 which allocates space by sending events to the DAS_CP_MBOX 62, and then requesting the bmu_dnldr 84 process to move the message to disk. These requests are sent to the bmu_dnldr 84 through the CP_DAS_LDR_MBOX 86.

The bmu_dnldr 84 process uses information in the message control table 82 to get the message from the hard disk 25. The message is sent piece by piece in events to the DAS_LD_MBOX 88, from which they are read by the das_msg_ldr 90 process. The das_msg_ldr 90 process uses the DAS RAM primitives 68 subroutine to write the messages to the DAS memory 78.

The das_msg_ldr 90 sends responses back to the bmu_dnldr 84 through the CP_DAS_LDR_MBOX 86.

Events are sent from the BRL 21 to the central processing unit 26 CALL_SERVICES_MBOX 66 to setup a call to the Messages key 40, Fig. 2, on an agent telephonic unit 12. The BMU or CPU Call Services capabilities control the menus on an agent unit 12 through events sent to the CALL_CONTROL_MBOX 92 on the BRL 21. These are events read by the bprh_driver 94 essential task, which calls the appropriate key filter or menu handler state machines. The Agent Voice Services Main and Sub Menu Handlers interacts with an agent recording, playing and erasing their personal announcement messages. For further details covering the interrelationship between the software

components in the automatic call distributor of Fig. 3A and the recording and playing back of voice messages through interaction of an agent at an agent set and the central processing unit as seen in Figs. 4A and 4B, reference should be made to the software listings and related notes of Appendix A attached hereto.

While the advantages of the invention are preferably obtained with the automatic call distributor 20 described above with reference to Fig. 1, the method of the invention can be practiced with any other automated call distributors having a switch for selectively interconnecting a plurality of agent sets with customer sets of an external telephonic switching system network. In any event, the preferred method of practicing the invention comprises the steps of (1) storing preannouncement voice messages in a voice of a particular agent, (2) associating a sign-in code with the particular agent, (3) entering the sign-in code at any one of the plurality of agent sets, and (4) playing a voice message of the particular agent associated with the sign-in code entered at any one of the plurality of agent sets to customers who have been connected by the automatic call distributor to the one agent set at which the sign-in code of the one particular agent has been entered.

Also, a method is provided of automatically playing preannouncement voice messages comprising the steps of (1) internally storing in a memory accessible to a plurality of agent sets a plurality of different preannouncement voice messages respectively associated with different types of customers, (2) determining the type of customer call from one of the plurality of customer telephone sets, and (3) selectively playing to a customer the one of the plurality of different preannouncement voice messages associated with the type of customer determined to be making the incoming call.

Also, preferably performed are the steps of (1) storing a plurality of voice messages and corresponding display screen messages respectively associated with a plurality of different customer types, (2) automatically providing one of the

plurality of display screen messages to the display screen of the agent set associated with the type of customer connected thereto, and (3) playing to the connected customer one of the plurality voice messages which corresponds to the display screen message being provided and is associated with the customer type in response to receipt of information identifying the identified customer type.

Referring to Fig. 4A, the method of recording and storing a plurality of preannouncement voice messages is done in the automatic call distributor 20 by an agent entering a sign-in code at an agent telephonic set 12 while in an idle and unavailable state to service customer calls, as seen in step 100. In step 102, the Messages key 40, Fig. 2, on the agent set 12 is actuated and the central processing unit 26 accesses the messaging system software in step 104, Fig. 4A, to perform recording. In step 106, the agent actuates the Persannc Key 44, Fig. 2, to obtain a menu of the preannouncement voice messages and the corresponding trigger number which are assigned to the particular agent and are shown on the visual display 32.

A sign-in code is associated with each particular agent and stored in the main memory 24 of the automatic call distributor 20 through the system administrator unit 28. The system administrator assigns the messages for recording which correspond to different types of received calls from the customer telephonic sets 16. The agent records a message as directed by the menu appearing at the visual display which is responsive to a particular type of customer call identified by a trigger received at the automatic call distributor. In step 108, Fig. 4A, the agent scrolls the visually displayed menu using a scroll key 45, Fig. 2, on the agent set 12 to obtain an unrecorded message.

In step 110, Fig. 4A, the agent determines if all the assigned messages are recorded. If all the assigned preannouncement voice messages are recorded by the agent, then the agent is prepared to service incoming customer calls. The agent goes from the unavailable or idle state by actuating an

Available Key 46, Fig. 2, on the agent set 12 to become available to receive and service customer calls, as seen in step 134 of Fig. 4B. If all the messages assigned to the agent are not recorded, then in step 112, Fig. 4A, the agent selects the next unrecorded message from the menu on the display screen 32, Fig. 2, by utilizing the scroll keys 45.

In step 114, the RECORD key 42, Fig. 2, is actuated to indicate the recording of the preannouncement voice message. In response to the actuation of the RECORD Key 42, recording space is made available in the memory of the Digital Audio Source Card 30, Fig. 1, as seen in step 116, Fig. 4A. In step 118, the agent announces the appropriate voice message into a telephone 36, Fig. 2, handset or headset at the agent telephonic set 12, and the message is recorded on the DAS card 30. The stop key 48, Fig. 2, is actuated upon completion of the recording of the voice message in step 120. In step 122, Fig. 4A, the store key 50 is actuated to store the recorded preannouncement voice message in the voice of the agent at the set 12. In step 124, Fig. 4A, the recorded message is up loaded from the DAS 30, Fig. 1, and saved at the hard disk 25. The preannouncement voice message is digitally stored at the hard disk through the use of pulse coded modulation.

After completing the recording of a voice message the agent returns to step 108, Fig. 4A, to scroll the menu to record and store different preannouncement voice messages respectively associated and responsive to different types of customer application calls. The hard disk 25 internally stores multiple preannouncement voice messages in different voices which correspond to the different agents recording the voice messages. The hard disk 25 memory is accessible to the plurality of agent sets 12, Fig. 1, for storing the different preannouncement voice messages. Often, these stored preannouncement voice messages are identically worded message announcements recorded in the voices of the different agents.

Referring the Fig. 4B the method of automatically playing preannouncement voice messages is accomplished by the agent

signing-in on the numeric key 38, Fig. 2, of the agent telephonic set 12, as seen in step 130. In response to the agent signing in on the telephonic set 12, Fig. 1, the recorded preannouncement voice messages of the agent are downloaded from the hard disk 25 to the digital audio source card 30 as seen in step 132, Fig. 4B. In step 134, the Available Key 46, Fig. 2, is actuated to indicate to the central processing unit 26, Fig. 1, that the agent is available to service incoming customer calls from customer telephonic sets 16. In step 136, Fig. 4B, a customer telephonic call and an appropriate call identification trigger (DNIS number, customer prompt, designated trunk line, etc.) is received on the multiport switch 22, Fig. 1. In response to the call being received on the switch 22, the central processing unit 26 determines the type of customer call based on the received call identification trigger in step 138, Fig. 4B.

In step 140, the central processing unit 26 determines if there is a recorded preannouncement voice message associated with the type of received customer call for the particular signed-in agent. If a different type of customer call is received on the switch 22, the central processing unit 26 associates or correlates a different preannouncement voice message with the different call. If there is no recorded preannouncement voice message associated with the type of customer calling, then the call is connected directly to the agent set 12, Fig. 1, in step 142, Fig. 4B, and the agent services the customer call in step 144.

If there is a recorded message associated with the received customer call, the call is connected to the agent set 12, Fig. 1, in step 146, and information concerning the customer is displayed on the visual display 32 of the agent set, as seen in step 148. This customer information or display screen message is stored in the automatic call distributor 20 main memory 24, Fig. 1, and is respectively associated with the plurality of different customer calls or customer types.

In response to receipt of a call identification trigger

the central processing unit 26 automatically provides one of a plurality of display screen messages to the display screen 32, Fig. 2, which is associated with the type of customer call connected to the agent set 12 via the automatic call distributor 20. In step 150, Fig. 4B, the preannouncement voice message associated with the type of customer call is played at both the agent set 12, Fig. 1, and customer set 16, in the voice of the particular agent having entered the sign-in code at the agent set.

The preannouncement voice message is automatically played while the information concerning the customers is visually displayed at the agent set 12 at which the sign-in code is entered. The preannouncement voice message and the displayed information message alert the agent as to the type of customer call connected via the automatic call distributor 20. The agent does not have to repeat the same greeting messages over and over again during the working shift for a variety of different application customer calls. After the preannouncement voice message is played, the agent services the customer call in step 144.

While a detailed description of the preferred embodiment of the invention has been given, it should be appreciated that many variations can be made thereto without departing from the scope of the invention as set forth in the appended claims.

Foot

2103204

4
106 253 STATE SDOG_check_port
164 360 STATE SDOG_check_request
9 96 STATE SDOG_idle
118 283 STATE SDOG_no_port_req
12 62 STATE SDOG_pos
50 166 STATE SDOG_port_no_req

REL : ASCII STATE MACHINE

92/11/89
12:19:08

/tmp/59613

APPENDIX A to
U.S. Patent Application of
Baker & Steinlicht entitled
"Call Distributor With Auto-
matic Preannouncement System
and Method"

Docket 91CR114

1

```
1 /*****
2
3
4 REL 2 ASCII STATE MACHINE
5
6 ****
7
8 State Machine File Prologue
9
10 ****
11 ****
12 /****
13
14 Pseudo LGH (SDOG) State Machine
15
16 SCI of DID: G3KE-09-0193
17
18
19 Description: The Pseudo LGH handles two types of queues, a generic
20 pseudo queue and an emergency pseudo queue.
21 The emergency pseudo port gets mapped in when SDOG come
22 in service. The ACD-Mall, Personal Announcement and
23 Predictive Dial pseudo port gets mapped when a capability
24 makes a request to acquire the pseudo port.
25
26 Design Notes:
27 The pseudo ports gets mapped as a generic pseudo port
28 during subinit.
29
30 SDOG_COMMON USE_NEW_ID is a generic queue id.
31 SDOG_EMT_USE_NEW_ID is an emergency queue id.
32 This define is located in _hdr/cp_cbk_d.h
33
34 These pseudo queues do not have an assign queue since
35 there is range in sys_parms.h. max_logical_ports-max_ports
36 are the pseudo ports defined on the system.
37
38 If the available queue is empty, then the request to acquire
39 a pseudo port for Predictive Call or pre-transfer of ACD-Mall
40 messages does not get queue, all other features get queued.
41
42 ****
43
```

```
44
45 /****
46
47 * SDOG_00s
48 *
49 * Description: This state indicates the device is out of service.
50 * The LGH is not available to service any request.
51 *
52 * Resources Active: none
53 *
54 *
55 * Critical Dependencies:
56 * The SDOG maps the emergency ports if the emergency queue
57 * is empty and emergency TAG group indicates an emergency
58 * feature on a DAS(s).
59 *
60 ****
61
62 STATE SDOG_00s
63
64 EVENT ANY_GROUP_IN_SERVICE
65 FUNCTION SDOG_set_to_sdc_port
66 FUNCTION SDOG_install_all_sdc_port
67 TRANS <SDOG_port_no_req>
68
69 DONE SDOG_00s.tde
70
```

71 /
72 *****
73 *
74 * SDOG_idle
75 *
76 * Description: This state indicates the LCH is ready to process requests.
77 * The pseudo port is not available and the request is not
78 * pending.
79 *
80 * Resources Active: none
81 *
82 * Critical Dependencies:
83 * When the SDOG receives LCH_IDLE on the generic pseudo queue,
84 * the SDOG maps the pseudo port to generic device type 15 the
85 * port was mapped other than generic device type, then it add
86 * this pseudo port to the generic available queue.
87 *
88 * When the SDOG receives LCH_IDLE on the emergency pseudo queue,
89 * the SDOG just add this pseudo port to the emergency available
90 * queue.
91 *
92 *****
93
94
95
96
97

37 134 FUNCTION SDOG_rc_queue_request
38 135 TRANS <SDOG_nc_port_req>
39 136
40 137 EVENT LCH_RC_ELT_EMGY_PORT
41 138 FUNCTION SDOG_rc_queue_request
42 139 TRANS <SDOG_nc_port_req>
43 140
44 141 EVENT APT_STABLE_CALL
45 142 FUNCTION SDOG_cmc_any_stable_call
46 143 TRANS SDOG_cmc_restart
47 144
48 145 TRANS <SDOG_idle>
49 146
50 147 EVENT LCH_STABLE_ACK
51 148 TRANS <SDOG_idle>
52 149
53 150 TRANS SDOG_idle

2103204

STATE SDOG_idle
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
218

2103204

```
151 /*****
152 /*****
153 * SDOG_port_no_req
154 *
155 *
156 * Description: This state indicates that there are at least one pseudo
157 *               port available and the request is not pending.
158 *
159 * Resources Active: none
160 *
161 * Critical Dependencies:
162 *               none
163 *
164 *****/
165
166 STATE SDOG_port_no_req
167
168 EVENT LGH_IDLE
169 FUNCTION
170 TRANS <SDOG_port_no_req>
171
172 EVENT LGH_ACQUIRE_ACD_MALL
173 FUNCTION
174 TRANS <SDOG_chk_if_port_avail>
175
176 EVENT LGH_ACQUIRE_ACD_MALL_TN
177 FUNCTION
178 TRANS <SDOG_chk_if_port_avail>
179
180 TRANS <SDOG_chk_port>
181
182 EVENT LGH_ACQUIRE_PIER
183 FUNCTION
184 TRANS <SDOG_chk_port>
185
186 EVENT LGH_ACQUIRE_PREDICTIVE_CALL
187 FUNCTION
188 TRANS <SDOG_chk_if_port_avail>
189
190 TRANS <SDOG_chk_port>
191
192 EVENT LGH_ACQUIRE_EMGY_RECORD
193 FUNCTION
194 TRANS <SDOG_chk_if_port_avail>
195
196 TRANS <SDOG_chk_port>
197
198 EVENT LGH_ACQUIRE_EMGY_REVIEW
199 FUNCTION
200 TRANS <SDOG_chk_if_port_avail>
201
202 EVENT LGH_ACQUIRE_PA
203 FUNCTION
204 TRANS <SDOG_chk_if_port_avail>
205
206 TRANS <SDOG_chk_port>
207
208 EVENT LGH_ACQUIRE_SYS_ANN
209 FUNCTION
210 TRANS <SDOG_chk_if_port_avail>
211
212 EVENT LGH_RC_ADD_EMGY_PORT
213 FUNCTION
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
```

2103204

237	237	269	269
238	/*****	270	270
239	SDOG_check_port	271	271
240		272	272
241	Description: This is a transition state, it indicates that a check has	273	273
242	been made in the previous state to determine whether there	274	274
243	is a pseudo port available.	275	275
244		276	276
245	Resources Active: none	277	277
246		278	278
247	Critical Dependencies: none	279	279
248		280	280
249	*****	281	281
250	STATE SDOG_check_port	282	282
251		283	283
252	14	284	284
253	EVENT SDOG_1_PORT	285	285
254	TRANS <SDOG_port_no_req>	286	286
255		287	287
256	EVENT SDOG_1_NO_PORT	288	288
257	TRANS <SDOG_idle>	289	289
258		290	290
259	EVENT SDOG_1_NO_PORT	291	291
260	TRANS <SDOG_idle>	292	292
261		293	293
262	EVENT ANY_STABLE_CALL	294	294
263	FUNCTION SDOG_remove_any_stable_call	295	295
264	FUNCTION SDOG_remove_request	296	296
265	TRANS <SDOG_check_port>	297	297
266		298	298
267	EVENT SDOG_check_port_idle	299	299
268		300	300
		301	301
		302	302
		303	303
		304	304
		305	305
		306	306
		307	307
		308	308
		309	309
		310	310
		311	311
		312	312
		313	313
		314	314
		315	315
		316	316
		317	317
		318	318
		319	319
		320	320
		321	321
		322	322
		323	323
		324	324
		325	325
		326	326
		327	327
		328	328
		329	329
		330	330
		331	331

```
332 TRANS <SDOG_no_port_req>
333
334 EVENT ANY_STABLE_CALL SDOG_time_any_stable_call
335 FUNCTION SDOG_time_restart
336 FUNCTION SDOG_time_restart
337 TRANS <SDOG_no_port_req>
338
339 EVENT ANY_STABLE_ACR
340 TRANS <SDOG_no_port_req>
341
342 DONE SDOG_no_port_req.tde
343
```

```
344
345 /*****
346 *
347 * SDOG_check_request
348 *
349 * Description: This is a transition state, it indicates a check has
350 * been made in the previous state to determine whether
351 * there is any more request in the queue.
352 *
353 * Resources Active: none
354 *
355 * Critical Dependencies: none
356 *
357 *****/
358
359
360 STATE SDOG_check_request
361
362 EVENT SDOG_I_REQ
363 TRANS <SDOG_no_port_req>
364
365 EVENT SDOG_I_NO_REQ
366 FUNCTION SDOG_set_rc_sdc_port
367 TRANS <SDOG_idle>
368
369 EVENT ANY_STABLE_CALL
370 FUNCTION SDOG_time_any_stable_call
371 FUNCTION SDOG_time_restart
372 FUNCTION SDOG_chk_if_request
373 TRANS <SDOG_check_request>
374
375 DONE SDOG_check_request.tde
376
377 /** SDCS UNPROCESSED COMMENTS BEGIN **
378 SDCS UNPROCESSED COMMENTS END **/
379 /** SDCS PROCESSED COMMENTS BEGIN **
380 03/17/92 chng created
381 SDCS PROCESSED COMMENTS END **/
```

2103204

92/11/09
13:18:15

/tmp/59523

62	208	STATE	AMG_CHK_FWD_C
63	175	STATE	AMG_CHK_JUDEDE
9	97	STATE	AMG_IDLE
2	70	STATE	AMG_DOS
71	241	STATE	AMG_PXFR_T_C
4			

REL 2 ASCII STATE MACHINE

2103204

2103204

```
1 /*****
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
```

REL 2 ASCII STATE MACHINE

State Machine File Prologue

ACD Mail LGH

SCR of DID: GARE-09-0191

Description: The ACD Mail LGH manages the ACD Mail Message Queue and removes ACD Mail messages from DAS memory.

Design Notes of Interest:

02/05/92 Original

Prologues for Individual States.

```
36
37 /*****
38
39 *
40 *
41 *
42 *
43 *
44 *
45 *
46 *
47 *
48 *
49 *
50
51
52
53 /*****
54 *
55 *
56 *
57 *
58 *
59 *
60 *
61 *
62 *
63 *
64 *
65 *
66
67
68
69
70 STATE
71
72
73
74
75
76
77
78
79
```

AMG_OOS

Description:

In this state, the machine receives the set in service event and sets up an initial timeout.

Resources Active:

Critical Dependencies:

AMG_OOS

Description:

In this state, the machine receives the set in service event and immediately checks the pre-transfer conditions.

Resources Active:

Critical Dependencies:

STATE

EVENT ANY_GROUP_IN_SERVICE

FUNCTION AMG_Init_IV_data

FUNCTION TRANS <AMG_PXFE_T_O>

AMG_check_for_msg_after

DONE

AMG_OOS.tde

/imap/59521



Line	Address	Disassembly	Comment
45	143	EVENT LGH_UPLOAD_COMPLETE	
46	144	FUNCTION	AMG_update_msg_avail
47	145	TRANS <AMG_IDLE>	
48	146	TRANS	
49	147	EVENT LGH_RC_DELETE_MSGO	
50	148	FUNCTION	AMG_rc_delete_msgq
51	149	TRANS <AMG_IDLE>	
52	150	TRANS	
53	151	DONE	
54	152	AMG_IDLE.tde	
55	153		
56	154		
57	155		
58	156		

2103204

```
157 /*****
158 *
159 *
160 *   AMG_CHK_QUEUE
161 *
162 *   Description:
163 *
164 *   In this state, the machine gets the result of a check to see
165 *   whether a just received message should be removed from DAS.
166 *
167 *   Resources Active:
168 *
169 *   Critical Dependencies:
170 *
171 * *****/
172
173
174
175 STATE   AMG_CHK_QUEUE
176
177 EVENT   AMG_I_OVER_LIMIT
178         FUNCTION
179         TRANS <AMG_IDLE>
180
181 EVENT   AMG_I_NOT_OVER_LIMIT
182         FUNCTION
183         TRANS <AMG_IDLE>
184
185 DONE   AMG_CHK_QUEUE_rde
186
187
188
189
```

```
190 /*****
191 *
192 *
193 *   AMG_CHK_FWD_Q
194 *
195 *   Description:
196 *
197 *   In this state, the machine receives an internal event to tell it
198 *   whether or not to remove a just forwarded message from DAS.
199 *
200 *   Resources Active:
201 *
202 *   Critical Dependencies:
203 *
204 * *****/
205
206
207
208 STATE   AMG_CHK_FWD_Q
209
210 EVENT   AMG_I_OVER_LIMIT
211         FUNCTION
212         TRANS <AMG_IDLE>
213
214 EVENT   AMG_I_NOT_OVER_LIMIT
215         FUNCTION
216         TRANS <AMG_IDLE>
217
218 DONE   AMG_CHK_FWD_Q_rde
219
220
221
222
```

2103204

```
223 /*****
224 *
225 *   AMG_PXFF_T_O
226 *
227 *   Description:
228 *
229 *   In this state, the machine receives an internal event to tell it
230 *   whether or not to pre-transfer messages to DAS. A new timeout
231 *   is set-up in either case.
232 *
233 *   Resources Active:
234 *
235 *   Critical Dependencies:
236 *
237 *   *****/
238
239
240
241 STATE   AMG_PXFF_T_O
242
243
244
245
246
247
248
249
250
251
252
253
254
```

```
255
256
257
258
259
260 02/11/92 aceto
261 02/22/92 aceto
262 03/16/92 aceto
263
264 04/03/92 comstock
265
266 04/24/92 aceto
267 05/04/92 aceto
268 06/24/92 chung
269
270 06/30/92 chung
271
272
```

```
/** SCCS UNPROCESSED COMMENTS BEGIN **
SCCS UNPROCESSED COMMENTS END **/
/** SCCS PROCESSED COMMENTS BEGIN **
File created.
Added AMG_OOS
Added ANY_GROUP_FORCE_OOS events, LGH_REQUEST_MSG and
LGH_PXFF_COMPLETE.
Changed OOS state to check for pre-transfer conditions
Immediately upon set in service instead of timer set.
Removed ANY_GROUP_FORCE_OOS handling.
Fixed function name.
added LGH_RC_DELETE_MSGO event to AMG_IDLE, and
added AMG_update_msg_avail.
Removed ANY_UPLOAD_FAILURE case.
SCCS PROCESSED COMMENTS END **/
```

2103204

1 1 STATE SDC_IDLE

92/11/09
12:17:31

/tmp/59403

1

1	STATE	SDC_IDLE	
2	EVENT	ANY_SET_IN_SERVICE	
3	TRANS	<SDC_IDLE>	
4	EVENT	IDH_RESTART	
5	TRANS	<SDC_IDLE>	SDC_tmrc_restart_ack
6	EVENT	ANY_STABLE_CALL	
7	TRANS	<SDC_IDLE>	SDC_tmrc_stable_ack
8	EVENT	FUNCTION	
9	TRANS	<SDC_IDLE>	
10	DONE	SDC_IDLE_ack	

2103204

39	151	STATE	PAD_chk_15_pa_req_ent
15	339	STATE	PAD_chk_339_complete
2	57	STATE	PAD_idle
163	838	STATE	PAD_req_delete_req
32	384	STATE	PAD_wait_319_act_ack1
50	446	STATE	PAD_wait_419_act_ack1
81	250	STATE	PAD_wait_download
17	95	STATE	PAD_wait_request
56	191	STATE	PAD_wait_tag_ack
58	636	STATE	PAD_wait_upld_act_ack
27	576	STATE	PAD_wait_upld_ackack1
28	776	STATE	PAD_wait_upld_imp_ack1
93	708	STATE	PAD_wait_upld_imp_ack1
92	511	STATE	PAD_wait_upldack

2103204



2103204

```
78
79
80 /*****
81 *
82 *   PAD_wait_request
83 *
84 *   Description:      This state waits for an event from requestor
85 *                     indicating the specific action to be performed.
86 *
87 *   Resources Active: none
88 *
89 *   Critical Dependencies: none
90 *
91 * *****/
92
93
94
95 STATE   PAD_wait_request
96
97 EVENT   LDH_ACQUIRE_PA_DOWNLOAD /* PA Download requested */
98 FUNCTION PAD_save_downloaded_msg_info
99 FUNCTION PAD_chk_pa_tag_and_swelfirst
100
101 TRANS   <PAD_chk_if_pa_tag_ack>
102
103 EVENT   LDH_FINISH_PA /* PA Upload requested */
104
105 FUNCTION save_the_message_number_and_the_state_index /*
106 FUNCTION PAD_save_uploaded_msg_info
107
108 * set pseudo port id in the msg info table */
109 FUNCTION PAD_set_sdc_in_msg_info_tbl
110
111 * send upload req and set uploading bit in mcr */
112 FUNCTION PAD_snd_das_upload_msg
113
114 TRANS   <PAD_wait_upload>
115
116 EVENT   LDH_IDLE
117 FUNCTION
118 TRANS   <PAD_idle>
119
120 *
121 EVENT   AUT_STABLE_CALL
122 FUNCTION PAD_snd_cmc_any_unstable_ack
123 TRANS   <PAD_wait_request>
124
125 EVENT   AUT_FORCE_OOS
126 FUNCTION PAD_snd_any_sdc_failure
127 FUNCTION PAD_snd_1gh_idle
128
129 DONE   PAD_wait_request.cde
130
131
132
133 /*****
134 *
135 *   PAD_chk_if_pa_tag_ack
136 *
137 *   Description:      This internal state transitions the state
138 *                     machine based on whether or not a Personal
139 *                     Announcement TAG group exists (has been entered).
140 *                     The tag tone table will be checked for a valid
141 *                     group id (non-null).
142 *
143 *   Resources Active: none
144 *
145 *   Critical Dependencies:
146 *
147 * *****/
148
149
150 STATE   PAD_chk_if_pa_tag_ack
151
152 EVENT   PAD_PA_TAG_ENTERED
153 FUNCTION PAD_snd_tag_1gh_acquire
154 TRANS   <PAD_wait_tag_ack>
155
156 EVENT   PAD_PA_TAG_NOT_ENTERED
157 FUNCTION PAD_snd_any_activate_ack
158 FUNCTION PAD_snd_1gh_idle
159 TRANS   <PAD_idle>
160
161 EVENT   ANY_STABLE_CALL
162 FUNCTION PAD_snd_cmc_any_unstable_ack
163 TRANS   <PAD_chk_if_pa_tag_ack>
164
165 EVENT   ANY_FORCE_OOS
166 FUNCTION PAD_snd_any_sdc_failure
167 FUNCTION PAD_snd_1gh_idle
168 TRANS   <PAD_idle>
169
170 DONE   PAD_chk_if_pa_tag_ack.cde
171
172
```

2103204

```
173
174 /*****
175 *
176 * PAD_wait_tag_ack
177 *
178 * Description:
179 * This state is waiting for an acknowledgement from
180 * the LDH that indicates memory has been allocated for
181 * downloading of the recorded messages and which DAS
182 * port has been assigned.
183 *
184 * Resources Active: none
185 *
186 * Critical Dependencies: none
187 *
188 *
189
190
191 STATE PAD_wait_tag_ack
192
193 EVENT ANY_ACTIVATE_ACK /* DAS port and memory has been assigned */
194
195 FUNCTION PAD_save_tag_port_id
196 FUNCTION PAD_set_das_unit_from_tag_id
197 FUNCTION PAD_snd_any_activate_pa_ack
198
199 /* send download req to DAS and set download bit */
200 FUNCTION PAD_snd_das_load_msg
201
202 TRANS <PAD_wait_download>
203
204 EVENT LDH_IDLE /* DAS port has not been assigned */
205
206 /* send LDH_CANCEL_ACQUIRE to TAG_LSH */
207 FUNCTION PAD_ldh_cancel_acquire
208
209 /* send LDH_IDLE to Pseudo LDH */
210 FUNCTION PAD_snd_ldh_idle
211
212 TRANS <PAD_idle>
213
214 EVENT ANY_ACTIVATE_NACK
215
216 FUNCTION PAD_snd_any_activate_nack
217 FUNCTION PAD_snd_ldh_idle
218
219 TRANS <PAD_idle>
220
221 EVENT ANY_STABLE_CALL
222
223 FUNCTION PAD_snd_tmrc_any_unstable_ack
224
225 TRANS <PAD_wait_tag_ack>
226
227 EVENT ANY_FORCE_OOS
228
229 FUNCTION PAD_snd_any_sdc_failure
230 FUNCTION PAD_ldh_cancel_acquire
231 FUNCTION PAD_snd_ldh_idle
232
233 TRANS <PAD_idle>
234
235 DONE PAD_wait_tag_ack.tde
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250 STATE PAD_wait_download
251
252 EVENT ANY_LOAD_COMPLETE /* Load successful */
253
254 FUNCTION PAD_clr_sdc_in_msg_info_tbl
255
256 /* Increment local z/w counter "number of messages
257 * loaded" then generate internal event to indicate
258 * if there are more PA messages to download */
259 FUNCTION PAD_chk_if_more_to_download
260
261 TRANS <PAD_chk_load_complete>
262
263 EVENT ANY_LOAD_FAILURE
264
265 /* current download is unsuccessful - we're history */
266 FUNCTION PAD_snd_any_sdc_failure
267 FUNCTION PAD_clr_sdc_in_msg_info_tbl
268
269 /* deallocate memory for all personal announcements
270 * other than the one that failed, since das appi will
271 * deallocate for the failed message */
272 FUNCTION PAD_snd_das_mem_deall_other
273 FUNCTION PAD_clr_rdy_to_play_bit_all
274 FUNCTION PAD_snd_ldh_idle
275
276 TRANS <PAD_idle>
277
278 EVENT LDH_ABORT_DOWNLOAD /* PAC says abort the download - agent hit
279 * record button while downloading, we'll
280 * delete all and let agent re-record ALL */
281
282 /* send DAS_LOAD_ABORT to BMU Downloader to */
283 /* abort download of current message */
284 FUNCTION PAD_snd_das_load_abort_dnid
285
286 TRANS <PAD_wait_dld_ack>
287
288 EVENT LDH_IDLE /* when agent signing out, PC deleting PA or
289 * abort download of current message */
290
291 /* send DAS_LOAD_ABORT to BMU DAS Downloader to */
292 /* abort download of current message */
293 FUNCTION PAD_snd_das_load_abort_dnid
```

```
294
295 TRANS <PAD_wait_dld_abt_ack>
296
297 EVENT ANY_STABLE_CALL
298 FUNCTION PAD_snd_tmc_any_unstable_ack
299 <PAD_wait_download>
300
301 EVENT ANY_FORCE_OOS
302 FUNCTION PAD_snd_any_sdc_failure
303 FUNCTION PAD_snd_das15_abrtchld_norsp
304 FUNCTION PAD_clf_sdc_in_msg_info_tbl
305
306 /* deallocate memory for all personal announcements
307 other than the one that failed, since das appl will
308 deallocate for the aborted message */
309 FUNCTION PAD_snd_das_mem_deall_other
310 FUNCTION PAD_clf_rdy_to_play_bit_all
311 FUNCTION PAD_snd_1gh_idle
312 TRANS <PAD_idle>
313
314
315 DONE PAD_wait_download.tde
316
```

2103204

```
317
318
319 /*****
320 *
321 * PAD_chk_load_complete
322 *
323 * Description: This internal state will determine the transition
324 * state based on an internal event generated by a
325 * function executed in the previous state. The
326 * internal event will indicate if there are more
327 * messages to download to the DAS or not.
328 *
329 * Resources Active: none
330 *
331 * Critical Dependencies: Assumes an internal event is generated by the
332 * previous state.
333 *
334 *
335 *****/
336
337
338 STATE PAD_chk_load_complete
339
340
341 EVENT PAD_I_LOAD_MESSAGE /* Download another PA message */
342 /* download next message */
343 FUNCTION PAD_snd_das_load_msg
344 TRANS <PAD_wait_download>
345
346
347 EVENT PAD_I_LOAD_COMPLETE /* a job well done, all PAs downloaded */
348 FUNCTION PAD_snd_das_pa_active
349 FUNCTION PAD_snd_1gh_idle
350 TRANS <PAD_idle>
351
352
353 EVENT ANY_STABLE_CALL PAD_snd_tmc_any_unstable_ack
354 FUNCTION <PAD_chk_load_complete>
355
356
357 EVENT ANY_FORCE_OOS PAD_snd_any_sdc_failure
358 FUNCTION PAD_snd_1gh_idle
359 TRANS <PAD_idle>
360
361 DONE PAD_chk_load_complete.tde
362
```

490
190 491 DONE PAD_wait_dld_abt_ackz.tde
191 492

2103204

```
427  
428  
429  
430 /*****  
431 * PAD_wait_dld_abt_ackz  
432 *  
433 * Description: This state waits for ANY_ABORT_MSG_ACK to  
434 * erase all Personal Announcements and go idle.  
435 *  
436 *  
437 * Resources Active: DAS memory for downloading  
438 *  
439 * Critical Dependencies: none  
440 *  
441 *  
442 *  
443  
444  
445  
446 STATE PAD_wait_dld_abt_ackz  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000
```

```
493
494
495 /*****
496 *
497 *   PAD_wait_upload
498 *
499 *   Description:
500 *       This state waits for the upload to complete.
501 *       Agent could sign out during this time or choose
502 *       to re-record or delete the announcement.
503 *
504 *   Resources Active: DAS memory for recording (PAC acquired it)
505 *
506 *   Critical Dependencies: none
507 *
508 *
509 *
510 *
511 STATE PAD_wait_upload
512
513 EVENT LDH_UPLOAD_COMPLETE
514 FUNCTION PAD_clr_sdc_in_msg_info_tbl
515 FUNCTION PAD_snd_1gh_idle
516 TRANS <PAD_idle>
517
518 EVENT LDH_ABORT_UPLOAD /* agent wants to re-record */
519 FUNCTION PAD_snd_das_load_abort_upld
520 TRANS <PAD_wait_upld_act_ack>
521
522 EVENT LDH_DELETE_PA /* agent wants to delete */
523 FUNCTION PAD_snd_das_load_abort_upld
524 TRANS <PAD_wait_upld_actack>
525
526 EVENT LDH_IDLE /* agent signing out */
527 TRANS <PAD_wait_upld_imp_rel>
528
529 EVENT ANY_UPLOAD_FAILURE
530 /* uploader removes the disk file on an upload failure */
531 FUNCTION PAD_snd_any_sdc_failure
532 FUNCTION PAD_clr_msg_num_in_staff_tbl
533 FUNCTION PAD_clr_sdc_in_msg_info_tbl
534 FUNCTION PAD_release_msg_num
535 TRANS <PAD_idle>
536
537 TRANS <PAD_idle>
538
539 EVENT ANY_STABLE_CALL
540 FUNCTION PAD_snd_tmc_any_unstable_ack
541 TRANS <PAD_wait_upload>
542
543 EVENT ANY_FORCE_OOS
544 /* uploader removes the disk file on an upload abort */
545 FUNCTION PAD_snd_any_sdc_failure
546 FUNCTION PAD_snd_das_load_abort_norsp
547 FUNCTION PAD_clr_sdc_in_msg_info_tbl
548 FUNCTION PAD_clr_ready_to_play_bit
549 FUNCTION PAD_snd_das_mem_deall_sngl
550 FUNCTION PAD_clr_msg_num_in_staff_tbl
551 FUNCTION PAD_release_msg_num
552 TRANS <PAD_idle>
553
554 DONE PAD_wait_upload.tde
555
```

```
556
557
558 /*****
559 *
560 *   PAD_wait_upld_actack
561 *
562 *   Description:
563 *       This state waits for ANY_ABORT_MSG_ACT to
564 *       erase the uploaded Personal Announcement.
565 *       We get here by getting an LDH_DELETE_PA while
566 *       uploading (wait_upload state).
567 *
568 *   Resources Active: DAS memory for recording (PAC acquired it)
569 *
570 *   Critical Dependencies: none
571 *
572 *
573 *
574 *
575 STATE PAD_wait_upld_actack
576
577 EVENT ANY_ABORT_MSG_ACT
578 /* uploader removes the disk file on an upload abort */
579 FUNCTION PAD_clr_msg_num_in_staff_tbl
580 FUNCTION PAD_clr_sdc_in_msg_info_tbl
581 FUNCTION PAD_clr_ready_to_play_bit
582 FUNCTION PAD_snd_das_mem_deall_sngl
583 TRANS <PAD_release_msg_num>
584
585 TRANS <PAD_1gh_idle>
586
587 EVENT ANY_ABORT_MSG_NACT
588 /* uploader removes the disk file on an upload abort */
589 FUNCTION PAD_clr_msg_num_in_staff_tbl
590 FUNCTION PAD_clr_sdc_in_msg_info_tbl
591 FUNCTION PAD_clr_ready_to_play_bit
592 FUNCTION PAD_snd_das_mem_deall_sngl
593 TRANS <PAD_release_msg_num>
594
595 TRANS <PAD_1gh_idle>
596
597 TRANS <PAD_idle>
598
599 TRANS <PAD_idle>
600
601 TRANS <PAD_wait_upld_actack>
602
603 EVENT ANY_FORCE_OOS
604 /* uploader removes the disk file on an upload abort */
605 FUNCTION PAD_snd_any_sdc_failure
606 FUNCTION PAD_clr_msg_num_in_staff_tbl
607 FUNCTION PAD_clr_sdc_in_msg_info_tbl
608 FUNCTION PAD_clr_ready_to_play_bit
609 FUNCTION PAD_snd_das_mem_deall_sngl
610 TRANS <PAD_release msg_num>
611
612 TRANS <PAD_1gh_idle>
613
614 DONE PAD_wait_upld_actack.tde
615
```

2103204

```
687
688
689 /*****
690 *
691 * PAD_wait_upld_cmp_rel
692 *
693 * Description:
694 * This state waits for the upload to complete to
695 * release the DAS memory associated with the
696 * message just uploaded. We come to this state
697 * when PAC says to go idle while upload is in progress.
698 * This would happen if agent signed out while upload
699 * was in progress.
700 *
701 * Resources Active: DAS memory
702 *
703 * Critical Dependencies: none
704 *
705 *****/
706
707
708 STATE PAD_wait_upld_cmp_rel
709
710 EVENT IDH_UPLOAD_COMPLETE
711 FUNCTION PAD_clr_sec_in_msg_info_tbl
712 FUNCTION PAD_clr_ready_to_play_bit
713 FUNCTION PAD_snd_das_mem_deall_sngl
714 FUNCTION PAD_snd_1gh_idle
715 TRANS <PAD_idle>
716
717 * We only get the notify event if an agent
718 * signed out while upload was in progress and
719 * signs back in while upload is still in progress
720 * If agent signs back in at different port, save new port */
721
722 EVENT IDH_NOTIFY_UPLOAD_COMPLETE
723 FUNCTION PAD_save_inform_req_data
724 TRANS <PAD_wait_upld_cmp_rel>
725
726 EVENT ANT_UPLOAD_FAILURE
727 FUNCTION PAD_clr_msg_num_in_stat_tbl
728 FUNCTION PAD_clr_sec_in_msg_info_tbl
729 FUNCTION PAD_clr_ready_to_play_bit
730 FUNCTION PAD_snd_das_mem_deall_sngl
731 /* uploader removes the disk file on an upload failure */
732 FUNCTION PAD_release_msg_num
733 FUNCTION PAD_snd_1gh_idle
734 TRANS <PAD_idle>
735
736 EVENT IDH_RC_DELETE_PA
737 FUNCTION PAD_snd_das_load_abort_upld
738 TRANS <PAD_rc_delete_pa> /* wait for abort response to delete */
739
740 EVENT ANY_STABLE_CALL
741 FUNCTION PAD_snd_tmw_any_unstable_ack
742 TRANS <PAD_wait_upld_cmp_rel>
743
744 EVENT ANY_FORCE_OOS /* don't send sdc failure, PAC is gone */
745 FUNCTION PAD_clr_msg_num_in_stat_tbl
746 FUNCTION PAD_snd_dasld_abrtupld_norsp
747 FUNCTION PAD_clr_sec_in_msg_info_tbl
748 FUNCTION PAD_clr_ready_to_play_bit
749 FUNCTION PAD_snd_das_mem_deall_sngl
```

10

```

757
758
759
760 /*****
761 *
762 * PAD_wait_upld_cmpl_nry
763 *
764 * Description:
765 *
766 * This state waits for an IDH_UPLOAD_COMPLETE
767 * to notify PAC. Agent signed out while upload was
768 * in progress and signed back in again while still
769 * in progress. PAC knows that the upload is in
770 * progress and wants to know when it can go on.
771 *
772 * Resources Active: none
773 *
774 * Critical Dependencies: none.
775 *
776 *
777 *
778 *
779 *
780 *
781 *
782 *
783 *
784 *
785 *
786 *
787 *
788 *
789 *
790 *
791 *
792 *
793 *
794 *
795 *
796 *
797 *
798 *
799 *
800 *
801 *
802 *
803 *
804 *
805 *
806 *
807 *
808 *
809 *
810 *
811 *
812 *
813 *
814 *
815 *
816 *
817 *
818 *
819 *
820 *
821 *
822 *
823 *
824 *
825 *
826 *
827 *
828 *
829 *
830 *
831 *
832 *
833 *
834 *
835 *
836 *
837 *
838 *
839 *
840 *
841 *
842 *
843 *
844 *
845 *
846 *
847 *
848 *
849 *
850 *
851 *
852 *
853 *
854 *
855 *
856 *
857 *
858 *
859 *
860 *
861 *
862 *
863 *
864 *
865 *
866 *
867 *
868 *
869 *
870 *
871 *
872 *
873 *
874 *
875 *
876 *
877 *
878 *
879 *
880 *
881 *
882 *
883 *
884 *
885 *
886 *
887 *
888 *
889 *
890 *
891 *
892 *
893 *
894 *
895 *
896 *
897 *
898 *
899 *
900 *
901 *
902 *
903 *
904 *
905 *
906 *
907 *
908 *
909 *
910 *
911 *
912 *
913 *
914 *
915 *
916 *
917 *
918 *
919 *
920 *
921 *
922 *
923 *
924 *
925 *
926 *
927 *
928 *
929 *
930 *
931 *
932 *
933 *
934 *
935 *
936 *
937 *
938 *
939 *
940 *
941 *
942 *
943 *
944 *
945 *
946 *
947 *
948 *
949 *
950 *
951 *
952 *
953 *
954 *
955 *
956 *
957 *
958 *
959 *
960 *
961 *
962 *
963 *
964 *
965 *
966 *
967 *
968 *
969 *
970 *
971 *
972 *
973 *
974 *
975 *
976 *
977 *
978 *
979 *
980 *
981 *
982 *
983 *
984 *
985 *
986 *
987 *
988 *
989 *
990 *
991 *
992 *
993 *
994 *
995 *
996 *
997 *
998 *
999 *
1000 *

```



```

822 /*****
823 *
824 *
825 *      PAD_rc_delete_pa
826 *
827 *      Description:          This state waits for an ANY_ABORT_MSG_ACK to
828 *                          delete a personal announcement message.
829 *
830 *      Resources Active: DAS memory and disk memory
831 *
832 *      Critical Dependencies: none
833 *
834 *****/
835
836
837
838 STATE PAD_rc_delete_pa
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884

```

92/11/09
12:16:07

1225 2667 STATE PAC_adort_record
350 954 STATE PAC_active
483 1287 STATE PAC_chk_delete_stat
88 322 STATE PAC_chk_status
265 759 STATE PAC_download
308 856 STATE PAC_download_option
666 1734 STATE PAC_aom_chk_fall_tag
619 1624 STATE PAC_aom_playing
76 277 STATE PAC_get_tag_id
12 89 STATE PAC_idle
52 214 STATE PAC_int_chk_status
20 120 STATE PAC_int_get_tag_id
122 384 STATE PAC_nc_pa
32 158 STATE PAC_nc_tag
2 56 STATE PAC_pos
413 1112 STATE PAC_play
506 1351 STATE PAC_play_option
1369 3548 STATE PAC_play_record
11 126 STATE PAC_post_aom_chk_tag
122 227 STATE PAC_post_aom_playing
328 2415 STATE PAC_pre_eip_chk_tag
388 2317 STATE PAC_pre_eip_playing
394 1056 STATE PAC_re_active_chk
684 1766 STATE PAC_re_allow_sys_no
101 1842 STATE PAC_record_option
281 2478 STATE PAC_recording
1007 3444 STATE PAC_sdc_upload
1288 3088 STATE PAC_sel_record_opt
136 1161 STATE PAC_select_option
561 1493 STATE PAC_start_aom
1072 2803 STATE PAC_start_post_aom
338 2196 STATE PAC_start_pre_eip
184 2053 STATE PAC_start_record
291 2577 STATE PAC_stop_record
1231 3192 STATE PAC_store_chk
1287 3344 STATE PAC_store_record
1255 3252 STATE PAC_store_wait
1422 3670 STATE PAC_wait_failure_ack
229 674 STATE PAC_wait_pa_ack
198 599 STATE PAC_wait_sdc_ack
162 510 STATE PAC_wait_sdc_release
129 431 STATE PAC_wait_tag_ack
546 1112 STATE PAC_wt_aom_tag
105 55 STATE PAC_wt_post_aom_tag
619 143 STATE PAC_wt_pre_eip_tag
756 1976 STATE PAC_wt_record_memory

2103204

/tmp/59203



2103204

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39

Design Notes of Interest:

Description: The PAC controls the playing, recording,
uploading and downloading of the personal announcements.
It interacts with the Protocol Handler (BPRH),
TAG LCH, and TAG LDH for recording and review of the
personal announcements.
It interacts with the Accept Application Call
Capability Handler (AAC), Position Call Handler (PCH),
BPRH, and TAG LDH for
the playback of the personal announcements.
It interacts with the Pseudo Port LCH (SDOG) and
Personal Announcements Pseudo Logical Device Handler (PAD)
for the uploading and downloading of the Personal
Announcement (PA).

40
41
42 /*****
43 *
44 * PAC_oos
45 *
46 * Description: The capability is in out of service state and not
47 * available for personal announcements message handling.
48 *
49 * Resources Active: none
50 *
51 * Critical Dependencies:
52 * none.
53 *
54 *****
55
56 STATE PAC_oos
57
58 EVENT ANY_SET_IN_SERVICE
59 /* Initialize read/write area */
60 FUNCTION PAC_init_rw
61 /* Populate PA information from Staff Table */
62 FUNCTION PAC_get_pa_info_from_staff
63 /* get the PA TAG group number is */
64 FUNCTION PAC_get_pa_tag_grp_id
65 TRANS <PAC_init_get_tag_id>
66
67 EVENT ANY_FORCE_OOS
68 TRANS <PAC_oos>
69
70 DONE PAC_oos.tde
71

2103204

```
72
73 /*****
74 *
75 *   PAC_idle
76 *
77 *   Description:   This state indicates the PAC is idle and waiting for
78 *                  the Agent IDH (ADH) to activate it.
79 *
80 *   Resources Active: none
81 *
82 *   Critical Dependencies:
83 *                       none.
84 *
85 *****/
86
87
88
89 STATE PAC_idle
90
91 EVENT PAC_activate /* activate event from ADH */
92 FUNCTION /* set the PA TAG group member id */
93 TRANS <PAC_init_get_tag_id>
94
95
96 EVENT ANY_FORCE_OOS
97 TRANS <PAC_oos>
98
99 DONE PAC_idle.tde
100
```

```
101
102
103
104 /*****
105 *
106 *   PAC_init_get_tag_id
107 *
108 *   Description:   This transition state is waiting for the
109 *                  Member ID of the TAG Group, dedicated for the
110 *                  Personal Announcements.
111 *
112 *   Resources Active: none
113 *
114 *   Critical Dependencies: none.
115 *
116 *****/
117
118
119
120 STATE PAC_init_get_tag_id
121
122 EVENT PAC_I_TAG_ID /* found valid PA TAG group member id */
123 FUNCTION /* Check current feature and messages status */
124 TRANS <PAC_init_chk_status>
125
126
127 EVENT PAC_I_NO_TAG_ID /* no PA TAG group member id defined */
128 FUNCTION /* set the feature status to "no tag defined" */
129 TRANS <PAC_oos>
130
131
132 TRANS <PAC_no_tag>
133
134 EVENT ANY_FORCE_OOS
135 TRANS <PAC_oos>
136
137 DONE PAC_init_get_tag_id.tde
138
139
```

2103204

```
139
140
141 /*****
142
143
144 * PAC_no_tag
145 *
146 * Description: This state indicates that personal
147 * announcements TAG group member id is not defined.
148 *
149 * Resources Active: none
150 *
151 * Critical Dependencies:
152 * none.
153 *
154 *****/
155
156 STATE PAC_no_tag
157
158 EVENT PAC_AAC_REQUEST_PA /* AAC's request for PA msg */
159
160 FUNCTION /* feature not ready yet; send NACK */
161 <PAC_no_tag> PAC_snd_aac_any_pa_request_nack
162
163 TRANS
164
165 EVENT PAC_PCH_REQUEST_PA /* PCH's request for PA msg */
166
167 FUNCTION /* feature not ready yet; send NACK */
168 <PAC_no_tag> PAC_snd_pch_any_pa_request_nack
169
170 EVENT PAC_SIGN_OUT /* agent signed out */
171
172 FUNCTION /* check if PC had changed state table after signed in */
173 <PAC_idle> PAC_chk_staff_chg_after_signin
174
175 EVENT PAC_PC_GROUP_UPDATE /* agent group assignment has changed */
176
177 FUNCTION /* check if PC had changed state table after signed in */
178 PAC_chk_staff_chg_after_signin
179
180 FUNCTION /* populate P/W area PA table from staff table */
181 PAC_get_pa_info_from_staff
182
183 FUNCTION /* get the PA TAG group member id */
184 <PAC_get_tag_id> PAC_get_pa_tag_grp_id
185
186 TRANS
187
188 EVENT ANY_FORCE_OOS
189
190 FUNCTION /* check if PC had changed state table after signed in */
191 PAC_chk_staff_chg_after_signin
192
193 TRANS
194
195 DONE PAC_no_tag.tde
196
```

```
191
192
193 /*****
194
195 * PAC_init_chk_status
196 *
197 * Description: This transition state is waiting for the result of
198 * the initial PA messages status check. The checks
199 * indicate whether the PA names have been defined,
200 * we need to wait for the previous upload to complete
201 * before proceeding, need to download messages,
202 * or just get TAG port because no messages have been
203 * recorded.
204 *
205 * Resources Active: none
206 *
207 * Critical Dependencies:
208 * This state should be reached only once, initially.
209 *
210 *****/
211
212 STATE PAC_init_chk_status
213
214 EVENT PAC_I_DOWNLOAD /* at least one PA message to be downloaded */
215
216 FUNCTION /* set feature status to CP_PA_FEAT_QUEUED */
217 PAC_set_feat_status_queued
218
219 FUNCTION /* send feature status to the BPRM menu handler */
220 PAC_snd_bprh_mh_pa_status
221
222 FUNCTION /* acquire PA pseudo IDH (PAD) */
223 PAC_snd_sdc_igh_acquire_pa
224
225 TRANS
226
227 EVENT PAC_I_DOWNLOAD_WAIT /* wait for uploading to complete */
228
229 FUNCTION /* set feature status to CP_PA_FEAT_UPLOADING */
230 PAC_set_feat_status_uploading
231
232 FUNCTION /* send feature status to the BPRM menu handler */
233 PAC_snd_bprh_mh_pa_status
234
235 FUNCTION /* tell PAD to notify when uploading is complete */
236 PAC_snd_pad_igh_notify_upload_complete
237
238 FUNCTION /* just wait for uploading to complete */
239 <PAC_wait_sdc_release>
240
241 TRANS
242
243 EVENT PAC_I_NOT_RECORDED /* no PA messages recorded */
244
245 FUNCTION /* set feature status to CP_PA_FEAT_QUEUED */
246 PAC_set_feat_status_queued
247
248 FUNCTION /* send feature status to the BPRM menu handler */
249 PAC_snd_bprh_mh_pa_status
250
251 FUNCTION /* get TAG port for PA */
252 <PAC_wait_tag_ack> PAC_snd_tag_igh_acquire
253
254 TRANS
255
256 EVENT PAC_I_NO_PA /* there are no PA names defined for agent grp */
257
258 FUNCTION /* set feature status to CP_PA_FEAT_NOT_PROVISIONED */
259 PAC_set_feat_status_not_provisioned
260
261 FUNCTION /* send feature status to the BPRM menu handler */
262 PAC_snd_bprh_mh_pa_status
263
264 TRANS
265
266 <PAC_no_pa>
267
268 EVENT ANY_FORCE_OOS
269
270 TRANS
271 <PAC_oos>
```

92/11/09
12:16:06

/tmp/59201

254
74 255 DONE PAC_init_chk_status.tde
75 256

```
257  
258  
259  
260  
261 /*****  
262 *  
263 * PAC_get_tag_id  
264 *  
265 * Description: This transition state is waiting for the  
266 * Member id of the TAG Group, dedicated for the  
267 * Personal Announcements.  
268 *  
269 * Resources Active: none  
270 *  
271 * Critical Dependencies: none.  
272 *  
273 *****/  
274  
275  
276  
277 STATE PAC_get_tag_id  
278  
279 EVENT PAC_I_TAG_ID /* found valid PA TAG group member id */  
280 /* Check current feature and messages status */  
281 FUNCTION PAC_chk_cur_pa_status  
282 <PAC_chk_status>  
283  
284 EVENT PAC_I_NO_TAG_ID /* no PA TAG group member id defined */  
285 /* set the feature status to "no tag defined" */  
286 FUNCTION PAC_set_feat_status_no_tag  
287 /* send feature status to the BPRH menu handler */  
288 FUNCTION PAC_snd_bprh_mh_pa_status  
289 <PAC_no_tag>  
290  
291 EVENT ANY_FORCE_OOS  
292 TRANS <PAC_oos>  
293  
294 DONE PAC_get_tag_id.tde  
295
```

2103204

296	359		
297	108	360	EVENT ANY_FORCE_OOS
298	109	361	TRANS <PAC_OOS>
299	110	363	
300	111	364	
301			PAC_chk_status.tde

PAC_chk_status

Description: This transition state is waiting for the result of the PA messages status check.

This state is similar to PAC_chk_init_status. The only difference is that it sends PA status to the BPRH Protocol Handler rather than BPRH Menu Handler. The first PA status event goes to menu handler, whereas all the subsequent ones go to protocol handler.

Resources Active: none

Critical Dependencies: none.

STATE PAC_chk_status

EVENT

PAC_I_DOWNLOAD /* at least one PA message to be downloaded */
FUNCTION /* set feature status to CP_PA_FEAT_QUEUED */
PAC_set_feat_status_queued

FUNCTION /* send feature status to the BPRH protocol handler */
PAC_snd_bprh_ph_pa_status

FUNCTION /* acquire PA pseudo LDH (PAD) */
PAC_snd_sdc_ldh_acquire_pa

TRANS

<PAC_wait_sdc_ack>

EVENT

PAC_I_DOWNLOAD_WAIT /* wait for uploading to complete */
FUNCTION /* set feature status to CP_PA_FEAT_UPLOADING */
PAC_set_feat_status_uploading

FUNCTION /* send feature status to the BPRH protocol handler */
PAC_snd_bprh_ph_pa_status

FUNCTION /* call PAD to notify when uploading is complete */
PAC_snd_pad_ldh_notify_upload_complete

TRANS

<PAC_wait_sdc_release>

EVENT

PAC_I_NOT_RECORDED /* no PA messages recorded */
FUNCTION /* set feature status to CP_PA_FEAT_QUEUED */
PAC_set_feat_status_queued

FUNCTION /* send feature status to the BPRH protocol handler */
PAC_snd_bprh_ph_pa_status

FUNCTION /* get TAG port for PA */
PAC_snd_tag_1gh_acquire

TRANS

<PAC_wait_tag_ack>

EVENT

PAC_I_NO_PA /* there are no PA names defined for agent grp */
FUNCTION /* set feature status to CP_PA_FEAT_NOT_PROVISIONED */
PAC_set_feat_status_not_provisioned

FUNCTION /* send feature status to the BPRH protocol handler */
PAC_snd_bprh_ph_pa_status

TRANS

<PAC_no_pa>

TRANS

<PAC_no_pa>

2103204

```
365
366
367 /*****
368 *
369 *   PAC_no_pa
370 *
371 *   Description:   This state indicates that there are no personal
372 *                  announcements provisioned for the agent's current
373 *                  primary agent group assignment.
374 *
375 *   Resources Active: none
376 *
377 *   Critical Dependencies:
378 *       none.
379 *
380 *****/
381
382
383 STATE PAC_no_pa
384
385
386 EVENT PAC_AAC_REQUEST_PA /* AAC's request for PA msg */
387     FUNCTION /* feature not ready yet; send NACK */
388         PAC_snd_aac_any_pa_request_nack
389
390 TRANS
391
392 EVENT PAC_PCH_REQUEST_PA /* PCH's request for PA msg */
393     FUNCTION /* feature not ready yet; send NACK */
394         PAC_snd_pch_any_pa_request_nack
395
396 TRANS
397
398 EVENT PAC_SIGN_OUT /* agent signed out */
399     FUNCTION /* no action needed */
400
401 TRANS
402
403 EVENT PAC_RC_GROUP_UPDATE /* agent group assignment has changed */
404     FUNCTION /* populate R/W area PA table from staff table */
405         PAC_get_pa_info_from_staff
406         /* get the PA TAG group member id */
407         PAC_get_pa_tag_grp_id
408
409 TRANS
410
411 EVENT ANY_FORCE_OOS
412     TRANS <PAC_oos>
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427 *****/
428
429
430 STATE PAC_wait_tag_ack
431
432
433
434
435
436
437
438 *   PAC_wait_tag_ack
439 *
440 *   Description:   This state is waiting for the TAG port.
441 *
442 *   Resources Active: none
443 *
444 *   Critical Dependencies:
445 *       none.
446
447 *****/
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
```

```
413
414
415
416 /*****
417 *
418 *   PAC_wait_tag_ack
419 *
420 *   Description:   This state is waiting for the TAG port.
421 *
422 *   Resources Active: none
423 *
424 *   Critical Dependencies:
425 *       none.
426 *
427 *****/
428
429
430 STATE PAC_wait_tag_ack
431
432
433
434
435
436
437
438 *   PAC_wait_tag_ack
439 *
440 *   Description:   This state is waiting for the TAG port.
441 *
442 *   Resources Active: none
443 *
444 *   Critical Dependencies:
445 *       none.
446
447 *****/
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
```

```
EVENT ANY_ACTIVATE_ACK /* we got TAG port */
    FUNCTION /* save TAG port id */
        PAC_save_tag_port_id
    FUNCTION /* get DAS unit # of the TAG port */
        PAC_get_das_unit_from_tag_id
    FUNCTION /* set feature status to active */
        PAC_set_feat_status_active
    FUNCTION /* send feature status to the BPPH protocol handler */
        PAC_snd_bpph_pa_pa_status
    TRANS <PAC_active>

EVENT PAC_AAC_REQUEST_PA /* AAC's request for PA msg */
    FUNCTION /* feature not ready yet; send NACK */
        PAC_snd_aac_any_pa_request_nack
    TRANS <PAC_wait_tag_ack>

EVENT PAC_PCH_REQUEST_PA /* PCH's request for PA msg */
    FUNCTION /* feature not ready yet; send NACK */
        PAC_snd_pch_any_pa_request_nack
    TRANS <PAC_wait_tag_ack>

EVENT PAC_RC_GROUP_UPDATE /* agent group assignment has changed */
    FUNCTION /* cancel TAG acquire request */
        PAC_snd_tag_lgh_cancel_acquire
    FUNCTION /* check if RC had changed Staff table after signed in */
        PAC_chk_staff_chg_after_signin
    FUNCTION /* populate R/W area PA table from staff table */
        PAC_get_pa_info_from_staff
    FUNCTION /* get the PA TAG group member id */
        PAC_get_pa_tag_grp_id
    TRANS <PAC_get_tag_id>

EVENT ANY_ACTIVATE_NACK /* no TAG port */
    FUNCTION /* set feature status to no TAG */
        PAC_set_feat_status_no_tag
    FUNCTION /* send feature status to the BPPH protocol handler */
```

```
154 476 FUNCTION PAC_snd_bprh_ph_pa_status
155 477 TRANS <PAC_nc_tag>
156 478
157 480 EVENT ANY_FORCE_OOS
158 481 FUNCTION /* cancel TAG acquire request */
159 482 /* PAC_snd_tag_igh_cancel acquire
160 483 /* check if RC had changed Staff table after signed in */
161 484 TRANS <PAC_pos>
162 485 PAC_chk_staff_chg_after_signin
163 486 DONE PAC_wait_tag_ack.cde
164 487
165 488

489
490
491
492 /*****
493 *
494 * PAC_wait_sdc_release
495 *
496 * Description: This state is waiting for the PA Pseudo LDH (PAD)
497 * to complete the upload initiated during the previous
498 * signin. The feature cannot proceed until the upload
499 * is complete.
500 *
501 * Resources Active: none
502 *
503 * Critical Dependencies:
504 * none.
505 *
506 *****/
507
508
509
510 STATE PAC_wait_sdc_release
511
512
513 EVENT PAC_AAC_REQUEST_PA /* AAC's request for PA msg */
514 FUNCTION /* feature not ready yet; send NACK */
515 TRANS <PAC_wait_sdc_release>
516
517 EVENT PAC_PCH_REQUEST_PA /* PCH's request for PA msg */
518 FUNCTION /* feature not ready yet; send NACK */
519 TRANS <PAC_wait_sdc_release>
520
521
522 EVENT PAC_UPLOAD_COMPLETE /* upload is complete */
523 FUNCTION /* check if another upload is in progress */
524 /* get the PA TAG group member id */
525 TRANS <PAC_get_tag_id>
526
527
528 EVENT PAC_SIGN_OUT /* agent signed out */
529 FUNCTION /* Inform PAD that PAC is going away */
530 TRANS <PAC_idle>
531
532
533 EVENT PAC_RC_GROUP_UPDATE /* agent group assignment has changed */
534 FUNCTION /* Inform PAD that PAC is going away */
535 TRANS <PAC_idle>
536
537
538 FUNCTION PAC_snd_pad_ldh_idle
539 /* check if RC had changed Staff table after signed in */
540 FUNCTION PAC_chk_staff_chg_after_signin
541 /* populate R/W area PA table from staff table */
542 FUNCTION PAC_get_pa_info_from_staff
543 /* get the PA TAG group member id */
544 TRANS <PAC_get_tag_id>
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
```

/tmp/59201

2103204

Part II

/tmp/59201

/tmp/59201

/tmp/59201

/tmp/59201

92/11/09
12:16:06

/tmp/59201

2103204

```
289 802 FUNCTION PAC_chk_staff_chg_after_signin 825
290 803 /* populate R/W area PA table from staff table */ 836
291 804 FUNCTION PAC_get_pa_infrc_from_staff 837
292 805 /* get the PA TAG group member id */ 838
293 806 FUNCTION PAC_get_pa_tag_grp_id 839
294 807 <PAC_get_tag_id> 840
295 808 841 * PAC_download_option
296 809 ANY_SDC_FAILURE /* PAD reported failure */ 842
297 810 /* idle the TAG port */ 843
298 811 FUNCTION PAC_snd_tag_ldh_idle 844
299 812 /* get the PA TAG group member id */ 845
300 813 FUNCTION PAC_get_pa_tag_grp_id 846
301 814 <PAC_get_tag_id> 847
302 815 848 * Resources Active: TAG port and PAD port.
303 816 ANY_MSG_FAILURE /* TAG reported failure */ 849
304 817 /* idle the PAD */ 850
305 818 FUNCTION PAC_snd_pad_ldh_idle 851
306 819 /* get the PA TAG group member id */ 852
307 820 FUNCTION PAC_get_pa_tag_grp_id 853
308 821 <PAC_get_tag_id> 854
309 822 855
310 823 ANY_FORCE_OOS 856
311 824 /* idle the PAD */ 857
312 825 FUNCTION PAC_snd_pad_ldh_idle 858
313 826 /* idle the TAG port */ 859
314 827 FUNCTION PAC_snd_tag_ldh_idle 860
315 828 /* check if PC has changed Staff table after signed in */ 861
316 829 FUNCTION PAC_chk_staff_chg_after_signin 862
317 830 <PAC_003> 863
318 831 864
319 832 DONE 865
320 833 PAC_download.cde 866
321 834 867
322 835 868
323 836 869
324 837 870
325 838 871
326 839 872
327 840 873
328 841 874
329 842 875
330 843 876
331 844 877
332 845 878
333 846 879
334 847 880
335 848 881
336 849 882
337 850 883
338 851 884
339 852 885
340 853 886
341 854 887
342 855 888
343 856 889
344 857 890
345 858 891
346 859 892
347 860 893
348 861 894
349 862 895
350 863 896
351 864 897
352 865 898
353 866 899
354 867 900
355 868 901
356 869 902
357 870 903
358 871 904
359 872 905
360 873 906
361 874 907
362 875 908
363 876 909
364 877 910
365 878 911
366 879 912
367 880 913
368 881 914
369 882 915
370 883 916
371 884 917
372 885 918
373 886 919
374 887 920
375 888 921
376 889 922
377 890 923
378 891 924
379 892 925
380 893 926
381 894 927
382 895 928
383 896 929
384 897 930
385 898 931
386 899 932
387 900 933
388 901 934
389 902 935
390 903 936
391 904 937
392 905 938
393 906 939
394 907 940
395 908 941
396 909 942
397 910 943
398 911 944
399 912 945
400 913 946
401 914 947
402 915 948
403 916 949
404 917 950
405 918 951
406 919 952
407 920 953
408 921 954
409 922 955
410 923 956
411 924 957
412 925 958
413 926 959
414 927 960
415 928 961
416 929 962
417 930 963
418 931 964
419 932 965
420 933 966
421 934 967
422 935 968
423 936 969
424 937 970
425 938 971
426 939 972
427 940 973
428 941 974
429 942 975
430 943 976
431 944 977
432 945 978
433 946 979
434 947 980
435 948 981
436 949 982
437 950 983
438 951 984
439 952 985
440 953 986
441 954 987
442 955 988
443 956 989
444 957 990
445 958 991
446 959 992
447 960 993
448 961 994
449 962 995
450 963 996
451 964 997
452 965 998
453 966 999
454 967 1000
455 968 1001
456 969 1002
457 970 1003
458 971 1004
459 972 1005
460 973 1006
461 974 1007
462 975 1008
463 976 1009
464 977 1010
465 978 1011
466 979 1012
467 980 1013
468 981 1014
469 982 1015
470 983 1016
471 984 1017
472 985 1018
473 986 1019
474 987 1020
475 988 1021
476 989 1022
477 990 1023
478 991 1024
479 992 1025
480 993 1026
481 994 1027
482 995 1028
483 996 1029
484 997 1030
485 998 1031
486 999 1032
487 1000 1033
488 1001 1034
489 1002 1035
490 1003 1036
491 1004 1037
492 1005 1038
493 1006 1039
494 1007 1040
495 1008 1041
496 1009 1042
497 1010 1043
498 1011 1044
499 1012 1045
500 1013 1046
501 1014 1047
502 1015 1048
503 1016 1049
504 1017 1050
505 1018 1051
506 1019 1052
507 1020 1053
508 1021 1054
509 1022 1055
510 1023 1056
511 1024 1057
512 1025 1058
513 1026 1059
514 1027 1060
515 1028 1061
516 1029 1062
517 1030 1063
518 1031 1064
519 1032 1065
520 1033 1066
521 1034 1067
522 1035 1068
523 1036 1069
524 1037 1070
525 1038 1071
526 1039 1072
527 1040 1073
528 1041 1074
529 1042 1075
530 1043 1076
531 1044 1077
532 1045 1078
533 1046 1079
534 1047 1080
535 1048 1081
536 1049 1082
537 1050 1083
538 1051 1084
539 1052 1085
540 1053 1086
541 1054 1087
542 1055 1088
543 1056 1089
544 1057 1090
545 1058 1091
546 1059 1092
547 1060 1093
548 1061 1094
549 1062 1095
550 1063 1096
551 1064 1097
552 1065 1098
553 1066 1099
554 1067 1100
555 1068 1101
556 1069 1102
557 1070 1103
558 1071 1104
559 1072 1105
560 1073 1106
561 1074 1107
562 1075 1108
563 1076 1109
564 1077 1110
565 1078 1111
566 1079 1112
567 1080 1113
568 1081 1114
569 1082 1115
570 1083 1116
571 1084 1117
572 1085 1118
573 1086 1119
574 1087 1120
575 1088 1121
576 1089 1122
577 1090 1123
578 1091 1124
579 1092 1125
580 1093 1126
581 1094 1127
582 1095 1128
583 1096 1129
584 1097 1130
585 1098 1131
586 1099 1132
587 1100 1133
588 1101 1134
589 1102 1135
590 1103 1136
591 1104 1137
592 1105 1138
593 1106 1139
594 1107 1140
595 1108 1141
596 1109 1142
597 1110 1143
598 1111 1144
599 1112 1145
600 1113 1146
601 1114 1147
602 1115 1148
603 1116 1149
604 1117 1150
605 1118 1151
606 1119 1152
607 1120 1153
608 1121 1154
609 1122 1155
610 1123 1156
611 1124 1157
612 1125 1158
613 1126 1159
614 1127 1160
615 1128 1161
616 1129 1162
617 1130 1163
618 1131 1164
619 1132 1165
620 1133 1166
621 1134 1167
622 1135 1168
623 1136 1169
624 1137 1170
625 1138 1171
626 1139 1172
627 1140 1173
628 1141 1174
629 1142 1175
630 1143 1176
631 1144 1177
632 1145 1178
633 1146 1179
634 1147 1180
635 1148 1181
636 1149 1182
637 1150 1183
638 1151 1184
639 1152 1185
640 1153 1186
641 1154 1187
642 1155 1188
643 1156 1189
644 1157 1190
645 1158 1191
646 1159 1192
647 1160 1193
648 1161 1194
649 1162 1195
650 1163 1196
651 1164 1197
652 1165 1198
653 1166 1199
654 1167 1200
655 1168 1201
656 1169 1202
657 1170 1203
658 1171 1204
659 1172 1205
660 1173 1206
661 1174 1207
662 1175 1208
663 1176 1209
664 1177 1210
665 1178 1211
666 1179 1212
667 1180 1213
668 1181 1214
669 1182 1215
670 1183 1216
671 1184 1217
672 1185 1218
673 1186 1219
674 1187 1220
675 1188 1221
676 1189 1222
677 1190 1223
678 1191 1224
679 1192 1225
680 1193 1226
681 1194 1227
682 1195 1228
683 1196 1229
684 1197 1230
685 1198 1231
686 1199 1232
687 1200 1233
688 1201 1234
689 1202 1235
690 1203 1236
691 1204 1237
692 1205 1238
693 1206 1239
694 1207 1240
695 1208 1241
696 1209 1242
697 1210 1243
698 1211 1244
699 1212 1245
700 1213 1246
701 1214 1247
702 1215 1248
703 1216 1249
704 1217 1250
705 1218 1251
706 1219 1252
707 1220 1253
708 1221 1254
709 1222 1255
710 1223 1256
711 1224 1257
712 1225 1258
713 1226 1259
714 1227 1260
715 1228 1261
716 1229 1262
717 1230 1263
718 1231 1264
719 1232 1265
720 1233 1266
721 1234 1267
722 1235 1268
723 1236 1269
724 1237 1270
725 1238 1271
726 1239 1272
727 1240 1273
728 1241 1274
729 1242 1275
730 1243 1276
731 1244 1277
732 1245 1278
733 1246 1279
734 1247 1280
735 1248 1281
736 1249 1282
737 1250 1283
738 1251 1284
739 1252 1285
740 1253 1286
741 1254 1287
742 1255 1288
743 1256 1289
744 1257 1290
745 1258 1291
746 1259 1292
747 1260 1293
748 1261 1294
749 1262 1295
750 1263 1296
751 1264 1297
752 1265 1298
753 1266 1299
754 1267 1300
755 1268 1301
756 1269 1302
757 1270 1303
758 1271 1304
759 1272 1305
760 1273 1306
761 1274 1307
762 1275 1308
763 1276 1309
764 1277 1310
765 1278 1311
766 1279 1312
767 1280 1313
768 1281 1314
769 1282 1315
770 1283 1316
771 1284 1317
772 1285 1318
773 1286 1319
774 1287 1320
775 1288 1321
776 1289 1322
777 1290 1323
778 1291 1324
779 1292 1325
780 1293 1326
781 1294 1327
782 1295 1328
783 1296 1329
784 1297 1330
785 1298 1331
786 1299 1332
787 1300 1333
788 1301 1334
789 1302 1335
790 1303 1336
791 1304 1337
792 1305 1338
793 1306 1339
794 1307 1340
795 1308 1341
796 1309 1342
797 1310 1343
798 1311 1344
799 1312 1345
800 1313 1346
801 1314 1347
802 1315 1348
803 1316 1349
804 1317 1350
805 1318 1351
806 1319 1352
807 1320 1353
808 1321 1354
809 1322 1355
810 1323 1356
811 1324 1357
812 1325 1358
813 1326 1359
814 1327 1360
815 1328 1361
816 1329 1362
817 1330 1363
818 1331 1364
819 1332 1365
820 1333 1366
821 1334 1367
822 1335 1368
823 1336 1369
824 1337 1370
825 1338 1371
826 1339 1372
827 1340 1373
828 1341 1374
829 1342 1375
830 1343 1376
831 1344 1377
832 1345 1378
833 1346 1379
834 1347 1380
835 1348 1381
836 1349 1382
837 1350 1383
838 1351 1384
839 1352 1385
840 1353 1386
841 1354 1387
842 1355 1388
843 1356 1389
844 1357 1390
845 1358 1391
846 1359 1392
847 1360 1393
848 1361 1394
849 1362 1395
850 1363 1396
851 1364 1397
852 1365 1398
853 1366 1399
854 1367 1400
855 1368 1401
856 1369 1402
857 1370 1403
858 1371 1404
859 1372 1405
860 1373 1406
861 1374 1407
862 1375 1408
863 1376 1409
864 1377 1410
865 1378 1411
866 1379 1412
867 1380 1413
868 1381 1414
869 1382 1415
870 1383 1416
871 1384 1417
872 1385 1418
873 1386 1419
874 1387 1420
875 1388 1421
876 1389 1422
877 1390 1423
878 1391 1424
879 1392 1425
880 1393 1426
881 1394 1427
882 1395 1428
883 1396 1429
884 1397 1430
885 1398 1431
886 1399 1432
887 1400 1433
888 1401 1434
889 1402 1435
890 1403 1436
891 1404 1437
892 1405 1438
893 1406 1439
894 1407 1440
895 1408 1441
896 1409 1442
897 1410 1443
898 1411 1444
899 1412 1445
900 1413 1446
901 1414 1447
902 1415 1448
903 1416 1449
904 1417 1450
905 1418 1451
906 1419 1452
907 1420 1453
908 1421 1454
909 1422 1455
910 1423 1456
911 1424 1457
912 1425 1458
913 1426 1459
914 1427 1460
915 1428 1461
916 1429 1462
917 1430 1463
918 1431 1464
919 1432 1465
920 1433 1466
921 1434 1467
922 1435 1468
923 1436 1469
924 1437 1470
925 1438 1471
926 1439 1472
927 1440 1473
928 1441 1474
929 1442 1475
930 1443 1476
931 1444 1477
932 1445 1478
933 1446 1479
934 1447 1480
935 1448 1481
936 1449 1482
937 1450 1483
938 1451 1484
939 1452 1485
940 1453 1486
941 1454 1487
942 1455 1488
943 1456 1489
944 1457 1490
945 1458 1491
946 1459 1492
947 1460 1493
948 1461 1494
949 1462 1495
950 1463 1496
951 1464 1497
952 1465 1498
953 1466 1499
954 1467 1500
955 1468 1501
956 1469 1502
957 1470 1503
958 1471 1504
959 1472 1505
960 1473 1506
961 1474 1507
962 1475 1508
963 1476 1509
964 1477 1510
965 1478 1511
966 1479 1512
967 1480 1513
968 1481 1514
969 1482 1515
970 1483 1516
971 1484 1517
972 1485 1518
973 1486 1519
974 1487 1520
975 1488 1521
976 1489 1522
977 1490 1523
978 1491 1524
979 1492 1525
980 1493 1526
981 1494 1527
982 1495 1528
983 1496 1529
984 1497 1530
985 1498 1531
986 1499 1532
987 1500 1533
988 1501 1534
989 1502 1535
990 1503 1536
991 1504 1537
992 1505 1538
993 1506 1539
994 1507 1540
995 1508 1541
996 1509 1542
997 1510 1543
998 1511 1544
999 1512 1545
1000 1513 1546
1001 1514 1547
1002 1515 1548
1003 1516 1549
1004 1517 1550
1005 1518 1551
1006 1519 1552
1007 1520 1553
1008 1521 1554
1009 1522 1555
1010 1523 1556
1011 1524 1557
1012 1525 1558
1013 1526 1559
1014 1527 1560
1015 1528 1561
1016 1529 1562
1017 1530 1563
1018 1531 1564
1019 1532 1565
1020 1533 1566
1021 1534 1567
1022 1535 1568
1023 1536 1569
1024 1537 1570
1025 1538 1571
1026 1539 1572
1027 1540 1573
1028 1541 1574
1029 1542 1575
1030 1543 1576
1031 1544 1577
1032 1545 1578
1033 1546 1579
1034 1547 1580
1035 1548 1581
1036 1549 1582
1037 1550 1583
1038 1551 1584
1039 1552 1585
1040 1553 1586
1041 1554 1587
1042 1555 1588
1043 1556 1589
1044 1557 1590
1045 1558 1591
1046 1559 1592
1047 1560 1593
1048 1561 1594
1049 1562 1595
1050 1563 1596
1051 1564 1597
1052 1565 1598
1053 1566 1599
1054 1567 1600
1055 1568 1601
1056 1569 1602
1057 1570 1603
1058 1571 1604
1059 1572 1605
1060 1573 1606
1061 1574 1607
1062 1575 1608
1063 1576 1609
1064 1577 1610
1065 1578 1611
1066 1579 1612
1067 1580 1613
1068 1581 1614
1069 1582 1615
1070 1583 1616
1071 1584 1617
1072 1585 1618
1073 1586 1619
1074 1587 1620
1075 1588 1621
1076 1589 1622
1077 1590 1623
1078 1591 1624
1079 1592 1625
1080 1593 1626
1081 1594 1627
1082 1595 1628
1083 1596 1629
1084 1597 1630
1085 1598 1631
1086 1599 1632
1087 1600 1633
1088 1601 1634
1089 1602 1635
1090 1603 1636
1091 1604 1637
1092 1605 1638
1093 1606 1639
1094 1607 1640
1095 1608 1641
1096 1609 1642
1097 1610 1643
1098 1611 1644
1099 1612 1645
1100 1613 1646
1101 1614 1647
1102 1615 1648
1103 1616 1649
1104 1617 1650
1105 1618 1651
1106 1619 1652
1107 1620 1653
1108 1621 1654
1109 1622 1655
1110 1623 1656
1111 1624 1657
1112 1625 1658
1113 1626 1659
1114 1627 1660
1115 1628 1661
1116 1629 1662
1117 1630 1663
1118 1631 1664
1119 1632 1665
1120 1633 1666
1121 1634 1667
1122 1635 1668
1123 1636 1669
1124 1637 1670
1125 1638 1671
1126 1639 1672
1127 1640 1673
1128 1641 1674
1129 1642 1675
1130 1643 1676
1131 1644 1677
1132 1645 1678
1133 1646 1679
1134 1647 1680
1135 1648 1681
1136 1649 1682
1137 1650 1683
1138 1651 1684
1139 1652 1685
1140 1653 1686
1141 1654 1687
1142 1655 1688
1143 1656 1689
1144 1657 1690
1145 1658 1691
1146 1659 1692
1147 1660 1693
1148 1661 1694
1149 1662 1695
1150 1663 1696
1151 1664 1697
1152 1665 1698
1153 1666 1699
1154 1667 1700
1155 1668 1701
1156 1669 1702
1157 1670 1703
1158 1671 1704
1159 1672 1705
1160 1673 1706
1
```

12

THE

2103204

92/11/09
12:16:06

/tmp/59201

13

2103204

```
372 994 EVENT PAC_RC_GRP_UPDATE /* agent group assignment has changed */
373 995 /* check if any PA msgs are different for new agent grp
374 996 FUNCTION PAC_comp_pa_tbl_with_staff
375 997 <PAC_rc_active_chk>
376 998
377 999
378 1000 ANY_SDC_FAILURE /* Uploading PAD reported failure */
379 1001 /* clear the sys msg # from PA tbl associated with PAD */
380 1002 FUNCTION PAC_clear_upload_fail_pad_sys_msg_num
381 1003 <PAC_active>
382 1004
383 1005 ANY_MSG_FAILURE /* TAG reported failure */
384 1006 /* if there is a PAD currently uploading a message, */
385 1007 /* tell it that PAC is going away */
386 1008 FUNCTION PAC_snd_pad_ldh_idle_if_upload
387 1009 /* get the PA TAG group member id */
388 1010 FUNCTION PAC_get_pa_tag_grp_id
389 1011 <PAC_get_tag_id>
390 1012
391 1013 ANY_FORCE_OOS
392 1014 /* if there is a PAD currently uploading a message, */
393 1015 /* tell it that PAC is going away */
394 1016 FUNCTION PAC_snd_pad_ldh_idle_if_upload
395 1017 /* idle the TAG port */
396 1018 FUNCTION PAC_snd_tag_ldh_idle
397 1019 /* deallocate DAS memory for PA messages */
398 1020 FUNCTION PAC_dealloc_msgs_on_signout
399 1021 /* check if RC had changed Staff table after signed in */
400 1022 FUNCTION PAC_chk_staff_chg_after_signin
401 1023 <PAC_oos>
402 1024
403 1025 PAC_STOP_PA /* can get because of race condition */
404 1026 <PAC_active>
405 1027
406 1028 /* we need to handle this, because we transition here from PAC_recording
407 1029 * state, on getting ANY_SDC_FAILURE */
408 1030 EVENT ANY_STOP_RECORD_ACK
409 1031 TRANS <PAC_active>
410 1032
411 1033 PAC_active.tde
412 1034
1035
1036
1037
1038
1039 /*****
1040 *
1041 * PAC_rc_active_chk
1042 *
1043 * Description: This state is waiting for the check to see if any
1044 * PA messages are different for the new agent group.
1045 *
1046 * Resources Active: TAG port
1047 * Pseudo port (PAD) may be active, for uploading.
1048 *
1049 * Critical Dependencies:
1050 * none.
1051 *
1052 *****/
1053
1054
1055
394 1056 STATE PAC_rc_active_chk
395 1057
396 1058 EVENT PAC_IN_CHANGE /* no PA msgs changed for new agent group */
397 1059 /* no action: remain active */
398 1060 TRANS <PAC_active>
399 1061
400 1062
401 1063
402 1064
403 1065
404 1066
405 1067
406 1068
407 1069
408 1070
409 1071
410 1072
411 1073
412 1074
413 1075
414 1076
415 1077
416 1078
417 1079
418 1080
419 1081
420 1082
421 1083
422 1084
423 1085
424 1086
425 1087
426 1088
427 1089
428 1090
429 1091
430 1092
431 1093
432 1094
433 1095
434 1096
435 1097
436 1098
437 1099
438 1100
439 1101
440 1102
441 1103
442 1104
443 1105
444 1106
445 1107
446 1108
447 1109
448 1110
449 1111
450 1112
451 1113
452 1114
453 1115
454 1116
455 1117
456 1118
457 1119
458 1120
459 1121
460 1122
461 1123
462 1124
463 1125
464 1126
465 1127
466 1128
467 1129
468 1130
469 1131
470 1132
471 1133
472 1134
473 1135
474 1136
475 1137
476 1138
477 1139
478 1140
479 1141
480 1142
481 1143
482 1144
483 1145
484 1146
485 1147
486 1148
487 1149
488 1150
489 1151
490 1152
491 1153
492 1154
493 1155
494 1156
495 1157
496 1158
497 1159
498 1160
499 1161
500 1162
501 1163
502 1164
503 1165
504 1166
505 1167
506 1168
507 1169
508 1170
509 1171
510 1172
511 1173
512 1174
513 1175
514 1176
515 1177
516 1178
517 1179
518 1180
519 1181
520 1182
521 1183
522 1184
523 1185
524 1186
525 1187
526 1188
527 1189
528 1190
529 1191
530 1192
531 1193
532 1194
533 1195
534 1196
535 1197
536 1198
537 1199
538 1200
539 1201
540 1202
541 1203
542 1204
543 1205
544 1206
545 1207
546 1208
547 1209
548 1210
549 1211
550 1212
551 1213
552 1214
553 1215
554 1216
555 1217
556 1218
557 1219
558 1220
559 1221
560 1222
561 1223
562 1224
563 1225
564 1226
565 1227
566 1228
567 1229
568 1230
569 1231
570 1232
571 1233
572 1234
573 1235
574 1236
575 1237
576 1238
577 1239
578 1240
579 1241
580 1242
581 1243
582 1244
583 1245
584 1246
585 1247
586 1248
587 1249
588 1250
589 1251
590 1252
591 1253
592 1254
593 1255
594 1256
595 1257
596 1258
597 1259
598 1260
599 1261
600 1262
601 1263
602 1264
603 1265
604 1266
605 1267
606 1268
607 1269
608 1270
609 1271
610 1272
611 1273
612 1274
613 1275
614 1276
615 1277
616 1278
617 1279
618 1280
619 1281
620 1282
621 1283
622 1284
623 1285
624 1286
625 1287
626 1288
627 1289
628 1290
629 1291
630 1292
631 1293
632 1294
633 1295
634 1296
635 1297
636 1298
637 1299
638 1300
639 1301
640 1302
641 1303
642 1304
643 1305
644 1306
645 1307
646 1308
647 1309
648 1310
649 1311
650 1312
651 1313
652 1314
653 1315
654 1316
655 1317
656 1318
657 1319
658 1320
659 1321
660 1322
661 1323
662 1324
663 1325
664 1326
665 1327
666 1328
667 1329
668 1330
669 1331
670 1332
671 1333
672 1334
673 1335
674 1336
675 1337
676 1338
677 1339
678 1340
679 1341
680 1342
681 1343
682 1344
683 1345
684 1346
685 1347
686 1348
687 1349
688 1350
689 1351
690 1352
691 1353
692 1354
693 1355
694 1356
695 1357
696 1358
697 1359
698 1360
699 1361
700 1362
701 1363
702 1364
703 1365
704 1366
705 1367
706 1368
707 1369
708 1370
709 1371
710 1372
711 1373
712 1374
713 1375
714 1376
715 1377
716 1378
717 1379
718 1380
719 1381
720 1382
721 1383
722 1384
723 1385
724 1386
725 1387
726 1388
727 1389
728 1390
729 1391
730 1392
731 1393
732 1394
733 1395
734 1396
735 1397
736 1398
737 1399
738 1400
739 1401
740 1402
741 1403
742 1404
743 1405
744 1406
745 1407
746 1408
747 1409
748 1410
749 1411
750 1412
751 1413
752 1414
753 1415
754 1416
755 1417
756 1418
757 1419
758 1420
759 1421
760 1422
761 1423
762 1424
763 1425
764 1426
765 1427
766 1428
767 1429
768 1430
769 1431
770 1432
771 1433
772 1434
773 1435
774 1436
775 1437
776 1438
777 1439
778 1440
779 1441
780 1442
781 1443
782 1444
783 1445
784 1446
785 1447
786 1448
787 1449
788 1450
789 1451
790 1452
791 1453
792 1454
793 1455
794 1456
795 1457
796 1458
797 1459
798 1460
799 1461
800 1462
801 1463
802 1464
803 1465
804 1466
805 1467
806 1468
807 1469
808 1470
809 1471
810 1472
811 1473
812 1474
813 1475
814 1476
815 1477
816 1478
817 1479
818 1480
819 1481
820 1482
821 1483
822 1484
823 1485
824 1486
825 1487
826 1488
827 1489
828 1490
829 1491
830 1492
831 1493
832 1494
833 1495
834 1496
835 1497
836 1498
837 1499
838 1500
839 1501
840 1502
841 1503
842 1504
843 1505
844 1506
845 1507
846 1508
847 1509
848 1510
849 1511
850 1512
851 1513
852 1514
853 1515
854 1516
855 1517
856 1518
857 1519
858 1520
859 1521
860 1522
861 1523
862 1524
863 1525
864 1526
865 1527
866 1528
867 1529
868 1530
869 1531
870 1532
871 1533
872 1534
873 1535
874 1536
875 1537
876 1538
877 1539
878 1540
879 1541
880 1542
881 1543
882 1544
883 1545
884 1546
885 1547
886 1548
887 1549
888 1550
889 1551
890 1552
891 1553
892 1554
893 1555
894 1556
895 1557
896 1558
897 1559
898 1560
899 1561
900 1562
901 1563
902 1564
903 1565
904 1566
905 1567
906 1568
907 1569
908 1570
909 1571
910 1572
911 1573
912 1574
913 1575
914 1576
915 1577
916 1578
917 1579
918 1580
919 1581
920 1582
921 1583
922 1584
923 1585
924 1586
925 1587
926 1588
927 1589
928 1590
929 1591
930 1592
931 1593
932 1594
933 1595
934 1596
935 1597
936 1598
937 1599
938 1600
939 1601
940 1602
941 1603
942 1604
943 1605
944 1606
945 1607
946 1608
947 1609
948 1610
949 1611
950 1612
951 1613
952 1614
953 1615
954 1616
955 1617
956 1618
957 1619
958 1620
959 1621
960 1622
961 1623
962 1624
963 1625
964 1626
965 1627
966 1628
967 1629
968 1630
969 1631
970 1632
971 1633
972 1634
973 1635
974 1636
975 1637
976 1638
977 1639
978 1640
979 1641
980 1642
981 1643
982 1644
983 1645
984 1646
985 1647
986 1648
987 1649
988 1650
989 1651
990 1652
991 1653
992 1654
993 1655
994 1656
995 1657
996 1658
997 1659
998 1660
999 1661
1000 1662
1001 1663
1002 1664
1003 1665
1004 1666
1005 1667
1006 1668
1007 1669
1008 1670
1009 1671
1010 1672
1011 1673
1012 1674
1013 1675
1014 1676
1015 1677
1016 1678
1017 1679
1018 1680
1019 1681
1020 1682
1021 1683
1022 1684
1023 1685
1024 1686
1025 1687
1026 1688
1027 1689
1028 1690
1029 1691
1030 1692
1031 1693
1032 1694
1033 1695
1034 1696
1035 1697
1036 1698
1037 1699
1038 1700
1039 1701
1040 1702
1041 1703
1042 1704
1043 1705
1044 1706
1045 1707
1046 1708
1047 1709
1048 1710
1049 1711
1050 1712
1051 1713
1052 1714
1053 1715
1054 1716
1055 1717
1056 1718
1057 1719
1058 1720
1059 1721
1060 1722
1061 1723
1062 1724
1063 1725
1064 1726
1065 1727
1066 1728
1067 1729
1068 1730
1069 1731
1070 1732
1071 1733
1072 1734
1073 1735
1074 1736
1075 1737
1076 1738
1077 1739
1078 1740
1079 1741
1080 1742
1081 1743
1082 1744
1083 1745
1084 1746
1085 1747
1086 1748
1087 1749
1088 1750
1089 1751
1090 1752
1091 1753
1092 1754
1093 1755
1094 1756
1095 1757
1096 1758
1097 1759
1098 1760
1099 1761
1100 1762
1101 1763
1102 1764
1103 1765
1104 1766
1105 1767
1106 1768
1107 1769
1108 1770
1109 1771
1110 1772
1111 1773
1112 1774
1113 1775
1114 1776
1115 1777
1116 1778
1117 1779
1118 1780
1119 1781
1120 1782
1121 1783
1122 1784
1123 1785
1124 1786
1125 1787
1126 1788
1127 1789
1128 1790
1129 1791
1130 1792
1131 1793
1132 1794
1133 1795
1134 1796
1135 1797
1136 1798
1137 1799
1138 1800
1139 1801
1140 1802
1141 1803
1142 1804
1143 1805
1144 1806
1145 1807
1146 1808
1147 1809
1148 1810
1149 1811
1150 1812
1151 1813
1152 1814
1153 1815
1154 1816
1155 1817
1156 1818
1157 1819
1158 1820
1159 1821
1160 1822
1161 1823
1162 1824
1163 1825
1164 1826
1165 1827
1166 1828
1167 1829
1168 1830
1169 1831
1170 1832
1171 1833
1172 1834
1173 1835
1174 1836
1175 1837
1176 1838
1177 1839
1178 1840
1179 1841
1180 1842
1181 1843
1182 1844
1183 1845
1184 1846
1185 1847
1186 1848
1187 1849
1188 1850
1189 1851
1190 1852
1191 1853
1192 1854
1193 1855
1194 1856
1195 1857
1196 1858
1197 1859
1198 1860
1199 1861
1200 1862
1201 1863
1202 1864
1203 1865
1204 1866
1205 1867
1206 1868
1207 1869
1208 1870
1209 1871
1210 1872
1211 1873
1212 1874
1213 1875
1214 1876
1215 1877
1216 1878
1217 1879
1218 1880
1219 1881
1220 1882
1221 1883
1222 1884
1223 1885
1224 1886
1225 1887
1226 1888
1227 1889
1228 1890
1229 1891
1230 1892
1231 1893
1232 1894
1233 1895
1234 1896
1235 1897
1236 1898
1237 1899
1238 1900
1239 1901
1240 1902
1241 1903
1242 1904
1243 1905
1244 1906
1245 1907
1246 1908
1247 1909
1248 1910
1249 1911
1250 1912
1251 1913
1252 1914
1253 1915
1254 1916
1255 1917
1256 1918
1257 1919
1258 1920
1259 1921
1260 1922
1261 1923
1262 1924
1263 1925
1264 1926
1265 1927
1266 1928
1267 1929
1268 1930
1269 1931
1270 1932
1271 1933
1272 1934
1273 1935
1274 1936
1275 1937
1276 1938
1277 1939
1278 1940
1279 1941
1280 1942
1281 1943
1282 1944
1283 1945
1284 1946
1285 1947
1286 1948
1287 1949
1288 1950
1289 1951
1290 1952
1291 1953
1292 1954
1293 1955
1294 1956
1295 1957
1296 1958
1297 1959
1298 1960
1299 1961
1300 1962
1301 1963
1302 1964
1303 1965
1304 1966
1305 1967
1306 1968
1307 1969
1308 1970
1309 1971
1310 1972
1311 1973
1312 1974
1313 1975
1314 1976
1315 1977
1316 1978
1317 1979
1318 1980
1319 1981
1320 1982
1321 1983
1322 1984
1323 1985
1324 1986
1325 1987
1326 1988
1327 1989
1328 1990
1329 1991
1330 1992
1331 1993
1332 1994
1333 1995
1334 1996
1335 1997
1336 1998
1337 1999
1338 2000
1339 2001
1340 2002
1341 2003
1342 2004
1343 2005
1344 2006
1345 2007
1346 2008
1347 2009
1348 2010
1349 2011
1350 2012
1351 2013
1352 2014
1353 2015
1354 2016
1355 2017
1356 2018
1357 2019
1358 2020
1359 2021
1360 2022
1361 2023
1362 2024
1363 2025
1364 2026
1365 2027
1366 2028
1367 2029
1368 2030
1369 2031
1370 2032
1371 2033
1372 2034
1373 2035
1374 2036
1375 2037
1376 2038
1377 2039
1378 2040
1379 2041
1380 2042
1381 2043
1382 2044
1383 2045
1384 2046
1385 2047
1386 2048
1387 2049
1388 2050
1389 2051
1390 2052
1391 2053
1392 2054
1393 2055
1394 2056
1395 2057
1396 2058
1397 2059
1398 2060
1399 2061
1400 2062
1401 2063
1402 2064
1403 2065
1404 2066
1405 2067
1406 2068
1407 2069
1408 2070
1409 2071
1410 2072
1411 2073
1412 2074
1413 2075
1414 2076
1415 2077
1416 2078
1417 2079
1418 2080
1419 2081
1420 2082
1421 2083
1422 2084
1423 2085
1424 2086
1425 2087
1426 2088
1427 2089
1428 2090
1429 2091
1430 2092
1431 2093
1432 2094
1433 2095
1434 2096
1435 2097
1436 2098
1437 2099
1438 2100
1439 2101
1440 2102
1441 2103
1442 2104
1443 2105
1444 2106
1445 2107
1446 2108
1447 2109
1448 2110
1449 2111
1450 2112
1451 2113
1452 2114
1453 2115
1454 2116
1455 2117
1456 2118
1457 2119
1458 2120
1459 2121
1460 2122
1461 2123
1462 2124
1463 2125
1464 2126
1465 2127
1466 2128
1467 2129
1468 2130
1469 2131
1470 2132
1471 2133
1472 2134
1473 2135
1474 2136
1475 2137
1476 2138
1477 2139
1478 2140
1479 2141
1480 2142
1481 2143
1482 2144
1483 2145
1484 2146
1485 2147
1486 2148
1487 2149
1488 2150
1489 2151
1490 2152
1491 2153
1492 2154
1493 2155
1494 2156
1495 2157
1496 2158
1497 2159
1498 2160
1499 2161
1500 2162
1501 2163
1502 2164
1503 2165
1504 2166
1505 2167
1506 2168
1507 2169
1508 2170
1509 2171
1510 2172
1511 2173
1512 2174
1513 2175
1514 2176
1515 2177
1516 2178
1517 2179
1518 2180
1519 2181
1520 2182
1521 2183
1522 2184
1523 2185
1524 2186
1525 2187
1526 2188
1527 2189
1528 2190
1529 2191
1530 2192
1531 2193
1532 2194
1533 2195
1534 2196
1535 2197
1536 2198
1537 2199
1538 2200
1539 2201
1540 2202
1541 2203
1542 2204
1543 2205
1544 2206
1545 2207
1546 2208
1547 2209
1548 2210
1549 2211
1550 2212
1551 2213
1552 2214
1553 2215
1554 2216
1555 2217
1556 2218
1557 2219
1558 2220
1559 2221
1560 2222
1561 2223
1562 2224
1563 2225
1564 2226
1565 2227
1566 2228
1567 2229
1568 2230
1569 2231
1570 2232
1571 2233
1572 2234
1573 2235
1574 2236
1575 2237
1576 2238
1577 2239
1578 2240
1579 2241
1580 2242
1581 2243
1582 2244
1583 2245
1584 2246
1585 2247
1586 2248
1587 224
```

92/11/09
12:16:06

/tmp/59201

436 1156 DONE PAC_play.cde
1157
437 1158

```
1093
1094
1095
1096 /*****
1097 *
1098 *   PAC_play
1099 *
1100 *   Description:   In this state a PA message is playing.
1101 *
1102 *   Resources Active: TAG port
1103 *                   Pseudo port (PAD) may be active, for uploading.
1104 *
1105 *   Critical Dependencies:
1106 *                   none.
1107 *
1108 *****/
1109
1110
1111 STATE PAC_play
1112
1113 EVENT ANY_ENT_OF_MESSAGE /* PA msgs finished playing */
1114
1115 FUNCTION /* tell BRL digit collector that PA msg is done playing */
1116
1117 TRANS <PAC_active>
1118
1119 EVENT PAC_STOP_PA /* agent wants to stop PA msg playing */
1120
1121 FUNCTION /* tell TAG LDH to stop playing */
1122
1123 TRANS <PAC_active>
1124
1125
1126 EVENT ANY_SEC_FAILURE /* Uploading PAD reported failure */
1127
1128 FUNCTION /* Clear the sys msg # from PA tbl associated with PAD */
1129
1130 FUNCTION /* tell TAG LDH to stop playing */
1131
1132 FUNCTION /* tell TAG LDH to stop playing */
1133
1134 FUNCTION /* tell BRL digit collector that PA msg is done playing */
1135
1136 TRANS <PAC_active>
1137
1138
1139 EVENT ANY_MSG_FAILURE /* TAG reported failure */
1140
1141 FUNCTION /* if there is a PAD currently uploading a message, */
1142
1143 FUNCTION /* tell it that PAC is going away. */
1144
1145 FUNCTION /* NOTE: there may be multiple PADs doing uploads */
1146
1147 FUNCTION /* PAC_snd_pad_ldh_idle if upload
1148
1149 FUNCTION /* deallocate DAS memory for PA messages */
1150
1151 FUNCTION /* get the PA TAG group member id */
1152
1153 FUNCTION /* PAC_get_pa_tag_grp_id
1154
1155 TRANS <PAC_get_tag_id>
1156
1157
1158 EVENT ANY_FORCE_COS
1159
1160 FUNCTION /* if there is a PAD currently uploading a message, */
1161
1162 FUNCTION /* tell it that PAC is going away */
1163
1164 FUNCTION /* PAC_snd_pad_ldh_idle if upload
1165
1166 FUNCTION /* idle the TAG port */
1167
1168 FUNCTION /* PAC_snd_tag_ldh_idle
1169
1170 FUNCTION /* deallocate DAS memory for PA messages */
1171
1172 FUNCTION /* PAC_dealloc_msgs_on_signout
1173
1174 FUNCTION /* check if PC had changed Staff table after signed in */
1175
1176 TRANS <PAC_cos>
1177
```

92/H1/09
13:16:06

/tmp/59201

```
1159 *****
1160
1161
1162
1163 /*****
1164 *
1165 *   PAC_select_option
1166 *
1167 *   Description:   In this state, the PA main menu is displayed to the
1168 *                 agent. The feature is active and agent can scan,
1169 *                 review, record or delete messages.
1170 *
1171 *   Resources Active: TAG port
1172 *                   PAD port may be active for uploading
1173 *
1174 *   Critical Dependencies:
1175 *                   none.
1176 *
1177 *****
1178
1179
1180 STATE   PAC_select_option
1181
1182 EVENT   PAC_NEW_REQ /* agent wants to look at next PA message */
1183          /* get the next PA table entry */
1184          FUNCTION   PAC_next_pa_entry_num
1185          /* send message status to the BPRH menu handler */
1186          FUNCTION   PAC_snd_bprh_mb_prh_pa_info
1187          <PAC_select_option>
1188
1189 TRANS
1190
1191 EVENT   PAC_PREV_REQ /* agent wants to look at previous PA message */
1192          /* get the previous PA table entry */
1193          FUNCTION   PAC_prev_pa_entry_num
1194          /* send message status to the BPRH menu handler */
1195          FUNCTION   PAC_snd_bprh_mb_prh_pa_info
1196          <PAC_select_option>
1197
1198 EVENT   PAC_EXIT /* agent exited the PA menu */
1199          /* tell VMS that PA menu is exited */
1200          FUNCTION   PAC_snd_vms_pac_deactivated
1201          <PAC_active>
1202
1203 TRANS
1204
1205 EVENT   PAC_PLAY_REQ /* agent wants to play the current PA message */
1206          /* tell TAG LDH to play the message */
1207          FUNCTION   PAC_snd_tag_ldh_play
1208          <PAC_play_option>
1209
1210 TRANS
1211
1212 EVENT   PAC_RECORD_REQ /* agent wants to record a message */
1213          /* if new message, allocate PA system message # */
1214          /* if existing message, allocate temp system message # */
1215          FUNCTION   PAC_alloc_pa_or_tmp_sys_msg_num
1216          <PAC_rec_alloc_sys_no>
1217
1218 TRANS
1219
1220 EVENT   PAC_STOP_REQ /* we get this before delete request */
1221          <PAC_select_option>
1222
1223 TRANS
1224
1225 EVENT   PAC_DELETE_REQ /* agent wants to delete the message */
1226          /* check message status */
1227          FUNCTION   PAC_chk_message_status
1228          <PAC_chk_delete_stat>
1229
1230 TRANS
1231
1232 EVENT   ANY_SDC_FAILURE /* Uploading PAC reported failure */
1233          <PAC_select_option.tde>
```

```
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3090
3091
3092
3093
3094
3095
3096
3097
3098
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3110
3111
3112
3113
3114
3115
3116
3117
3118
3119
3120
3121
3122
3123
3124
3125
3126
3127
3128
3129
3130
3131
3132
3133
3134
3135
3136
3137
3138
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3150
3151
3152
3153
3154
3155
3156
3157
3158
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3190
3191
3192
3193
3194
3195
3196
3197
3198
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3210
3211
3212
3213
3214
3215
3216
3217
3218
3219
3220
3221
3222
3223
3224
3225
3226
3227
3228
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238
3239
3240
3241
3242
3243
3244
3245
3246
3247
3248
3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269
3270
3271
3272
3273
3274
3275
3276
3277
3278
3279
3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3290
3291
3292
3293
3294
3295
3296
3297
3298
3299
3300
3301
3302
3303
3304
3305
3306
3307
3308
3309
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388
3389
3390
3391
3392
3393
3394
3395
3396
3397
3398
3399
3400
3401
3402
3403
3404
3405
3406
3407
3408
3409
3410
3411
3412
3413
3414
3415
3416
3417
3418
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3479
3480
3481
3482
3483
3484
3485
3486
3487
3488
3489
3490
3491
3492
3493
3494
3495
3496
3497
3498
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3510
3511
3512
3513
3514
3515
```

92/11/09
12:16:06

/tmp/59201

505 1330

```
1267
1268
1269
1270 /*****
1271 *
1272 * PAC_chk_delete_stat
1273 *
1274 * Description: This state is waiting for the message status check
1275 * to determine what actions to take for message deletion.
1276 *
1277 * Resources Active: TAG port
1278 * Pseudo port (PAD) may be active, for uploading.
1279 *
1280 * Critical Dependencies:
1281 * none.
1282 *
1283 *****/
1284 STATE PAC_chk_delete_stat
1285
1286
1287
1288
1289 EVENT PAC_UPLOADING /* message to be deleted is uploading */
1290 /* get PAD port id */
1291 FUNCTION PAC_get_pad_port_id_from_info
1292 /* tell PAC that the message is deleted */
1293 FUNCTION PAC_snd_pad_ldh_delete_pa
1294 /* clear the system message # in the read/write area */
1295 FUNCTION PAC_delete_msg_in_rw
1296 /* give BPRH the new message status */
1297 FUNCTION PAC_snd_bprh_mh_prh_pa_info_notrec
1298 TRANS <PAC_select_option>
1299
1300
1301 EVENT PAC_READY_TO_PLAY /* delete the message */
1302 /* deallocate the DAS message memory */
1303 FUNCTION PAC_dealloc_current_msg
1304 /* delete the PA message disk file */
1305 FUNCTION PAC_snd_upldr_das_remove_file
1306 /* delete the PA system message # from staff table */
1307 /* by sending PC_UPDATE_STAFF_PA to BMU RC Handler */
1308 FUNCTION PAC_rmvsys_msg_no_from_staff
1309 /* free up the PA system message number */
1310 FUNCTION PAC_free_pa_sys_msg_num
1311 /* clear the system message # in the read/write area */
1312 FUNCTION PAC_delete_msg_in_rw
1313 /* give BPRH the new message status */
1314 FUNCTION PAC_snd_bprh_mh_prh_pa_info_notrec
1315 TRANS <PAC_select_option>
1316
1317
1318 EVENT ANY_FORCE_OCS
1319 /* if there is a PAD currently uploading a message, */
1320 /* tell it that PAC is going away */
1321 FUNCTION PAC_snd_pad_ldh_idle_if_upload
1322 /* idle the TAG port */
1323 FUNCTION PAC_snd_tag_ldh_idle
1324 /* deallocate DAS memory for PA messages */
1325 FUNCTION PAC_dealloc_msgs_on_signout
1326 /* check if RC had changed Staff table after signed in */
1327 FUNCTION PAC_chk_staff_chg_after_signin
1328 TRANS <PAC_000>
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3090
3091
3092
3093
3094
3095
3096
3097
3098
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3110
3111
3112
3113
3114
3115
3116
3117
3118
3119
3120
3121
3122
3123
3124
3125
3126
3127
3128
3129
3130
3131
3132
3133
3134
3135
3136
3137
3138
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3150
3151
3152
3153
3154
3155
3156
3157
3158
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3190
3191
3192
3193
3194
3195
3196
3197
3198
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3210
3211
3212
3213
3214
3215
3216
3217
3218
3219
3220
3221
3222
3223
3224
3225
3226
3227
3228
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238
3239
3240
3241
3242
3243
3244
3245
3246
3247
3248
3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269
3270
3271
3272
3273
3274
3275
3276
3277
3278
3279
3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3290
3291
3292
3293
3294
3295
3296
3297
3298
3299
3300
3301
3302
3303
3304
3305
3306
3307
3308
3309
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388
3389
3390
3391
3392
3393
3394
3395
3396
3397
3398
3399
3400
3401
3402
3403
3404
3405
3406
3407
3408
3409
3410
3411
3412
3413
3414
3415
3416
3417
3418
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3479
3480
3481
3482
3483
3484
3485
3486
3487
3488
3489
3490
3491
3492
3493
3494
3495
3496
3497
3498
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578
3579
3580
3581
3582
3583
3584
3585
3586
3587
3588
3589
3590
3591
3592
3593
3594
3595
3596
3597
3598
3599
3600
3601
3602
3603
3604
3605
3606
3607
3608
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619
3620
3621
3622
3623
3624
3625
3626
3627
3628
3629
3630
3631
3
```

92/11/09
13:16:06

/tmp/59201

2103204

```
1331
1332
1333
1334 /*****
1335 *
1336 * PAC_play_option
1337 *
1338 * Description: In this state, a PA message is being reviewed by an
1339 * agent.
1340 *
1341 * Resources Active: TAG port
1342 * PAD port may be active for uploading
1343 *
1344 * Critical Dependencies:
1345 * none.
1346 *
1347 *****/
1348
1349
1350 STATE PAC_play_option
1351
1352
1353 EVENT ANY_END_OF_MESSAGE /* message finished playing */
1354 /* get the End of Message (EOM) cone TAG id */
1355 FUNCTION PAC_get_eom_tag_frp_id
1356 <PAC_get_eom_tag>
1357
1358 EVENT PAC_PLAY_REC /* agent wants to replay the current PA message */
1359 /* tell TAG LDH to play the message */
1360 FUNCTION PAC_snd_tag_ldh_play
1361 <PAC_play_option>
1362
1363 EVENT PAC_STOP_REC /* stop playing */
1364 /* tell TAG LDH to stop playing the message */
1365 FUNCTION PAC_snd_tag_ldh_stop_play
1366 <PAC_select_option>
1367
1368 EVENT PAC_RECORD_REC /* agent wants to record a message */
1369 /* tell TAG LDH to stop playing the message */
1370 FUNCTION PAC_snd_tag_ldh_stop_play
1371
1372 FUNCTION /* existing message; allocate temp system message */
1373 <PAC_rec_alloc_sys_nc>
1374
1375 EVENT PAC_EXIT /* agent exited the PA menu */
1376 /* tell TAG LDH to stop playing */
1377 FUNCTION PAC_snd_tag_ldh_stop_play
1378
1379 FUNCTION /* tell VMS that PA menu is exited */
1380 <PAC_active>
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
```

```
FUNCTION PAC_dealloc_msgs_on_signout
/* tell VMS that PA menu is exited */
FUNCTION PAC_snd_vms_pac_deactivated
/* send BPRH_PA_FAILURE to the BPRH menu handler */
FUNCTION PAC_snd_bprh_mh_prh_pa_failure
<PAC_wait_failure_ack>
TRANS
EVENT ANY_MSG_FAILURE /* TAG reported failure */
/* if there is a PAD currently uploading a message, */
/* tell it that PAC is going away. */
/* NOTE: there may be multiple PADs doing uploads */
FUNCTION PAC_snd_pad_ldh_idle_if_upload
/* deallocate DAS memory for PA messages */
FUNCTION PAC_dealloc_msgs_on_signout
/* send BPRH_PA_FAILURE to the BPRH menu handler */
FUNCTION PAC_snd_bprh_mh_prh_pa_failure
<PAC_wait_failure_ack>
TRANS
EVENT ANY_FORCE_OOS
/* if there is a PAD currently uploading a message, */
/* tell it that PAC is going away */
FUNCTION PAC_snd_pad_ldh_idle_if_upload
/* idle the TAG port */
FUNCTION PAC_snd_tag_ldh_idle
/* deallocate DAS memory for PA messages */
FUNCTION PAC_dealloc_msgs_on_signout
/* check if PC had changed Staff table after signed in */
FUNCTION PAC_chk_staff_chng_after_signin
<PAC_oos>
TRANS
PAC_play_option.tde
DONE
```

92/EL/89
12:16:06

/tmp/59201

18

```
1427 /*****
1428
1429
1430 /*****
1431 *
1432 * PAC_wt_eom_tag
1433 *
1434 * Description: This state is waiting for the check
1435 * to determine whether eom TAG group is defined.
1436 *
1437 * Resources Active: TAG port
1438 * Pseudo port (PAD) may be active, for uploading.
1439 *
1440 * Critical Dependencies:
1441 * none.
1442 *
1443 *****/
1444
1445
1446
1447 STATE PAC_wt_eom_tag
1448
1449 EVENT PAC_INO_TAG_ID /* no eom TAG group defined */
1450 TRANS <PAC_select_option>
1451
1452 EVENT PAC_INO_TAG_ID /* valid eom TAG group */
1453 /* set eom tone timer */
1454 FUNCTION PAC_set_eom_tone_timer
1455 /* get eom tone TAG port */
1456 FUNCTION PAC_snd_tag_ldh_acquire_eom
1457 TRANS <PAC_start_eom>
1458
1459 EVENT ANI_FORCE_PDS
1460 /* if there is a PAD currently uploading a message, */
1461 /* tell it that PAC is going away */
1462 FUNCTION PAC_snd_pad_ldh_idle_if_upload
1463 /* idle the TAG port */
1464 FUNCTION PAC_snd_tag_ldh_idle
1465 /* deallocate DAS memory for PA messages */
1466 FUNCTION PAC_dealloc_msgs_on_signout
1467 /* check if PC had changed Staff table after signed in */
1468 FUNCTION PAC_chk_staff_chg_after_signin
1469 TRANS <PAC_dds>
1470
1471 DONE
1472
1473
```

```
1474
1475
1476
1477 /*****
1478 *
1479 * PAC_start_eom
1480 *
1481 * Description: This state is waiting for the eom tone TAG port.
1482 *
1483 * Resources Active: TAG port
1484 * Pseudo port (PAD) may be active, for uploading.
1485 *
1486 * Critical Dependencies:
1487 * none.
1488 *
1489 *****/
1490
1491
1492
1493 STATE PAC_start_eom
1494
1495 EVENT ANI_ACTIVATE_ACK /* eom started */
1496 /* save EOM tone TAG port id */
1497 FUNCTION PAC_save_tone_tag_port_id
1498 TRANS <PAC_eom_playing>
1499
1500 EVENT PAC_PLAY_CIF_TIMEOUT /* timed out waiting for eom tone */
1501 /* clear the timer id */
1502 FUNCTION PAC_clear_timer_id
1503 /* cancel eom port request */
1504 FUNCTION PAC_snd_tag_ldh_cancel_acquire_eom
1505 TRANS <PAC_select_option>
1506
1507 EVENT PAC_EXIT /* agent exited the PA menu */
1508 /* cancel timer */
1509 FUNCTION PAC_cancel_timer
1510 /* cancel eom port request */
1511 FUNCTION PAC_snd_tag_ldh_cancel_acquire_eom
1512 /* tell VMS that PA menu is exited */
1513 FUNCTION PAC_snd_vms_pac_deactivated
1514 TRANS <PAC_active>
1515
1516 EVENT PAC_PLAY_REQ /* agent wants to replay the current PA message */
1517 /* cancel timer */
1518 FUNCTION PAC_cancel_timer
1519 /* cancel eom port request */
1520 FUNCTION PAC_snd_tag_ldh_cancel_acquire_eom
1521 /* tell PA TAG LDH to play the message */
1522 FUNCTION PAC_snd_tag_ldh_play
1523 TRANS <PAC_play_option>
1524
1525 EVENT PAC_STOP_REQ /* stop playing */
1526 /* cancel timer */
1527 FUNCTION PAC_cancel_timer
1528 /* cancel eom port request */
1529 FUNCTION PAC_snd_tag_ldh_cancel_acquire_eom
1530 TRANS <PAC_select_option>
1531
1532 EVENT PAC_RECORD_REQ /* agent wants to record a message */
1533 /* cancel timer */
1534 FUNCTION PAC_cancel_timer
1535 /* cancel eom port request */
1536 FUNCTION PAC_snd_tag_ldh_cancel_acquire_eom
1537
```

2103204

92/11/09
12:16:06

/tmp/59201

```
1537
586 1538
587 1539
1540
588 1541
1542
589 1543
590 1544
1545
591 1546
1547
592 1548
1549
593 1550
1551
594 1552
1553
595
596
597
598
599
600
1566
601 1567
1568
602 1569
1570
603 1571
1572
1573
1574
604 1575
1576
605 1577
1578
606 79
607 1581
608 1582
1583
609 1584
1585
610 1586
1587
611 1588
1589
1590
612 1591
1592
613 1593
1594
614 1595
1596
615 1597
616 1598
1599
617 1600 DONE
PAC_start_eom.tde

/* existing message; allocate temp system message */
FUNCTION PAC_alloc_tmp_sys_msg_num 1601
<PAC_alloc_sys_no> 618 1602

EVENT ANY_ACTIVATE_NACK /* did not get eom port */
/* cancel timer */
FUNCTION PAC_cancel_timer
<PAC_select_option>

EVENT ANY_SDC_FAILURE /* Uploading PAD reported failure */
/* cancel timer */
FUNCTION PAC_cancel_timer
/* cancel eom port request */
FUNCTION PAC_snd_tag_lgh_cancel_acquire_eom
/* clear the sys_msg from PA tbl associated with PAD */
FUNCTION PAC_clear_upload_fail_pad_sys_msg_num
/* if there is a PAD currently uploading a message, */
/* tell it that PAC is going away. */
/* NOTE: there may be multiple PADs doing uploads */
FUNCTION PAC_snd_pad_ldh_idle_if_upload
/* deallocate DAS memory for PA messages */
FUNCTION PAC_dealloc_msgs_on_signout
/* idle the TAG port */
FUNCTION PAC_snd_tag_ldh_idle
/* tell VMS that PA menu is exited */
FUNCTION PAC_snd_vms_pac_deactivated
/* send PRH_PA_FAILURE to the BPRH menu handler */
FUNCTION PAC_snd_bprh_mh_prh_pa_failure
<EAC_wait_failure_ack>

EVENT ANY_MSG_FAILURE /* TAG reported failure */
/* cancel timer */
FUNCTION PAC_cancel_timer
/* cancel eom port request */
FUNCTION PAC_snd_tag_lgh_cancel_acquire_eom
/* if there is a PAD currently uploading a message, */
/* tell it that PAC is going away. */
/* NOTE: there may be multiple PADs doing uploads */
FUNCTION PAC_snd_pad_ldh_idle_if_upload
/* tell VMS that PA menu is exited */
FUNCTION PAC_snd_vms_pac_deactivated
/* deallocate DAS memory for PA messages */
FUNCTION PAC_dealloc_msgs_on_signout
/* send PRH_PA_FAILURE to the BPRH menu handler */
FUNCTION PAC_snd_bprh_mh_prh_pa_failure
<PAC_wait_failure_ack>

EVENT ANY_FORCE_GOS
/* cancel timer */
FUNCTION PAC_cancel_timer
/* cancel eom port request */
FUNCTION PAC_snd_tag_lgh_cancel_acquire_eom
/* if there is a PAD currently uploading a message, */
/* tell it that PAC is going away */
FUNCTION PAC_snd_pad_ldh_idle_if_upload
/* idle the TAG port */
FUNCTION PAC_snd_tag_ldh_idle
/* deallocate DAS memory for PA messages */
FUNCTION PAC_dealloc_msgs_on_signout
/* check if PC had changed Staff table after signed in */
FUNCTION PAC_chk_staff_chq_after_signin
<EAC_gos>
TRANS PAC_start_eom.tde
```

2103204

19

643	1667	1666
644	1668	1669
645	1670	1671
646	1672	1673
647	1674	1675
648	1676	1677
649	1678	1679
650	1680	1681
651	1682	1683
652	1684	1685
653	1686	1687
654	1688	1689
655	1690	1691
656	1692	1693
657	1694	1695
658	1696	1697
659	1698	1699
660	1700	1701
661	1702	1703
662	1704	1705
663	1706	1707
664	1708	1709
665	1710	

92/HU09
12:16:06

/tmp/59201

21

```
1711 *****
1712 PAC_eom_chk_fail_tag
1713
1714
1715 /*****
1716
1717 *
1718 * PAC_eom_chk_fail_tag
1719 *
1720 * Description: This transition state is waiting to determine
1721 * which TAG port failed while eom tone was playing.
1722 *
1723 * Resources Active: PA TAG port
1724 * EDM TAG port.
1725 * Pseudo port (PAD) may be active, for uploading.
1726 *
1727 * Critical Dependencies:
1728 * none.
1729 *
1730 *****
1731
1732
1733 STATE PAC_eom_chk_fail_tag
1734
1735
1736 EVENT PAC_i_pa_tag_fail /* PA TAG port failed */
1737
1738 FUNCTION PAC_snd_tag_ldh_idle_tone
1739 * If PAD is doing upload currently */
1740 * tell it that PAC is going away. */
1741 * NOTE: there may be multiple PADS doing uploads */
1742 FUNCTION PAC_snd_pad_ldh_idle_if_upload
1743 * deallocate DAS memory for PA messages */
1744 FUNCTION PAC_dealloc_msgs_on_signout
1745 * send PRH PA_FAILURE to the BPRH menu handler */
1746 FUNCTION PAC_snd_bprh_mh_prh_pa_failure
1747 <PAC_wait_failure_ack>
1748
1749
1750 EVENT PAC_i_tone_tag_fail /* tone TAG port failed; we dont care */
1751 TRANS <PAC_select_option>
1752
1753
1754 ANY_FORCE_OOS
1755
1756 FUNCTION * idle the eom port */
1757 PAC_snd_tag_ldh_idle_tone
1758
1759 FUNCTION * idle the TAG port */
1760 PAC_snd_tag_ldh_idle
1761
1762 FUNCTION * If there is a PAD currently uploading a message, */
1763 * tell it that PAC is going away */
1764 PAC_snd_pad_ldh_idle_if_upload
1765
1766 FUNCTION * deallocate DAS memory for PA messages */
1767 PAC_dealloc_msgs_on_signout
1768
1769 FUNCTION /* check if RC had changed Staff table after signed in */
1770 PAC_chk_staff_chg_after_signin
1771
1772 TRANS <PAC_oos>
1773
1774
1775 PAC_eom_chk_fail_tag.tde
1776
1777 DONE
1778
1779
```

2103204

92/11/09
13:16:06

/tmp/59201

22

```
1819
1820
1821
1822 /*****
1823 *
1824 * PAC_record_option
1825 *
1826 * Description:
1827 * This state is waiting for the DAS memory allocation
1828 * for recording. If this is re-recording of an existing
1829 * message, then the memory is allocated for the
1830 * temporary system message #. If it is for new recording
1831 * then memory is allocated for the PA system message #.
1832 *
1833 * Resources Active: TAG port
1834 * Pseudo port (PAD) may be active, for uploading.
1835 *
1836 * Critical Dependencies:
1837 * none.
1838 *
1839 *****/
1840
1841 STATE PAC_record_option
1842
1843
1844 EVENT ANY_ALLOCATE_COMPLETE /* got memory */
1845 /* cancel timer */
1846 FUNCTION PAC_cancel_timer
1847 /* save DAS message & unit #s */
1848 FUNCTION PAC_set_das_msg_info
1849 /* send PRH_RECORD_ACK */
1850 FUNCTION PAC_snd_bprh_mh_prh_record_ack
1851 /* send message status to the BPRH menu handler */
1852 FUNCTION PAC_snd_bprh_mh_prh_pa_info_notrec
1853 <PAC_start_record>
1854
1855 EVENT ANY_ALLOCATE_MSG_NACK /* could not get memory */
1856 /* cancel timer */
1857 FUNCTION PAC_cancel_timer
1858 /* set memory retry timer */
1859 FUNCTION PAC_set_wait_memory_timer
1860 /* free up the system message # */
1861 FUNCTION PAC_free_pa_or_tmp_sys_msg_num
1862 /* tell PRH that there is no memory currently */
1863 FUNCTION PAC_snd_bprh_mh_prh_pa_das_full
1864 <PAC_wait_record_memory>
1865
1866 EVENT ANY_SDC_FAILURE /* Uploading PAD reported failure */
1867 /* cancel timer */
1868 FUNCTION PAC_cancel_timer
1869 /* clear the sys msg # from PA tbl associated with PAD */
1870 FUNCTION PAC_clear_upload_fall_pad_sys_msg_num
1871 /* we did not receive ANY_ALLOCATE_COMPLETE yet */
1872 /* but it could be on the way */
1873 FUNCTION PAC_das_mem_dealloc_pa_or_tmp_sys_msg_num
1874 /* free up the system message # */
1875 FUNCTION PAC_free_pa_or_tmp_sys_msg_num
1876 /* if there is a PAD currently uploading a message, */
1877 /* tell it that PAC is going away. */
1878 /* NOTE: there may be multiple PADs doing uploads */
1879 FUNCTION PAC_snd_pad_ldh_idle_if_upload
1880 /* deallocate DAS memory for PA messages */
1881 FUNCTION PAC_dealloc_msgs_on_signout
```

```
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
```

```
/* idle the TAG port */
FUNCTION PAC_snd_tag_ldh_idle
/* tell VMS that PA menu is exited */
FUNCTION PAC_snd_vms_pac_deactivated
/* send PRH_PA_FAILURE to the BPRH menu handler */
FUNCTION PAC_snd_bprh_mh_prh_pa_failure
<PAC_wait_failure_ack>

TRANS

EVENT ANY_MSG_FAILURE /* TAG reported failure */
/* cancel timer */
FUNCTION PAC_cancel_timer
/* we did not receive ANY_ALLOCATE_COMPLETE yet */
/* but it could be on the way */
FUNCTION PAC_das_mem_dealloc_pa_or_tmp_sys_msg_num
/* free up the system message # */
FUNCTION PAC_free_pa_or_tmp_sys_msg_num
/* if there is a PAD currently uploading a message, */
/* tell it that PAC is going away. */
/* NOTE: there may be multiple PADs doing uploads */
FUNCTION PAC_snd_pad_ldh_idle_if_upload
/* deallocate DAS memory for PA messages */
FUNCTION PAC_dealloc_msgs_on_signout
/* tell VMS that PA menu is exited */
FUNCTION PAC_snd_vms_pac_deactivated
/* send PRH_PA_FAILURE to the BPRH menu handler */
FUNCTION PAC_snd_bprh_mh_prh_pa_failure
<PAC_wait_failure_ack>

TRANS

EVENT ANY_FORCE_OOS
/* cancel timer */
FUNCTION PAC_cancel_timer
/* we did not receive ANY_ALLOCATE_COMPLETE yet */
/* but it could be on the way */
FUNCTION PAC_das_mem_dealloc_pa_or_tmp_sys_msg_num
/* free up the system message # */
FUNCTION PAC_free_pa_or_tmp_sys_msg_num
/* if there is a PAD currently uploading a message, */
/* tell it that PAC is going away. */
FUNCTION PAC_snd_pad_ldh_idle_if_upload
/* idle the TAG port */
FUNCTION PAC_snd_tag_ldh_idle
/* deallocate DAS memory for PA messages */
FUNCTION PAC_dealloc_msgs_on_signout
/* check if RC had changed Staff table after signed in */
FUNCTION PAC_chk_staff_chg_after_signin
<PAC_oos>

TRANS

EVENT PAC_I_WAIT_ACT_TIMEOUT /* timed out waiting for response */
/* clear the timer id */
FUNCTION PAC_clear_timer_id
/* we did not receive ANY_ALLOCATE_COMPLETE yet */
/* but it could be on the way */
FUNCTION PAC_das_mem_dealloc_pa_or_tmp_sys_msg_num
/* free up the system message # */
FUNCTION PAC_free_pa_or_tmp_sys_msg_num
/* if there is a PAD currently uploading a message, */
/* tell it that PAC is going away. */
FUNCTION PAC_snd_pad_ldh_idle_if_upload
/* idle the TAG port */
FUNCTION PAC_snd_tag_ldh_idle
/* deallocate DAS memory for PA messages */
FUNCTION PAC_dealloc_msgs_on_signout
/* check if RC had changed Staff table after signed in */
FUNCTION PAC_chk_staff_chg_after_signin
```

2103204

92/11/09
12:16:06

/tmp/59201

23

```
1946
1947
1948
1949
1950
1951
1952 DONE
1953
1954

/* tell VMS that PA menu is exited */
FUNCTION PAC_snd_vms_pac_deactivated
/* send PRH_PA_FAILURE to the BPRH menu handler */
FUNCTION PAC_snd_bprh_mh_prh_pa_failure
<PAC_wait_failure_ack>
TRANS
PAC_record_option.cde
```

```
1955 /*****
1956
1957
1958
1959
1960 *
1961 * PAC_wt_record_memory
1962 *
1963 *
1964 *
1965 *
1966 *
1967 *
1968 *
1969 *
1970 *
1971 *
1972
1973
1974
1975
1976 STATE PAC_wt_record_memory
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
```

```

1955
1956
1957
1958
1959 /*****
1960 *
1961 * PAC_wt_record_memory
1962 *
1963 *
1964 *
1965 *
1966 *
1967 *
1968 *
1969 *
1970 *
1971 *
1972
1973
1974
1975
1976 STATE PAC_wt_record_memory
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017

EVENT PAC_i_mem_retry_timeout /* try to get sys msg & and memory again
/* if new message, allocate PA system message & */
/* if existing message, allocate temp system message & */
FUNCTION PAC_alloc_pa_or_tmp_sys_msg_num
<PAC_rec_alloc_sys_no>
TRANS
PAC_ABORT_RECORD /* agent aborted recording */
<PAC_select_option>
TRANS
ANY_SDC_FAILURE /* Uploading PAD reported failure */
/* clear the sys msg & from PA tbl associated with PAD */
FUNCTION PAC_clear_upload_fail_pad_sys_msg_num
/* if there is a PAD currently uploading a message, */
/* tell it that PAC is going away. */
/* NOTE: there may be multiple PADs doing uploads */
FUNCTION PAC_snd_pad_ldh_idle_if_upload
/* deallocate DAS memory for PA messages */
FUNCTION PAC_dealloc_msgs_on_signout
/* idle the TAG port */
FUNCTION PAC_snd_tag_ldh_idle
/* tell VMS that PA menu is exited */
FUNCTION PAC_snd_vms_pac_deactivated
/* send PRH_PA_FAILURE to the BPRH menu handler */
FUNCTION PAC_snd_bprh_mh_prh_pa_failure
<PAC_wait_failure_ack>
TRANS
ANY_MSG_FAILURE /* TAG reported failure */
/* if there is a PAD currently uploading a message, */
/* tell it that PAC is going away. */
/* NOTE: there may be multiple PADs doing uploads */
FUNCTION PAC_snd_pad_ldh_idle_if_upload
/* deallocate DAS memory for PA messages */
FUNCTION PAC_dealloc_msgs_on_signout
/* tell VMS that PA menu is exited */
FUNCTION PAC_snd_vms_pac_deactivated
/* send PRH_PA_FAILURE to the BPRH menu handler */
FUNCTION PAC_snd_bprh_mh_prh_pa_failure
<PAC_wait_failure_ack>
TRANS
ANY_FORCE_OOS
EVENT
```

2103204

535-242-2222

/tmp/59201

```

2018 /* if there is a PAD currently uploading a message, */
2019 /* tell it that PAC is going away */
2020 FUNCTION PAC_snd_pad_ldh_idle_if_upload
2021 /* idle the TAG port */
2022 FUNCTION PAC_snd_tag_ldh_idle
2023 /* deallocate DAS memory for PA messages */
2024 FUNCTION PAC_dealloc_msgs_on_signout
2025 /* check if PC had changed Staff table after signed in */
2026 FUNCTION PAC_chk_staff_chg_after_signin
2027 TRANS <PAC_pos>
2028
2029 DONE PAC_wt_record_memory.tide
2030
2031
2032
2033
2034
2035 /*****
2036 * PAC_start_record
2037 *
2038 * Description: This state is waiting for the indication from the PPH
2039 * to start recording.
2040 *
2041 *
2042 * Resources Active: TAG port
2043 * Memory and PA or temporary system message # for recording
2044 * Pseudo port (PAD) may be active, for uploading.
2045 *
2046 * Critical Dependencies:
2047 * none.
2048 *
2049 *****/
2050
2051
2052
2053
2054
2055 STATE PAC_start_record
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
28
```

2103204

92/11/09
12:16:06

/tmp/59201

```
2095 /* NOTE: there may be multiple PADs doing uploads */
2096 FUNCTION PAC_snd_pad_ldh_idle_if_upload
2097 /* deallocate DAS memory for PA messages */
2098 FUNCTION PAC_dealloc_msgs_on_signout
2099 /* send PPH_PA_FAILURE to the BPRH menu handler */
2100 FUNCTION PAC_snd_bprh_mh_prh_pa_failure
2101 <PAC_wait_failure_ack>
2102
2103 TRANS
2104
2105 EVENT ANY_FORCE_OOS
2106 /* deallocate the DAS message memory */
2107 FUNCTION PAC_das_mem_dealloc_pa_or_tmp_sys_msg_num
2108 /* free up the PA/temporary system msg # */
2109 FUNCTION PAC_free_pa_or_tmp_sys_msg_num
2110 /* if there is a PAD currently uploading a message, */
2111 /* tell it that PAC is going away */
2112 FUNCTION PAC_snd_pad_ldh_idle_if_upload
2113 /* idle the TAG port */
2114 FUNCTION PAC_snd_tag_ldh_idle
2115 /* deallocate DAS memory for PA messages */
2116 FUNCTION PAC_dealloc_msgs_on_signout
2117 /* check if RC had changed Staff table after signed in */
2118 FUNCTION PAC_chk_staff_chg_after_signin
2119 <PAC_oos>
2120
2121 TRANS
2122
2123 PAC_start_record.tde
2124 DONE
2125
2126
```

```
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143 STATE PAC_wt_pre_zip_tag
2144
2145 EVENT PAC_i_tag_id /* valid zip TAG group */
2146 /* get zip tone TAG port */
2147 FUNCTION PAC_snd_tag_ldh_acquire_zip
2148 /* set zip tone timer */
2149 FUNCTION PAC_set_zip_tone_timer
2150 <PAC_start_pre_zip>
2151
2152 EVENT PAC_i_no_tag_id /* no zip tone TAG group defined */
2153 /* skip the zip; start recording */
2154 FUNCTION PAC_snd_bprh_mh_prh_pa_start_record_ack
2155 /* tell TAG LDH to start recording */
2156 FUNCTION PAC_snd_tag_ldh_record
2157 <PAC_recording>
2158
2159 EVENT ANY_FORCE_OOS
2160 /* deallocate the DAS message memory */
2161 FUNCTION PAC_das_mem_dealloc_pa_or_tmp_sys_msg_num
2162 /* free up the PA/temporary system msg # */
2163 FUNCTION PAC_free_pa_or_tmp_sys_msg_num
2164 /* if there is a PAD currently uploading a message, */
2165 /* tell it that PAC is going away */
2166 FUNCTION PAC_snd_pad_ldh_idle_if_upload
2167 /* idle the TAG port */
2168 FUNCTION PAC_snd_tag_ldh_idle
2169 /* deallocate DAS memory for PA messages */
2170 FUNCTION PAC_dealloc_msgs_on_signout
2171 /* check if RC had changed Staff table after signed in */
2172 FUNCTION PAC_chk_staff_chg_after_signin
2173 <PAC_oos>
2174
2175 TRANS
2176
2177 PAC_wt_pre_zip_tag.tde
2178 DONE
2179
2180
```

2103204

60/11/09
12:16:06

/tmap/59201



```

2178 2181 /*****
2179 2182 *
2180 2183 * PAC_start_pre_zip
2181 2184 *
2182 2185 * Description: This state is waiting for the zip tone TAG port
2183 2186 * for applying pre-recording zip.
2184 2187 *
2185 2188 * Resources Active: TAG port
2186 2189 * Pseudo Port (PAD) may be active, for uploading.
2187 2190 *
2188 2191 * Critical Dependencies:
2189 2192 * none.
2190 2193 *
2191 2194 *****/
2192 5
2193 --96
2194 2197 STATE PAC_start_pre_zip
2195 2198
2196 2199
2197 2200 EVENT ANY_ACTIVATE_ACR /* zip tone started */
2198 2201 /* save zip tone TAG Id */
2199 2202 FUNCTION PAC_save_tone_tag_port_id
2200 2203 TRANS <PAC_pre_zip_playing>
2201 2204
2202 2205 EVENT PAC_PLAY_ZIP_TIMEOUT /* timed out waiting for zip tone */
2203 2206 /* clear the timer id */
2204 2207 FUNCTION PAC_clear_timer_id
2205 2208 /* cancel zip port request */
2206 2209 FUNCTION PAC_snd_tag_lgh_cancel_acquire_zip
2207 2210 /* skip the zip; start recording */
2208 2211 FUNCTION PAC_snd_bprh_mh_prh_pa_start_record_ack
2209 2212 /* tell TAG LDH to start recording */
2210 2213 FUNCTION PAC_snd_tag_ldh_record
2211 2214 TRANS <PAC_recording>
2212 2215
2213 2216 EVENT PAC_RECORD_EXIT /* agent exited from record menu */
2214 2217 /* cancel timer */
2215 2218 FUNCTION PAC_cancel_timer
2216 2219 /* cancel zip port request */
2217 2220 FUNCTION PAC_snd_tag_lgh_cancel_acquire_zip
2218 2221 /* deallocate the DAS message memory */
2219 2222 FUNCTION PAC_das_mem_dealloc_pa_or_tmp_sys_msg_num
2220 2223 /* free up the PA/temporary system msg # */
2221 2224 FUNCTION PAC_free_pa_or_tmp_sys_msg_num
2222 2225 TRANS <PAC_select_option>
2223 2226
2224 2227 EVENT ANY_SDO_FAILURE /* Uploading PAD reported failure */
2225 2228 /* cancel timer */
2226 2229 FUNCTION PAC_cancel_timer
2227 2230 /* cancel zip port request */
2228 2231 FUNCTION PAC_snd_tag_lgh_cancel_acquire_zip
2229 2232 /* clear the sys msg # from PA tbl associated with PAD */
2230 2233 FUNCTION PAC_clear_upload_fail_pad_sys_msg_num
2231 2234 /* deallocate the DAS message memory */
2232 2235 FUNCTION PAC_das_mem_dealloc_pa_or_tmp_sys_msg_num
2233 2236 /* free up the PA/temporary system msg # */
2234 2237 FUNCTION PAC_free_pa_or_tmp_sys_msg_num
2235 2238 /* idle the TAG port */
2236 2239 FUNCTION PAC_snd_tag_ldh_idle
2237 2240 /* If there is a PAD currently uploading
2238 2241

```

```

2178 2181 /*****
2179 2182 *
2180 2183 * PAC_start_pre_zip
2181 2184 *
2182 2185 * Description: This state is waiting for the zip tone TAG port
2183 2186 * for applying pre-recording zip.
2184 2187 *
2185 2188 * Resources Active: TAG port
2186 2189 * Pseudo Port (PAD) may be active, for uploading.
2187 2190 *
2188 2191 * Critical Dependencies:
2189 2192 * none.
2190 2193 *
2191 2194 *****/
2192 5
2193 --96
2194 2197 STATE PAC_start_pre_zip
2195 2198
2196 2199
2197 2200 EVENT ANY_ACTIVATE_ACR /* zip tone started */
2198 2201 /* save zip tone TAG Id */
2199 2202 FUNCTION PAC_save_tone_tag_port_id
2200 2203 TRANS <PAC_pre_zip_playing>
2201 2204
2202 2205 EVENT PAC_PLAY_ZIP_TIMEOUT /* timed out waiting for zip tone */
2203 2206 /* clear the timer id */
2204 2207 FUNCTION PAC_clear_timer_id
2205 2208 /* cancel zip port request */
2206 2209 FUNCTION PAC_snd_tag_lgh_cancel_acquire_zip
2207 2210 /* skip the zip; start recording */
2208 2211 FUNCTION PAC_snd_bprh_mh_prh_pa_start_record_ack
2209 2212 /* tell TAG LDH to start recording */
2210 2213 FUNCTION PAC_snd_tag_ldh_record
2211 2214 TRANS <PAC_recording>
2212 2215
2213 2216 EVENT PAC_RECORD_EXIT /* agent exited from record menu */
2214 2217 /* cancel timer */
2215 2218 FUNCTION PAC_cancel_timer
2216 2219
2217 2220 FUNCTION PAC_snd_tag_lgh_cancel_acquire_zip
2218 2221 /* deallocate the DAS message memory */
2219 2222 FUNCTION PAC_das_mem_dealloc_pa_or_tmp_sys_msg_num
2220 2223 /* free up the PA/temporary system msg # */
2221 2224 FUNCTION PAC_free_pa_or_tmp_sys_msg_num
2222 2225 TRANS <PAC_select_option>
2223 2226
2224 2227 EVENT ANY_SDO_FAILURE /* Uploading PAD reported failure */
2225 2228 /* cancel timer */
2226 2229 FUNCTION PAC_cancel_timer
2227 2230
2228 2231 FUNCTION PAC_snd_tag_lgh_cancel_acquire_zip
2229 2232 /* clear the sys msg # from PA tbl associated with PAD */
2230 2233 FUNCTION PAC_clear_upload_fail_pad_sys_msg_num
2231 2234 /* deallocate the DAS message memory */
2232 2235 FUNCTION PAC_das_mem_dealloc_pa_or_tmp_sys_msg_num
2233 2236 /* free up the PA/temporary system msg # */
2234 2237 FUNCTION PAC_free_pa_or_tmp_sys_msg_num
2235 2238 /* idle the TAG port */
2236 2239 FUNCTION PAC_snd_tag_ldh_idle
2237 2240 /* If there is a PAD currently uploading
2238 2241

```

2103204

92/11/85
12:16:46

/tmp/59201

2103204

```
2296
2297
2298
2299
2300 /*****
2301 *
2302 * PAC_pre_zip_playing
2303 *
2304 * Description: In this state the pre-recording zip tone is playing.
2305 *
2306 * Resources Active: PA TAG port
2307 * zip TAG port.
2308 * Pseudo: port (PAD) may be active, for uploading.
2309 *
2310 * Critical Dependencies:
2311 * none.
2312 *
2313 *****/
2314
2315
2316
2317 STATE PAC_pre_zip_playing
2318
2319 EVENT PAC_PLAY_TIMEOUT /* done playing zip tone */
2320
2321 FUNCTION PAC_clear_timer /* clear the timer id */
2322
2323 FUNCTION /* idle the zip TAG port */
2324
2325 FUNCTION /* tell PRH that recording is started */
2326
2327 FUNCTION /* tell TAG LCH to start recording */
2328
2329 TRANS <PAC_recording>
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3090
3091
3092
3093
3094
3095
3096
3097
3098
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3110
3111
3112
3113
3114
3115
3116
3117
3118
3119
3120
3121
3122
3123
3124
3125
3126
3127
3128
3129
3130
3131
3132
3133
3134
3135
3136
3137
3138
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3150
3151
3152
3153
3154
3155
3156
3157
3158
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3190
3191
3192
3193
3194
3195
3196
3197
3198
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3210
3211
3212
3213
3214
3215
3216
3217
3218
3219
3220
3221
3222
3223
3224
3225
3226
3227
3228
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238
3239
3240
3241
3242
3243
3244
3245
3246
3247
3248
3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269
3270
3271
3272
3273
3274
3275
3276
3277
3278
3279
3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3290
3291
3292
3293
3294
3295
3296
3297
3298
3299
3300
3301
3302
3303
3304
3305
3306
3307
3308
3309
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388
3389
3390
3391
3392
3393
3394
3395
3396
3397
3398
3399
3400
3401
3402
3403
3404
3405
3406
3407
3408
3409
3410
3411
3412
3413
3414
3415
3416
3417
3418
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3479
3480
3481
3482
3483
3484
3485
3486
3487
3488
3489
3490
3491
3492
3493
3494
3495
3496
3497
3498
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578
3579
3580
3581
3582
3583
3584
3585
3586
3587
3588
3589
3590
3591
3592
3593
3594
3595
3596
3597
3598
3599
3600
3601
3602
3603
3604
3605
3606
3607
3608
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619
3620
3621
3622
3623
3624
3625
3626
3627
3628
3629
3630
3631
3632
3633
3634
3635
3636
3637
3638
3639
3640
3641
3642
3643
3644
3645
3646
3647
3648
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658
3659
3660
3661
3662
3663
3664
3665
3666
3667
3668
3669
3670
3671
3672
3673
3674
3675
3676
3677
3678
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3699
3700
3701
3702
3703
3704
3705
3706
3707
3708
3709
3710
3711
3712
3713
3714
3715
3716
3717
3718
3719
3720
3721
3722
3723
3724
3725
3726
3727
3728
3729
3730
3731
3732
3733
3734
3735
3736
3737
3738
3739
3740
3741
3742
3743
3744
3745
3746
3747
3748
3749
3750
3751
3752
3753
3754
3755
3756
3757
3758
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3779
3780
3781
3782
3783
3784
3785
3786
3787
3788
3789
3790
3791
3792
3793
3794
3795
3796
3797
3798
3799
3800
3801
3802
3803
3804
3805
3806
3807
3808
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3850
3851
3852
3853
3854
3855
3856
3857
3858
3859
3860
3861
3862
3863
3864
3865
3866
3867
3868
3869
3870
3871
3872
3873
3874
3875
3876
3877
3878
3879
3880
3881
3882
3883
3884
3885
3886
3887
3888
3889
3890
3891
3892
3893
3894
3895
3896
3897
3898
3899
3900
3901
3902
3903
3904
3905
3906
3907
3908
3909
3910
3911
3912
3913
3914
3915
3916
3917
3918
3919
3920
3921
3922
3923
3924
3925
3926
3927
3928
3929
3930
3931
3932
3933
3934
3935
3936
3937
3938
3939
3940
3941
3942
3943
3944
3945
3946
3947
3948
3949
3950
3951
3952
3953
3954
3955
3956
3957
3958
3959
3960
3961
3962
3963
3964
3965
3966
3967
3968
3969
3970
3971
3972
3973
3974
3975
3976
3977
3978
3979
3980
3981
3982
3983
3984
3985
3986
3987
3988
3989
3990
3991
3992
3993
3994
3995
3996
3997
3998
3999
4000
```

```
/* tell VMS that PA menu is exited */
FUNCTION PAC_snd_vms_pac_deactivated
/* send PRH_PA_FAILURE to the BPRH menu handler
FUNCTION PAC_snd_bprh_mh_prh_pa_failure
<PAC_wait_failure_ack>

TRANS

EVENT ANY_MSG_FAILURE /* TAG reported failure */
/* find out which TAG failed */
FUNCTION PAC_chk_tag_fail
<PAC_pre_zip_chk_tag>

TRANS

EVENT ANY_FORCE_COS
/* cancel timer */
FUNCTION PAC_cancel_timer
/* idle the zip TAG port */
FUNCTION PAC_snd_tag_ldh_idle_tone
/* deallocate the DAS message memory */
FUNCTION PAC_das_mem_dealloc_pa_or_tmp_sys_msg_num
/* free up the PA/temporary system msg # */
FUNCTION PAC_free_pa_or_tmp_sys_msg_num
/* if there is a PAC currently uploading a message, */
/* tell it that PAC is going away */
FUNCTION PAC_snd_pdh_ldh_idle_if_upload
/* idle the TAG port */
FUNCTION PAC_snd_tag_ldh_idle
/* deallocate DAS memory for PA messages */
FUNCTION PAC_dealloc_msgs_on_signout
/* check if RC had changed Staff table after signed in */
FUNCTION PAC_chk_staff_chg_after_signin
<PAC_pos>

TRANS PAC_pre_zip_playing.tde

DONE
```

PAC_pre_zip_playing.tde

DONE

2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
299

92/11/09
12:16:06

/tmp/59201

28

2103204

```
2393
2394
2395
2396
2397 /*****
2398 *
2399 *   PAC_pre_zip_chk_tag
2400 *
2401 *   Description:   This transition state is waiting to determine
2402 *                  which TAG port failed while pre-zip tone was playing.
2403 *
2404 *   Resources Active: PA TAG port
2405 *                  zip TAG port.
2406 *                  Pseudo port (PAD) may be active, for uploading.
2407 *
2408 *   Critical Dependencies:
2409 *                  none.
2410 *
2411 *****/
2412
2413
2414
2415 STATE   PAC_pre_zip_chk_tag
2416
2417 EVENT   PAC_pre_zip_chk_tag FAIL /* PA TAG port failed */
2418
2419 FUNCTION /* idle the zip TAG port */
2420
2421 FUNCTION /* deallocate the DAS message memory */
2422
2423 FUNCTION /* free up the PA/temporary system msg # */
2424
2425 FUNCTION /* free up the PA/temporary system msg # */
2426
2427 FUNCTION /* if PAD is doing upload currently */
2428
2429 FUNCTION /* tell it that PAC is going away. */
2430
2431 FUNCTION /* NOTE: there may be multiple PADs doing uploads */
2432
2433 FUNCTION /* PAC_snd_pad_ldh_idle if upload
2434           /* deallocate DAS memory for PA messages */
2435
2436 FUNCTION /* PAC_dealloc_msgs_on_signout
2437           /* call VMS that PA menu is exited */
2438
2439 FUNCTION /* PAC_snd_vms_pac_deactivated
2440           /* send PRH_PA_FAILURE to the BPPH menu handler */
2441
2442 FUNCTION /* PAC_snd_bpph_mh_prh_pa_failure
2443           <PAC_wait_failure_ack>
2444
2445 TRANS   14
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3090
3091
3092
3093
3094
3095
3096
3097
3098
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3110
3111
3112
3113
3114
3115
3116
3117
3118
3119
3120
3121
3122
3123
3124
3125
3126
3127
3128
3129
3130
3131
3132
3133
3134
3135
3136
3137
3138
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3150
3151
3152
3153
3154
3155
3156
3157
3158
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3190
3191
3192
3193
3194
3195
3196
3197
3198
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3210
3211
3212
3213
3214
3215
3216
3217
3218
3219
3220
3221
3222
3223
3224
3225
3226
3227
3228
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238
3239
3240
3241
3242
3243
3244
3245
3246
3247
3248
3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269
3270
3271
3272
3273
3274
3275
3276
3277
3278
3279
3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3290
3291
3292
3293
3294
3295
3296
3297
3298
3299
3300
3301
3302
3303
3304
3305
3306
3307
3308
3309
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388
3389
3390
3391
3392
3393
3394
3395
3396
3397
3398
3399
3400
3401
3402
3403
3404
3405
3406
3407
3408
3409
3410
3411
3412
3413
3414
3415
3416
3417
3418
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3479
3480
3481
3482
3483
3484
3485
3486
3487
3488
3489
3490
3491
3492
3493
3494
3495
3496
3497
3498
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578
3579
3580
3581
3582
3583
3584
3585
3586
3587
3588
3589
3590
3591
3592
3593
3594
3595
3596
3597
3598
3599
3600
3601
3602
3603
3604
3605
3606
3607
3608
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619
3620
3621
3622
3623
3624
3625
3626
3627
3628
3629
3630
3631
3632
3633
3634
3635
3636
3637
3638
3639
3640
3641
3642
3643
3644
3645
3646
3647
3648
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658
3659
3660
3661
3662
3663
3664
3665
3666
3667
3668
3669
3670
3671
3672
3673
3674
3675
3676
3677
3678
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3699
3700
3701
3702
3703
3704
3705
3706
3707
3708
3709
3710
3711
3712
3713
3714
3715
3716
3717
3718
3719
3720
3721
3722
3723
3724
3725
3726
3727
3728
3729
3730
3731
3732
3733
3734
3735
3736
3737
3738
3739
3740
3741
3742
3743
3744
3745
3746
3747
3748
3749
3750
3751
3752
3753
3754
3755
3756
3757
3758
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3779
3780
3781
3782
3783
3784
3785
3786
3787
3788
3789
3790
3791
3792
3793
3794
3795
3796
3797
3798
3799
3800
3801
3802
3803
3804
3805
3806
3807
3808
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3850
3851
3852
3853
3854
3855
3856
3857
3858
3859
3860
3861
3862
3863
3864
3865
3866
3867
3868
3869
3870
3871
3872
3873
3874
3875
3876
3877
3878
3879
3880
3881
3882
3883
3884
3885
3886
3887
3888
3889
3890
3891
3892
3893
3894
3895
3896
3897
3898
3899
3900
3901
3902
3903
3904
3905
3906
3907
3908
3909
3910
3911
3912
3913
3914
3915
3916
3917
3918
3919
3920
3921
3922
3923
3924
3925
3926
3927
3928
3929
3930
3931
3932
3933
3934
3935
3936
3937
3938
3939
3940
3941
3942
3943
3944
3945
3946
3947
3948
3949
3950
3951
3952
3953
3954
3955
3956
3957
3958
3959
3960
3961
3962
3963
3964
3965
3966
3967
3968
3969
3970
3971
3972
3973
3974
3975
3976
3977
3978
3979
3980
3981
3982
3983
3984
3985
3986
3987
3988
3989
3990
3991
3992
3993
3994
3995
3996
3997
3998
3999
4000
4001
4002
4003
4004
4005
4006
4007
4008
4009
4010
4011
4012
4013
4014
4015
4016
4017
4018
4019
4020
4021
4022
4023
4024
4025
4026
4027
4028
4029
4030
4031
4032
4033
4034
4035
4036
4037
4038
4039
4040
4041
4042
4043
4044
4045
4046
4047
4048
4049
4050
4051
4052
4053
4054
4055
4056
4057
4058
4059
4060
4061
4062
4063
4064
4065
4066
4067
4068
4069
4070
4071
4072
4073
4074
4075
4076
4077
4078
4079
4080
4081
4082
4083
4084
4085
4086
4087
4088
4089
4090
4091
4092
4093
4094
4095
4096
4097
4098
4099
4100
4101
4102
4103
4104
4105
4106
4107
4108
4109
4110
4111
4112
4113
4114
4115
4116
4117
4118
4119
4120
4121
4122
4123
4124
4125
4126
4127
4128
4129
4130
4131
4132
4133
4134
4135
4136
4137
4138
4139
4140
4141
4142
4143
4144
4145
4146
4147
4148
4149
4150
4151
4152
4153
4154
4155
4156
4157
4158
4159
4160
4161
4162
4163
4164
4165
4166
4167
4168
4169
4170
4171
4172
4173
4174
4175
4176
4177
4178
4179
4180
4181
4182
4183
4184
4185
4186
4187
4188
4189
4190
4191
4192
4193
4194
4195
4196
4197
4198
4199
4200
4201
4202
4203
4204
4205
4206
4207
4208
4209
4210
4211
4212
4213
4214
4215
4216
4217
4218
4219
4220
4221
4222
4223
4224
4225
4226
4227
4228
4229
4230
4231
4232
4233
4234
4235
4236
4237
4238
4239
4240
4241
4242
4243
4244
4245
4246
4247
4248
4249
4250
4251
4252
4253
4254
4255
4256
4257
4258
4259
4260
4261
4262
4263
4264
4265
4266
4267
4268
4269
4270
4271
4272
4273
4274
4275
4276
4277
4278
4279
4280
4281
4282
4283
4284
4285
4286
4287
4288
4289
4290
4291
4292
4293
4294
4295
4296
4297
4298
4299
4300
4301
4302
4303
4304
4305
4306
4307
4308
4309
4310
4311
4312
4313
4314
4315
4316
4317
4318
4319
4320
4321
4322
4323
4324
4325
4326
4327
4328
4329
4330
4331
4332
4333
4334
4335
4336
4337
4338
4339
4340
4341
4342
4343
4344
4345
4346
4347
4348
4349
4350
4351
4352
4353
4354
4355
4356
4357
4358
4359
4360
4361
4362
4363
4364
4365
4366
4367
4368
4369
4370
4371
4372
4373
4374
4375
4376
4377
4378
4379
4380
4381
4382
4383
4384
4385
4386
4387
4388
4389
4390
4391
4392
4393
4394
4395
4396
4397
4398
4399
4400
4401
4402
4403
4404
4405
4406
4407
4408
4409
4410
4411
4412
4413
4414
4415
4416
4417
4418
4419
4420
4421
4422
4423
4424
4425
4426
4427
4428
4429
4430
4431
4432
4433
4434
4435
4436
4437
4438
4439
4440
4441
4442
4443
4444
4445
4446
4447
4448
4449
4450
4451
4452
4453
4454
4455
4456
4457
4458
4459
4460
4461
4462
4463
4464
4465
4466
4467
4468
4469
4470
4471
4472
4473
4474
4475
4476
4477
4478
4479
4480
4481
4482
4483
4484
4485
4486
4487
4488
4489
4490
4491
4492
4493
4494
4495
4496
4497
4498
4499
4500
4501
4502
4503
4504
4505
4506
4507
4508
4509
4510
4511
4512
4513
4514
4515
4516
4517
4518
4519
4520
4521
4522
4523
4524
4525
4526
4527
4528
4529
4530
4531
4532
4533
4534
4535
4536
4537
4538
4539
4540
4541
4542
4543
4544
4545
4546
4547
4548
4549
4550
4551
4552
4553
4554
4555
4556
4557
4558
4559
4560
4561
4562
4563
4564
4565
4566
4567
4568
4569
4570
4571
4572
4573
4574
4575
4576
4577
4578
4579
4580
4581
4582
4583
4584
4585
4586
4587
4588
4589
4590
4591
4592
4593
4594
4595
4596
4597
4598
4599
4600
4601
4602
4603
4604
4605
4606
4607
4608
4609
4610
4611
4612
4613
4614
4615
4616
4617
4618
4619
4620
4621
4622
4623
4624
4625
4626
4627
4628
4629
4630
4631
4632
4633
4634
4635
4636
4637
4638
4639
4640
4641
4642
4643
4644
4645
4646
4647
4648
4649
4650
4651
4652
4653
4654
4655
4656
4657
4658
4659
4660
4661
4662
4663
4664
4665
4666
4667
4668
4669
4670
4671
4672
4673
4674
4675
4676
4677
4678
4679
4680
4681
4682
4683
4684
4685
4686
4687
4688
4689
4690
4691
4692
4693
4694
4695
4696
4697
4698
4699
4700
4701
4702
4703
4704
4705
4706
4707
4708
4709
4710
4711
4712
4713
4714
4715
4716
4717
4718
4719
4720
4721
4722
4723
4724
4725
4726
4727
4728
4729
4730
4731
4732
4733
4734
4735
4736
4737
4738
4739
4740
4741
4742
4743
4744
4745
4746
4747
4748
4749
4750
4751
4752
4753
4754
4755
4756
4757
4758
4759
4760
4761
4762
4763
4764
4765
4766
4767
4768
4769
4770
4771
4772
4773
4774
4775
4776
4777
4778
47
```

92/EL/09
12:16:06

/tmp/59201

20

```
2460
2461
2462 /*****
2463 *
2464 * PAC_recording
2465 *
2466 * Description: In this state PA message recording is in progress.
2467 *
2468 * Resources Active: TAG port
2469 *
2470 * Pseudo Port (PAD) may be active, for uploading.
2471 *
2472 * Critical Dependencies:
2473 * none.
2474 *
2475 *****
2476
2477
2478 STATE PAC_recording
2479
2480 EVENT PAC_STOP_RECORD /* agent stopped recording */
2481
2482 FUNCTION /* tell TAG LDH to stop recording */
2483 <PAC_stop_record>
2484
2485 EVENT ANY_EXCEED_MSG_SIZE /* recording exceeded the max limit */
2486
2487 FUNCTION /* inform BPRH of it */
2488 PAC_snd_bprh_mh any_exceed_msg_size
2489
2490 FUNCTION /* check if EOM tone TAG is defined */
2491 <PAC_wt_post_eom_tag>
2492
2493 PAC_RECORD_EXIT /* delete the message */
2494
2495 FUNCTION /* tell TAG LDH to stop recording */
2496 <PAC_abort_record>
2497
2498
2499 EVENT ANY_SDC_FAILURE /* Uploading PAD reported failure */
2500
2501 FUNCTION /* tell TAG LDH to stop recording */
2502 PAC_snd_tag_ldh_stop_record
2503
2504 FUNCTION /* clear the sys msg # from PA tbl associated with PAD */
2505 PAC_clear_upload_fail_pad_sys_msg_num
2506
2507 FUNCTION /* deallocate the DAS message memory */
2508 PAC_das_mem_dealloc_pa_or_tmp_sys_msg_num
2509
2510 FUNCTION /* free up the PA/temporary system msg # */
2511 PAC_free_pa_or_tmp_sys_msg_num
2512
2513 FUNCTION /* idle the TAG port */
2514 PAC_snd_tag_ldh_idle
2515
2516 FUNCTION /* if there is a PAD currently uploading a message, */
2517 /* tell it that PAC is going away. */
2518 /* NOTE: there may be multiple PADs doing uploads */
2519 PAC_snd_pad_ldh_idle_if_upload
2520
2521 FUNCTION /* deallocate DAS memory for PA messages */
2522 PAC_dealloc_msgs_on_signout
2523
2524 FUNCTION /* tell VMS that PA menu is exited */
2525 PAC_snd_vms_pac_deactivated
2526
2527 FUNCTION /* send BPRH_PA_FAILURE to the BPRH menu handler */
2528 PAC_snd_bprh_mh_prh_pa_failure
2529
2530 <PAC_wait_failure_ack>
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
```

```
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555

/* free up the PA/temporary system msg # */
PAC_free_pa_or_tmp_sys_msg_num
/* If there is a PAD currently uploading a mes. #, */
/* tell it that PAC is going away. */
/* NOTE: there may be multiple PADs doing uploads */
PAC_snd_pad_ldh_idle_if_upload
/* deallocate DAS memory for PA messages */
PAC_dealloc_msgs_on_signout
/* tell VMS that PA menu is exited */
PAC_snd_vms_pac_deactivated
/* send BPRH_PA_FAILURE to the BPRH menu handler */
PAC_snd_bprh_mh_prh_pa_failure
<PAC_wait_failure_ack>

TRANS
EVENT
ANY_FORCE_OOS
/* deallocate the DAS message memory */
PAC_das_mem_dealloc_pa_or_tmp_sys_msg_num
/* free up the PA/temporary system msg # */
PAC_free_pa_or_tmp_sys_msg_num
/* If there is a PAD currently uploading a message, */
/* tell it that PAC is going away. */
PAC_snd_pad_ldh_idle_if_upload
/* idle the TAG port */
PAC_snd_tag_ldh_idle
/* deallocate DAS memory for PA messages */
PAC_dealloc_msgs_on_signout
/* check if PC had changed Staff table after signed in */
PAC_chk_staff_chg_after_signin

TRANS
PAC_recording.tde
```

2103204

```
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555

/* free up the PA/temporary system msg # */
PAC_free_pa_or_tmp_sys_msg_num
/* If there is a PAD currently uploading a mes. #, */
/* tell it that PAC is going away. */
/* NOTE: there may be multiple PADs doing uploads */
PAC_snd_pad_ldh_idle_if_upload
/* deallocate DAS memory for PA messages */
PAC_dealloc_msgs_on_signout
/* tell VMS that PA menu is exited */
PAC_snd_vms_pac_deactivated
/* send BPRH_PA_FAILURE to the BPRH menu handler */
PAC_snd_bprh_mh_prh_pa_failure
<PAC_wait_failure_ack>

TRANS
EVENT
ANY_MSG_FAILURE /* TAG reported failure */
/* deallocate the DAS message memory */
PAC_das_mem_dealloc_pa_or_tmp_sys_msg_num
FUNCTION
```

/tmp/59201

```

1011 2619 FUNCTION      PAC_dealloc_msgs_on_signout
1012 2620 /* tell VMS that PA menu is exited */
1013 2621 FUNCTION      PAC_snd_vms_pac_deactivated
1014 2622 /* send PRH_PA_FAILURE to the BPPH menu handler */
1015 2623 FUNCTION      PAC_snd_bprh_mh_prh_pa_failure
1016 2624 <PAC_wait_failure_ack>
1017 2625
1018 2626 EVENT
1019 2627
1020 2628 FUNCTION      /* deallocate the DAS message memory */
1021 2629 PAC_das_mem_dealloc_pa_or_tmp_sys_msg_num
1022 2630 /* free up the PA/temporary system msg # */
1023 2631 FUNCTION      PAC_free_pa_or_tmp_sys_msg_num
1024 2632 /* if there is a PAD currently uploading a message, */
1025 2633 /* tell it that PAC is going away */
1026 2634 FUNCTION      PAC_snd_pad_ldh_idle_if_upload
1027 2635 /* idle the TAG port */
1028 2636 FUNCTION      PAC_snd_tag_ldh_idle
1029 2637 /* deallocate DAS memory for PA messages */
1030 2638 FUNCTION      PAC_dealloc_msgs_on_signout
1031 2639 /* check if PC had changed Staff table after signed in */
1032 2640 FUNCTION      PAC_chk_staff_chg_after_signin
1033 2641 <PAC_pos>
1034 2642
1035 2643 PAC_step_record:ide
1036 2644
1037 2645
1038 2646
1039 2647
1040 2648
1041 2649
1042 2650
1043 2651
1044 2652
1045 2653
1046 2654
1047 2655
1048 2656
1049 2657
1050 2658
1051 2659
1052 2660
1053 2661
1054 2662
1055 2663
1056 2664
1057 2665
1058 2666
1059 2667
1060 2668
1061 2669
1062 2670
1063 2671
1064 2672
1065 2673
1066 2674
1067 2675
1068 2676
1069 2677
1070 2678
1071 2679
1072 2680
1073 2681
1074 2682
1075 2683
1076 2684
1077 2685
1078 2686
1079 2687
1080 2688
1081 2689
1082 2690
1083 2691
1084 2692
1085 2693
1086 2694
1087 2695
1088 2696
1089 2697
1090 2698
1091 2699
1092 2700
1093 2701
1094 2702
1095 2703
1096 2704
1097 2705
1098 2706
1099 2707
1100 2708
1101 2709
1102 2710
1103 2711
1104 2712
1105 2713
1106 2714
1107 2715
1108 2716
1109 2717
1110 2718
1111 2719
1112 2720
1113 2721
1114 2722
1115 2723
1116 2724
1117 2725
1118 2726
1119 2727
1120 2728
1121 2729
1122 2730
1123 2731
1124 2732
1125 2733
1126 2734
1127 2735
1128 2736
1129 2737
1130 2738
1131 2739
1132 2740
1133 2741
1134 2742
1135 2743
1136 2744
1137 2745
1138 2746
1139 2747
1140 2748
1141 2749
1142 2750
1143 2751
1144 2752
1145 2753
1146 2754
1147 2755
1148 2756
1149 2757
1150 2758
1151 2759
1152 2760
1153 2761
1154 2762
1155 2763
1156 2764
1157 2765
1158 2766
1159 2767
1160 2768
1161 2769
1162 2770
1163 2771
1164 2772
1165 2773
1166 2774
1167 2775
1168 2776
1169 2777
1170 2778
1171 2779
1172 2780
1173 2781
1174 2782
1175 2783
1176 2784
1177 2785
1178 2786
1179 2787
1180 2788
1181 2789
1182 2790
1183 2791
1184 2792
1185 2793
1186 2794
1187 2795
1188 2796
1189 2797
1190 2798
1191 2799
1192 2800
1193 2801
1194 2802
1195 2803
1196 2804
1197 2805
1198 2806
1199 2807
1200 2808
1201 2809
1202 2810
1203 2811
1204 2812
1205 2813
1206 2814
1207 2815
1208 2816
1209 2817
1210 2818
1211 2819
1212 2820
1213 2821
1214 2822
1215 2823
1216 2824
1217 2825
1218 2826
1219 2827
1220 2828
1221 2829
1222 2830
1223 2831
1224 2832
1225 2833
1226 2834
1227 2835
1228 2836
1229 2837
1230 2838
1231 2839
1232 2840
1233 2841
1234 2842
1235 2843
1236 2844
1237 2845
1238 2846
1239 2847
1240 2848
1241 2849
1242 2850
1243 2851
1244 2852
1245 2853
1246 2854
1247 2855
1248 2856
1249 2857
1250 2858
1251 2859
1252 2860
1253 2861
1254 2862
1255 2863
1256 2864
1257 2865
1258 2866
1259 2867
1260 2868
1261 2869
1262 2870
1263 2871
1264 2872
1265 2873
1266 2874
1267 2875
1268 2876
1269 2877
1270 2878
1271 2879
1272 2880
1273 2881
1274 2882
1275 2883
1276 2884
1277 2885
1278 2886
1279 2887
1280 2888
1281 2889
1282 2890
1283 2891
1284 2892
1285 2893
1286 2894
1287 2895
1288 2896
1289 2897
1290 2898
1291 2899
1292 2900
1293 2901
1294 2902
1295 2903
1296 2904
1297 2905
1298 2906
1299 2907
1300 2908
1301 2909
1302 2910
1303 2911
1304 2912
1305 2913
1306 2914
1307 2915
1308 2916
1309 2917
1310 2918
1311 2919
1312 2920
1313 2921
1314 2922
1315 2923
1316 2924
1317 2925
1318 2926
1319 2927
1320 2928
1321 2929
1322 2930
1323 2931
1324 2932
1325 2933
1326 2934
1327 2935
1328 2936
1329 2937
1330 2938
1331 2939
1332 2940
1333 2941
1334 2942
1335 2943
1336 2944
1337 2945
1338 2946
1339 2947
1340 2948
1341 2949
1342 2950
1343 2951
1344 2952
1345 2953
1346 2954
1347 2955
1348 2956
1349 2957
1350 2958
1351 2959
1352 2960
1353 2961
1354 2962
1355 2963
1356 2964
1357 2965
1358 2966
1359 2967
1360 2968
1361 2969
1362 2970
1363 2971
1364 2972
1365 2973
1366 2974
1367 2975
1368 2976
1369 2977
1370 2978
1371 2979
1372 2980
1373 2981
1374 2982
1375 2983
1376 2984
1377 2985
1378 2986
1379 2987
1380 2988
1381 2989
1382 2990
1383 2991
1384 2992
1385 2993
1386 2994
1387 2995
1388 2996
1389 2997
1390 2998
1391 2999
1392 3000

```

2103204

92/11/09
90:91:71
17:16:06



2103204

[illegible]

92/11/09
12:16:06

/tmp/59201

32

2103204

```
2733 *****
2734 PAC_wt_post_eom_tag
2735
2736
2737 Description: This state is waiting for the check
2738 to determine whether EOM tone TAG group is defined,
2739 to apply the end of recording zip tone
2740
2741 Resources Active: PA TAG port
2742 Pseudo port (PAD) may be active, for uploading.
2743
2744 Critical Dependencies:
2745 none.
2746
2747 *****
2748
2749
2750
2751 STATE PAC_wt_post_eom_tag
2752
2753
2754
2755 EVENT PAC_wt_tag_id /* no EOM tone TAG group defined */
2756
2757 TRANS <PAC_sel_record_opt>
2758
2759
2760
2761 EVENT PAC_wt_tag_id /* valid EOM tone TAG group */
2762
2763 FUNCTION PAC_set_eom_tone_timer
2764
2765 FUNCTION /* get eom tone TAG port */
2766
2767 TRANS <PAC_start_post_eom>
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780 DONE PAC_wt_post_eom_tag.tde
2781
2782
```

```
2783
2784
2785
2786 *****
2787 PAC_start_post_eom
2788
2789
2790 Description: This state is waiting for the zip tone TAG port
2791 for applying post-recording zip.
2792
2793 Resources Active: TAG port
2794 Pseudo port (PAD) may be active, for uploading.
2795
2796 Critical Dependencies:
2797 none.
2798
2799 *****
2800
2801
2802
2803 STATE PAC_start_post_eom
2804
2805
2806
2807 EVENT ANY_ACTIVATE_ACK /* zip tone started */
2808
2809 FUNCTION PAC_save_tone_tag_port_id
2810
2811 TRANS <PAC_post_eom_playing>
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
```

```
1073 STATE PAC_start_post_eom
1074
1075
1076
1077 EVENT ANY_ACTIVATE_NACK /* no zip tone */
1078
1079 FUNCTION /* cancel timer */
1080
1081 TRANS <PAC_sel_record_opt>
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3090
3091
3092
3093
3094
3095
3096
3097
3098
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3110
3111
3112
3113
3114
3115
3116
3117
3118
3119
3120
3121
3122
3123
3124
3125
3126
3127
3128
3129
3130
3131
3132
3133
3134
3135
3136
3137
3138
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3150
3151
3152
3153
3154
3155
3156
3157
3158
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3190
3191
3192
3193
3194
3195
3196
3197
3198
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3210
3211
3212
3213
3214
3215
3216
3217
3218
3219
3220
3221
3222
3223
3224
3225
3226
3227
3228
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238
3239
3240
3241
3242
3243
3244
3245
3246
3247
3248
3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269
3270
3271
3272
3273
3274
3275
3276
3277
3278
3279
3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3290
3291
3292
3293
3294
3295
3296
3297
3298
3299
3300
3301
3302
3303
3304
3305
3306
3307
3308
3309
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338
3339

```

92111809
12:16:06

/tmp/59201

```
2846
2847
2848
1097 2849
2850
1098 2851
2852
1099 2853
2854
1100 2855
1101 2856
2857
1102 2858
2859
1103 2860
2861
1104 2862
2863
1105 2864
2865
1106 2866
2867
2868
2869
1107 2870
2871
1108 2872
2873
1109 2874
2875
1110 2876
1111 2877
2878
1112 2879
2880
1113 2881
2882
1114 2883
2884
1115 2885
2886
1116 2887
2888
89
1117 2890
2891
1118 2892
2893
1119 2894
2895
1120 2896
1121 2897
2898
1122 2899
2900
1123 2901
2902
1124 2903 DONE
2904
1125 2905

/* If there is a PAD currently uploading a message, */
/* tell it that PAC is going away. */
/* NOTE: there may be multiple PADs doing uploads */
FUNCTION PAC_snd_pad_ldh_idle_if_upload
/* deallocate DAS memory for PA messages */
FUNCTION PAC_dealloc_msgs_on_signout
/* tell VMS that PA menu is exited */
FUNCTION PAC_snd_vms_pac_deactivated
/* send PRH_PA_FAILURE to the BPRH menu handler */
FUNCTION PAC_snd_oprh_mb_prh_pa_failure
<PAC_wait_failure_ack>

TRANS

EVENT ANY_MSG_FAILURE /* TAG reported failure */
/* cancel timer */
FUNCTION PAC_cancel_timer
/* cancel zip port request */
FUNCTION PAC_snd_tag_ldh_cancel_acquire_zip
/* deallocate the DAS message memory */
FUNCTION PAC_das_mem_dealloc_pa_or_tmp_sys_msg_num
/* free up the PA/temporary system msg # */
FUNCTION PAC_free_pa_or_tmp_sys_msg_num
/* If there is a PAD currently uploading a message, */
/* tell it that PAC is going away. */
/* NOTE: there may be multiple PADs doing uploads */
FUNCTION PAC_snd_pad_ldh_idle_if_upload
/* deallocate DAS memory for PA messages */
FUNCTION PAC_dealloc_msgs_on_signout
/* tell VMS that PA menu is exited */
FUNCTION PAC_snd_vms_pac_deactivated
/* send PRH_PA_FAILURE to the BPRH menu handler */
FUNCTION PAC_snd_oprh_mb_prh_pa_failure
<PAC_wait_failure_ack>

TRANS

EVENT ANY_FORCE_OOS
/* cancel timer */
FUNCTION PAC_cancel_timer
/* cancel zip port request */
FUNCTION PAC_snd_tag_ldh_cancel_acquire_zip
/* deallocate the DAS message memory */
FUNCTION PAC_das_mem_dealloc_pa_or_tmp_sys_msg_num
/* free up the PA/temporary system msg # */
FUNCTION PAC_free_pa_or_tmp_sys_msg_num
/* If there is a PAD currently uploading a message, */
/* tell it that PAC is going away */
FUNCTION PAC_snd_pad_ldh_idle_if_upload
/* idle the TAG port */
FUNCTION PAC_snd_tag_ldh_idle
/* deallocate DAS memory for PA messages */
FUNCTION PAC_dealloc_msgs_on_signout
/* check if RC had changed Staff table after signed in */
FUNCTION PAC_chk_staff_chg_after_signin
<PAC_oos>

TRANS

EVENT PAC_STOP_RECORD /* we can get this, because of race-condition */
/* eat the event */
TRANS <PAC_start_post_eom>

PAC_start_post_eom.tde
1124 2903 DONE
2904
1125 2905

*****
PAC_post_eom_playing
*****
Description: In this state the post-recording eom tone is playing.
Resources Active: PA TAG port
ECM TAG port.
Pseudo port (PAD) may be active, for uploading.
Critical Dependencies:
none.
*****
***** STATE PAC_post_eom_playing
*****
EVENT PAC_PLAY_EOM_TIMEOUT /* done playing eom tone */
FUNCTION PAC_clear_timer_id
/* idle the eom TAG port */
FUNCTION PAC_snd_tag_ldh_idle_tone
<PAC_sel_record_opt>

TRANS

EVENT PAC_RECORD_EXIT /* agent exited the PA menu */
/* cancel timer */
FUNCTION PAC_cancel_timer
/* idle the eom TAG port */
FUNCTION PAC_snd_tag_ldh_idle_tone
/* deallocate the DAS message memory */
FUNCTION PAC_das_mem_dealloc_pa_or_tmp_sys_msg_num
/* free up the PA/temporary system msg # */
FUNCTION PAC_free_pa_or_tmp_sys_msg_num
<PAC_select_option>

TRANS

EVENT ANY_MSG_FAILURE /* PA or eom TAG reported failure */
/* cancel timer */
FUNCTION PAC_cancel_timer
/* find out which TAG failed */
FUNCTION PAC_chk_tag_fail
<PAC_post_eom_chk_tag>

TRANS

EVENT ANY_SDO_FAILURE /* Uploading PAD reported failure */
/* cancel timer */
FUNCTION PAC_cancel_timer
/* clear the sys msg # from PA tbl associated with PAD */
FUNCTION PAC_clear_upload_fail_pad_sys_msg_num
/* idle the eom TAG port */
FUNCTION PAC_snd_tag_ldh_idle_tone
/* deallocate the DAS message memory */
FUNCTION PAC_das_mem_dealloc_pa_or_tmp_sys_msg_num
/* free up the PA/temporary system msg # */
FUNCTION PAC_free_pa_or_tmp_sys_msg_num
/* If there is a PAD currently uploading a message, */
/* tell it that PAC is going away */
FUNCTION PAC_snd_pad_ldh_idle_if_upload
/* deallocate DAS memory for PA messages */
*****
```

2103204

92/11/09
12:16:06

/tmp/59201

```
3299
1275 3300
1276 3301
3302
1277 3303
3304
1278 3305
3306
1279 3307
3308
3309
1280 3310
3311
1281 3312
3313
1282 3314
3315
1283 3316
1284 17
1285 3319 DONE
3320
1286 3321

/* send BPRH_PA_FAILURE to the BPRH menu handler */
FUNCTION PAC_snd_bprh_mh_prh_pa_failure
<PAC_wait_failure_ack>

TRANS

EVENT ANY_FORCE_OOS

/* deallocate the DAS message memory */
FUNCTION PAC_das_mem_dealloc_pa_or_tmp_sys_msg_num
/* free up the PA/temporary system msg # */
FUNCTION PAC_free_pa_or_tmp_sys_msg_num
/* if there is a PAD currently uploading a message, */
/* tell it that PAC is going away */
FUNCTION PAC_snd_pad_ldh_idle_if_upload
/* idle the TAG port */
FUNCTION PAC_snd_tag_ldh_idle
/* deallocate DAS memory for PA messages */
FUNCTION PAC_dealloc_msgs_on_signout
/* check if RC had changed Staff table after signed in */
FUNCTION PAC_chk_staff_chg_after_signin
<PAC_oos>

TRANS

PAC_store_wait.cde
3320
1286 3321
```

```
*****
PAC_store_record
Description: This state is waiting for the acknowledgement from the
TAG LDH that the existing message has been replaced by
the newly recorded message.

Resources Active: TAG port
PAD port(s) may be active for uploading

Critical Dependencies:
none.
*****
```

```
1287 3344 STATE PAC_store_record
3345
1288 3346
3347
1289 3348
3349
1290 3350
3351
1291 3352
3353
1292 3354
3355
1293 3356
1294 3357
3358
1295 3359
3360
1296 3361
3362
1297 3363
1298 3364
3365
1299 3366
3367
1300 3368
3369
1301 3370
3371
1302 3372
3373
3374
3375
1303 3376
3377
1304 3378
3379
1305 3380
3381
1306 3382
3383
1307 3384

EVENT ANY_REPLACE_MSG_ACK /* message replaced */
/* copy the PA sys msg # into the PA table entry */
FUNCTION PAC_set_sys_msg_num_in_pa_tbl
/* update the PA sys msg # in the Staff table entry */
FUNCTION PAC_upd_sys_msg_no_in_staff
/* free up the temp sys msg # */
FUNCTION PAC_free_tmp_sys_msg_num
/* set "ready to play" in msg status */
FUNCTION PAC_set_msg_status_ready
/* acquire the pseudo (PAD) port for upload */
FUNCTION PAC_snd_sdo_lgh_acquire_pa
<PAC_sdc_upload>

TRANS

EVENT PAC_RECORD_EXIT /* agent wants to ABORT recording */
/* deallocate the DAS message memory */
FUNCTION PAC_das_mem_dealloc_pa_or_tmp_sys_msg_num
/* free up the PA/temporary system msg # */
FUNCTION PAC_free_pa_or_tmp_sys_msg_num
<PAC_select_option>

TRANS

EVENT ANY_SDO_FAILURE /* Uploading PAD reported failure */
/* clear the sys msg # from PA tbl associated with PAD */
FUNCTION PAC_clear_upload_fail_pac_sys_msg_num
/* deallocate the DAS message memory */
FUNCTION PAC_das_mem_dealloc_pa_or_tmp_sys_msg_num
/* free up the PA/temporary system msg # */
FUNCTION PAC_free_pa_or_tmp_sys_msg_num
/* if there is a PAD currently uploading a message, */
/* tell it that PAC is going away. */
/* NOTE: there may be multiple PADs doing uploads */
FUNCTION PAC_snd_pad_ldh_idle_if_upload
/* deallocate DAS memory for PA messages */
FUNCTION PAC_dealloc_msgs_on_signout
/* idle the TAG port */
FUNCTION PAC_snd_tag_ldh_idle
/* tell VMS that PA menu is exited */
FUNCTION PAC_snd_vms_pac_deactivated
/* send BPRH_PA_FAILURE to the BPRH menu handler */
FUNCTION PAC_snd_bprh_mh_prh_pa_failure
```

2103204

92/11/09
13:16:06

/tmp/59201

```
1308 3385 TRANS <PAC_wait_failure_ack>
1309 3386
1310 3387 EVENT ANY_MSG_FAILURE /* TAG reported failure */
1311 3388 FUNCTION PAC_das_mem_dealloc_pa_or_tmp_sys_msg_num
1312 3389 /* free up the PA/temporary system msg # */
1313 3390 FUNCTION PAC_free_pa_or_tmp_sys_msg_num
1314 3391 /* If there is a PAC currently uploading a message, */
1315 3392 /* tell it that PAC is going away. */
1316 3393 /* NOTE: there may be multiple PADS doing uploads */
1317 3394 /* tell it that PAC is going away. */
1318 3395 FUNCTION PAC_snd_pad_ldh_idle_if_upload
1319 3396 /* deallocate DAS memory for PA messages */
1320 3397 FUNCTION PAC_dealloc_msgs_on_signout
1321 3398 /* tell VMS that PA menu is exited */
1322 3399 FUNCTION PAC_snd_vms_pac_deactivated
1323 3400 /* send PPH_PA_FAILURE to the BPPH menu handler */
1324 3401 FUNCTION PAC_snd_pph_mh_prh_pa_failure
1325 3402 <PAC_wait_failure_ack>
1326 103
1327 3403 TRANS ANY_FORCE_OOS
1328 3404
1329 3405 FUNCTION /* deallocate the DAS message memory */
1330 3406 PAC_das_mem_dealloc_pa_or_tmp_sys_msg_num
1331 3407 /* free up the PA/temporary system msg # */
1332 3408 FUNCTION PAC_free_pa_or_tmp_sys_msg_num
1333 3409 /* If there is a PAC currently uploading a message, */
1334 3410 /* tell it that PAC is going away */
1335 3411 FUNCTION PAC_snd_pad_ldh_idle_if_upload
1336 3412 /* idle the TAG port */
1337 3413 FUNCTION PAC_snd_tag_ldh_idle
1338 3414 /* deallocate DAS memory for PA messages */
1339 3415 FUNCTION PAC_dealloc_msgs_on_signout
1340 3416 /* check if PC had changed Staff table after signed in */
1341 3417 FUNCTION PAC_chk_staff_chg_after_signin
1342 3418 <PAC_oos>
1343 3419
1344 3420 TRANS PAC_store_record.cde
1345 3421
1346 3422
```

```
3423
3424
3425
3426
3427 *****
3428 PAC_sdo_upload
3429
3430
3431 Description: This state is waiting for the pseudo (PAD) port for
3432 uploading the recorded message.
3433
3434 Resources Active: TAG port
3435 PAD port may be active for uploading
3436
3437 Critical Dependencies:
3438 none.
3439
3440 *****
3441 STATE PAC_sdo_upload
3442
3443
3444
3445
3446
3447
3448
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3479
3480
3481
3482
3483
3484
3485
3486
3487
3488
3489
3490
3491
3492
3493
3494
3495
3496
3497
3498
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578
3579
3580
3581
3582
3583
3584
3585
3586
3587
3588
3589
3590
3591
3592
3593
3594
3595
3596
3597
3598
3599
3600
3601
3602
3603
3604
3605
3606
3607
3608
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619
3620
3621
3622
3623
3624
3625
3626
3627
3628
3629
3630
3631
3632
3633
3634
3635
3636
3637
3638
3639
3640
3641
3642
3643
3644
3645
3646
3647
3648
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658
3659
3660
3661
3662
3663
3664
3665
3666
3667
3668
3669
3670
3671
3672
3673
3674
3675
3676
3677
3678
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3699
3700
3701
3702
3703
3704
3705
3706
3707
3708
3709
3710
3711
3712
3713
3714
3715
3716
3717
3718
3719
3720
3721
3722
3723
3724
3725
3726
3727
3728
3729
3730
3731
3732
3733
3734
3735
3736
3737
3738
3739
3740
3741
3742
3743
3744
3745
3746
3747
3748
3749
3750
3751
3752
3753
3754
3755
3756
3757
3758
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3779
3780
3781
3782
3783
3784
3785
3786
3787
3788
3789
3790
3791
3792
3793
3794
3795
3796
3797
3798
3799
3800
3801
3802
3803
3804
3805
3806
3807
3808
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3850
3851
3852
3853
3854
3855
3856
3857
3858
3859
3860
3861
3862
3863
3864
3865
3866
3867
3868
3869
3870
3871
3872
3873
3874
3875
3876
3877
3878
3879
3880
3881
3882
3883
3884
3885
3886
3887
3888
3889
3890
3891
3892
3893
3894
3895
3896
3897
3898
3899
3900
3901
3902
3903
3904
3905
3906
3907
3908
3909
3910
3911
3912
3913
3914
3915
3916
3917
3918
3919
3920
3921
3922
3923
3924
3925
3926
3927
3928
3929
3930
3931
3932
3933
3934
3935
3936
3937
3938
3939
3940
3941
3942
3943
3944
3945
3946
3947
3948
3949
3950
3951
3952
3953
3954
3955
3956
3957
3958
3959
3960
3961
3962
3963
3964
3965
3966
3967
3968
3969
3970
3971
3972
3973
3974
3975
3976
3977
3978
3979
3980
3981
3982
3983
3984
3985
3986
3987
3988
3989
3990
3991
3992
3993
3994
3995
3996
3997
3998
3999
4000
4001
4002
4003
4004
4005
4006
4007
4008
4009
4010
4011
4012
4013
4014
4015
4016
4017
4018
4019
4020
4021
4022
4023
4024
4025
4026
4027
4028
4029
4030
4031
4032
4033
4034
4035
4036
4037
4038
4039
4040
4041
4042
4043
4044
4045
4046
4047
4048
4049
4050
4051
4052
4053
4054
4055
4056
4057
4058
4059
4060
4061
4062
4063
4064
4065
4066
4067
4068
4069
4070
4071
4072
4073
4074
4075
4076
4077
4078
4079
4080
4081
4082
4083
4084
4085
4086
4087
4088
4089
4090
4091
4092
4093
4094
4095
4096
4097
4098
4099
4100
4101
4102
4103
4104
4105
4106
4107
4108
4109
4110
4111
4112
4113
4114
4115
4116
4117
4118
4119
4120
4121
4122
4123
4124
4125
4126
4127
4128
4129
4130
4131
4132
4133
4134
4135
4136
4137
4138
4139
4140
4141
4142
4143
4144
4145
4146
4147
4148
4149
4150
4151
4152
4153
4154
4155
4156
4157
4158
4159
4160
4161
4162
4163
4164
4165
4166
4167
4168
4169
4170
4171
4172
4173
4174
4175
4176
4177
4178
4179
4180
4181
4182
4183
4184
4185
4186
4187
4188
4189
4190
4191
4192
4193
4194
4195
4196
4197
4198
4199
4200
4201
4202
4203
4204
4205
4206
4207
4208
4209
4210
4211
4212
4213
4214
4215
4216
4217
4218
4219
4220
4221
4222
4223
4224
4225
4226
4227
4228
4229
4230
4231
4232
4233
4234
4235
4236
4237
4238
4239
4240
4241
4242
4243
4244
4245
4246
4247
4248
4249
4250
4251
4252
4253
4254
4255
4256
4257
4258
4259
4260
4261
4262
4263
4264
4265
4266
4267
4268
4269
4270
4271
4272
4273
4274
4275
4276
4277
4278
4279
4280
4281
4282
4283
4284
4285
4286
4287
4288
4289
4290
4291
4292
4293
4294
4295
4296
4297
4298
4299
4300
4301
4302
4303
4304
4305
4306
4307
4308
4309
4310
4311
4312
4313
4314
4315
4316
4317
4318
4319
4320
4321
4322
4323
4324
4325
4326
4327
4328
4329
4330
4331
4332
4333
4334
4335
4336
4337
4338
4339
4340
4341
4342
4343
4344
4345
4346
4347
4348
4349
4350
4351
4352
4353
4354
4355
4356
4357
4358
4359
4360
4361
4362
4363
4364
4365
4366
4367
4368
4369
4370
4371
4372
4373
4374
4375
4376
4377
4378
4379
4380
4381
4382
4383
4384
4385
4386
4387
4388
4389
4390
4391
4392
4393
4394
4395
4396
4397
4398
4399
4400
4401
4402
4403
4404
4405
4406
4407
4408
4409
4410
4411
4412
4413
4414
4415
4416
4417
4418
4419
4420
4421
4422
4423
4424
4425
4426
4427
4428
4429
4430
4431
4432
4433
4434
4435
4436
4437
4438
4439
4440
4441
4442
4443
4444
4445
4446
4447
4448
4449
4450
4451
4452
4453
4454
4455
4456
4457
4458
4459
4460
4461
4462
4463
4464
4465
4466
4467
4468
4469
4470
4471
4472
4473
4474
4475
4476
4477
4478
4479
4480
4481
4482
4483
4484
4485
4486
4487
4488
4489
4490
4491
4492
4493
4494
4495
4496
4497
4498
4499
4500
4501
4502
4503
4504
4505
4506
4507
4508
4509
4510
4511
4512
4513
4514
4515
4516
4517
4518
4519
4520
4521
4522
4523
4524
4525
4526
4527
4528
4529
4530
4531
4532
4533
4534
4535
4536
4537
4538
4539
4540
4541
4542
4543
4544
4545
4546
4547
4548
4549
4550
4551
4552
4553
4554
4555
4556
4557
4558
4559
4560
4561
4562
4563
4564
4565
4566
4567
4568
4569
4570
4571
4572
4573
4574
4575
4576
4577
4578
4579
4580
4581
4582
4583
4584
4585
4586
4587
4588
4589
4590
4591
4592
4593
4594
4595
4596
4597
4598
4599
4600
4601
4602
4603
4604
4605
4606
4607
4608
4609
4610
4611
4612
4613
4614
4615
4616
4617
4618
4619
4620
4621
4622
4623
4624
4625
4626
4627
4628
4629
4630
4631
4632
4633
4634
4635
4636
4637
4638
4639
4640
4641
4642
4643
4644
4645
4646
4647
4648
4649
4650
4651
4652
4653
4654
4655
4656
4657
4658
4659
4660
4661
4662
4663
4664
4665
4666
4667
4668
4669
4670
4671
4672
4673
4674
4675
4676
4677
4678
4679
4680
4681
4682
4683
4684
4685
4686
4687
4688
4689
4690
4691
4692
4693
4694
4695
4696
4697
4698
4699
4700
4701
4702
4703
4704
4705
4706
4707
4708
4709
4710
4711
4712
4713
4714
4715
4716
4717
4718
4719
4720
4721
4722
4723
4724
4725
4726
4727
4728
4729
4730
4731
4732
4733
4734
4735
4736
4737
4738
4739
4740
4741
4742
4743
4744
4745
4746
4747
4748
4749
4750
4751
4752
4753
4754
4755
4756
4757
4758
4759
4760
4761
4762
4763
4764
4765
4766
4767
4768
4769
4770
4771
4772
4773
4774
4775
4776
4777
4778
4779
4780
4781
4782
4783
4784
4785
4786
4787
4788
4789
4790
4791
4792
4793
4794
4795
4796
4797
4798
4799
4800
4801
4802
4803
4804
4805
4806
4807
4808
4809
4810
4811
4812
4813
4814
4815
4816
4817
4818
4819
4820
4821
4822
4823
4824
4825
4826
4827
4828
4829
4830
4831
4832
4833
4834
4835
4836
4837
4838
4839
4840
4841
4842
4843
4844
4845
4846
4847
4848
4849
4850
4851
4852
4853
4854
4855
4856
4857
4858
4859
4860
4861
4862
4863
4864
4865
4866
4867
4868
4869
4870
4871
4872
4873
4874
4875
4876
4877
4878
4879
4880
4881
4882
4883
4884
4885
4886
4887
4888
4889
4890
4891
4892
4893
4894
4895
4896
4897
4898
4899
4900
4901
4902
4903
4904
4905
4906
4907
4908
4909
4910
4911
4912
4913
4914
4915
4916
4917
4918
4919
4920
4921
4922
4923
4924
4925
4926
4927
4928
4929
4930
4931
4932
4933
4934
4935
4936
4937
4938
4939
4940
4941
4942
4943
4944
4945
4946
4947
4948
4949
4950
4951
4952
4953
4954
4955
4956
4957
4958
4959
4960
4961
4962
4963
4964
4965
4966
4967
4968
4969
4970
4971
4972
4973
4974
4975
4976
4977
4978
4979
4980
4981
4982
4983
4984
4985
4986
4987
4988
4989
4990
4991
4992
4993
4994
4995
4996
4997
4998
4999
5000
```

2103204

92/11/09
12:16:06

/tmp/59201

```
3486
1349 3487 EVENT ANY_MSG_FAILURE /* TAG reported failure */
1348 3488 FUNCTION /* cancel the pseudo port request */
1350 3489 /* deallocate the DAS message memory */
1351 3491 PAC_has_mem_dealloc_pa_or_tmp_sys_msg_num
13492 3493 /* free up the PA/temporary system msg # */
13494 3494 PAC_free_pa_or_tmp_sys_msg_num
13495 3495 /* If there is a PAD currently uploading a message, */
13496 3496 /* call it that PAC is going away. */
1353 3497 /* NOTE: there may be multiple PADs doing uploads */
13498 3498 PAC_snd_pad_ldh_idle_if_upload
1354 3499 /* deallocate DAS memory for PA messages */
1355 3501 PAC_dealloc_msgs_on_signout
13502 3502 /* tell VMS that PA menu is exited */
1355 3501 PAC_snd_vms_pac_deactivated
1356 3503 /* send PPH_PA_FAILURE to the BPRH menu handler */
13504 3504 PAC_snd_bprh_mh_prh_pa_failure
TRANS <PAC_wait_failure_ack>
3505
1358 3506 EVENT ANY_FORCE_OOS
1359 3507 /* cancel the pseudo port request */
1359 3508 PAC_snd_sdc_ldh_cancel_acquire
13509 3509 /* deallocate the DAS message memory */
1360 3510 PAC_has_mem_dealloc_pa_or_tmp_sys_msg_num
13511 3511 /* free up the PA/temporary system msg # */
1361 3512 PAC_free_pa_or_tmp_sys_msg_num
13513 3513 /* If there is a PAD currently uploading a message, */
13514 3514 /* call it that PAC is going away */
1362 3515 PAC_snd_pad_ldh_idle_if_upload
13516 3516 /* Idle the TAG port */
1363 3517 PAC_snd_tag_ldh_idle
13518 3518 /* deallocate DAS memory for PA messages */
1364 3519 PAC_dealloc_msgs_on_signout
13520 3520 /* check if PC had changed Staff table after signed in */
1365 3521 PAC_chk_staff_chg_after_signin
TRANS <PAC_oos>
1366 3522
1367 3524 DONE PAC_sdc_upload_end
13525 3525
1368 3526
```

```
3527
3528
3529
3530
3531 /*****
3532 *
3533 * PAC_play_record
3534 *
3535 * Description: In this state, a newly recorded PA message is being
3536 * reviewed by an agent.
3537 *
3538 * Resources Active: TAG port
3539 * PAD port may be active for uploading
3540 *
3541 * Critical Dependencies:
3542 * none.
3543 *
3544 *****/
3545
3546
3547
1369 3548 STATE PAC_play_record
1349 3549
1370 3550 EVENT ANY_END_OF_MESSAGE /* message finished playing */
1351 3551 FUNCTION /* get the EDM tone TAG id */
1371 3552 <PAC_wt_post_eom_tag> PAC_get_eom_tag_grip_id
1372 3553
1354 3554 PAC_PLAY_REQ /* agent wants to replay the current PA message */
1373 3555 FUNCTION /* call TAG LDH to play the message */
1374 3556 <PAC_play_record> PAC_snd_tag_ldh_play_rec
1375 3558
1359 3559 PAC_RECORD_REQ /* agent wants to record a message */
1376 3560 FUNCTION /* call TAG LDH to stop playing the message */
1377 3562 PAC_snd_tag_ldh_stop_play
1378 3564 FUNCTION /* deallocate the DAS message memory */
1379 3565 PAC_das_mem_dealloc_pa_or_tmp_sys_msg_num
1379 3566 /* allocate memory for new recording */
1380 3567 FUNCTION PAC_das_mem_alloc_pa_or_tmp_sys_msg_num
13568 3568 <PAC_record_option>
1381 3569
1382 3571 PAC_RECORD_EXIT /* agent exited the PA menu */
13570 3570 FUNCTION /* call TAG LDH to stop playing */
1382 3571 PAC_snd_tag_ldh_stop_play
13572 3572 FUNCTION /* deallocate the DAS message memory */
1382 3573 PAC_das_mem_dealloc_pa_or_tmp_sys_msg_num
13574 3574 /* free up the PA/temporary system msg # */
1384 3575 FUNCTION PAC_free_pa_or_tmp_sys_msg_num
1385 3576 <PAC_select_option>
13577 3577
1386 3578 EVENT PAC_STORE_REQ /* store and exit the PA menu */
13579 3579 FUNCTION /* call TAG LDH to stop playing */
1387 3580 PAC_snd_tag_ldh_stop_play
13581 3581 FUNCTION /* check if PAD is uploading this message */
1388 3582 PAC_chk_pad_upload_rec_msg
1389 3583 <PAC_store_chk>
13584 3584
13585 3585
1390 3586 EVENT PAC_STORE_EXIT /* store and exit the PA menu */
13587 3587 FUNCTION /* call TAG LDH to stop playing */
1391 3588 PAC_snd_tag_ldh_stop_play
13589 3589 FUNCTION /* check if PAD is uploading this message */
```

2103204

92/11/09
12:16:06

/tmp/59201

```
1392 3590
1393 3591
1394 3592
1395 3593
1396 3594
1397 3595
1398 3596
1399 3597
1400 3598
1401 3599
1402 3600
1403 3601
1404 3602
1405 3603
1406 3604
1407 3605
1408 3606
1409 3607
1410 3608
1411 3609
1412 3610
1413 3611
1414 3612
1415 3613
1416 3614
1417 3615
1418 3616
1419 3617
1420 3618
1421 3619
1422 3620
1423 3621
1424 3622
1425 3623
1426 3624
1427 3625
1428 3626
1429 3627
1430 3628
1431 3629
1432 3630
1433 3631
1434 3632
1435 3633
1436 3634
1437 3635
1438 3636
1439 3637
1440 3638
1441 3639
1442 3640
1443 3641
1444 3642
1445 3643
1446 3644
1447 3645
1448 3646
1449 3647
1450 3648
1451 3649

FUNCTION PAC_chk_pad_upload_rec_msg
<PAC_store_chk>

ANY_SDO_FAILURE /* Uploading PAD reported failure */
/* clear the sys msg # from PA tbl associated with PAD */
FUNCTION PAC_clear_upload_fail_pad_sys_msg_num
/* deallocate the DAS message memory */
FUNCTION PAC_das_mem_dealloc_pa_or_tmp_sys_msg_num
/* free up the PA/temporary system msg # */
FUNCTION PAC_free_pa_or_tmp_sys_msg_num
/* idle the TAG port */
FUNCTION PAC_snd_tag_ldh_idle
/* if there is a PAD currently uploading a message, */
/* tell it that PAC is going away. */
/* NOTE: there may be multiple PADs doing uploads */
FUNCTION PAC_snd_pad_ldh_idle_if_upload
/* deallocate DAS memory for PA messages */
FUNCTION PAC_dealloc_msgs_on_signout
/* tell VMS that PA menu is exited */
FUNCTION PAC_snd_vms_pac_deactivated
/* send PRH_PA_FAILURE to the BPRH menu handler */
FUNCTION PAC_snd_bprh_mh_prh_pa_failure
<PAC_wait_failure_ack>

ANY_MSG_FAILURE /* TAG reported failure */
/* deallocate the DAS message memory */
FUNCTION PAC_das_mem_dealloc_pa_or_tmp_sys_msg_num
/* free up the PA/temporary system msg # */
FUNCTION PAC_free_pa_or_tmp_sys_msg_num
/* if there is a PAD currently uploading a message, */
/* tell it that PAC is going away. */
/* NOTE: there may be multiple PADs doing uploads */
FUNCTION PAC_snd_pad_ldh_idle_if_upload
/* deallocate DAS memory for PA messages */
FUNCTION PAC_dealloc_msgs_on_signout
/* tell VMS that PA menu is exited */
FUNCTION PAC_snd_vms_pac_deactivated
/* send PRH_PA_FAILURE to the BPRH menu handler */
FUNCTION PAC_snd_bprh_mh_prh_pa_failure
<PAC_wait_failure_ack>

ANY_FORCE_OOS
/* deallocate the DAS message memory */
FUNCTION PAC_das_mem_dealloc_pa_or_tmp_sys_msg_num
/* free up the PA/temporary system msg # */
FUNCTION PAC_free_pa_or_tmp_sys_msg_num
/* if there is a PAD currently uploading a message, */
/* tell it that PAC is going away */
FUNCTION PAC_snd_pad_ldh_idle_if_upload
/* idle the TAG port */
FUNCTION PAC_snd_tag_ldh_idle
/* deallocate DAS memory for PA messages */
FUNCTION PAC_dealloc_msgs_on_signout
/* check if RC had changed Staff table after signed in */
FUNCTION PAC_chk_staff_chg_after_signin
<PAC_ooo>

TRANS PAC_play_record.tde
```

```
3650
3651
3652
3653
3654
3655
3656
3657
3658
3659
3660
3661
3662
3663
3664
3665
3666
3667
3668
3669
1422 3670 STATE PAC_wait_failure_ack
3671
1423 3672
3673
1424 3674
1425 3675
3676
1426 3677
1427 3678
3679
1428 3680
1429 3681
3682
1430 3683 DONE PAC_wait_failure_ack.tde
3684
3685
3686
3687
3688 08/17/92 chung
3689
3690 08/26/92 chung
3691

EVENT PAC_PA_FAILURE_ACK /* we can proceed */
/* get the PA TAG group member id */
FUNCTION PAC_get_pa_tag_grp_id
TRANS <PAC_get_tag_id>

EVENT ANY_SDO_FAILURE /* Uploading PAD reported failure */
TRANS <PAC_wait_failure_ack>

EVENT ANY_FORCE_OOS
TRANS <PAC_ooo>

/* SCCS UNPROCESSED COMMENTS BEGIN */
/* SCCS UNPROCESSED COMMENTS END */
/* SCCS PROCESSED COMMENTS BEGIN */
/* SCCS PROCESSED COMMENTS END */
In PAC_init_chk_status state.
send PRH_PA_STATUS to menu handler instead of pro
send download status to PRH when PAC gets ack fro
SCCS PROCESSED COMMENTS END */
```

2103204

92/11/09
12:53:01

/tmp/61443

29 105 STATE BPRH_MSG_active
57 180 STATE BPRH_MSG_chk_confdrop
74 222 STATE BPRH_MSG_chk_confdrop
43 141 STATE BPRH_MSG_chk_msg
1 35 STATE BPRH_MSG_idle
14 69 STATE BPRH_MSG_msg_on
90 259 STATE BPRH_MSG_wt_msgack
102 292 STATE BPRH_MSG_wt_msgackon
136 373 STATE BPRH_MSG_wt_rel_ind
115 327 STATE BPRH_MSG_wt_setur_conf

2103204

92/11/09
12:53:00

/tmp/61441

2103204

```
1 /*****
2
3 FSM Name: BPRH_MSG - CAP_ID_MESSAGE
4
5 Description:
6   This is the PRH Voice Messaging Key Filter Capability ID state machine.
7
8 Initial State:
9   BPRH_MSG_idle
10
11 Post Processing State:
12   BPRH_FT_POST_abvss
13
14 Notes:
15   None.
16
17 *****/
18 /*****
19 /*****
20
21 State Name: BPRH_MSG_idle
22
23 Description:
24   This is the initial state which will handle events to initiate call
25   setup to the message call appearance or turn on the message LED.
26
27 Resources Active:
28   None.
29
30 Critical Dependencies:
31   None.
32
33 *****/
34
35 STATE BPRH_MSG_idle
36
37   EVENT PRH_I_MESSAGE
38     FUNCTION BPRH_snd_ldh_messages_req
39     FUNCTION BPRH_set_call_state_change
40     TRANS <BPRH_MSG_wt_msgack>
41
42   EVENT PRH_MSG_CALL_ON
43     FUNCTION BPRH_snd_nl_info_msg_flash_red
44     FUNCTION BPRH_set_msg_waiting
45     FUNCTION BPRH_set_call_state_change
46     TRANS <BPRH_MSG_msg_on>
47
48   EVENT PRH_I_RECOVERY
49     TRANS <BPRH_MSG_idle>
50
51 DONE BPRH_MSG_idle.tde
52
53 /*****
54
55 State Name: BPRH_MSG_msg_on
56
57 Description:
58   This is a stimulus state which is waiting for a request to setup a call
59   to the message call appearance or turn off the message LED.
60
61 Resources Active:
62   Message Key LED is ON.
63
64 Critical Dependencies:
65   None.
66
67 *****/
68
69 STATE BPRH_MSG_msg_on
70
71   EVENT PRH_I_MESSAGE
72     FUNCTION BPRH_snd_ldh_messages_req
73     TRANS <BPRH_MSG_wt_msgack>
74
75   EVENT PRH_MSG_CALL_OFF
76     FUNCTION BPRH_snd_nl_info_msg_off
77     FUNCTION BPRH_clear_msg_waiting
78     FUNCTION BPRH_set_call_state_idle
79     TRANS <BPRH_MSG_idle>
80
81   EVENT PRH_I_RECOVERY
82     FUNCTION BPRH_snd_nl_info_msg_off
83     FUNCTION BPRH_clear_msg_waiting
84     FUNCTION BPRH_set_call_state_idle
85     TRANS <BPRH_MSG_idle>
86
87 DONE BPRH_MSG_msg_on.tde
88
89 *****/
90
91 State Name: BPRH_MSG_active
92
93 Description:
94   This is a stimulus state which is waiting for a call which is active on the
95   message call appearance to be released.
96
97 Resources Active:
98   None.
99
100 Critical Dependencies:
101   None.
102
103 *****/
104
105 STATE BPRH_MSG_active
106
107   EVENT PRH_MESSAGES_CALL_REL
108     FUNCTION BPRH_set_call_state_change
109     FUNCTION BPRH_chk_conf
110     TRANS <BPRH_MSG_chk_confdrop>
111
112   EVENT PRH_MSG_CALL_OFF
113     FUNCTION BPRH_clear_msg_waiting
114     TRANS <BPRH_MSG_active>
115
116   EVENT PRH_I_RECOVERY
117     FUNCTION BPRH_clear_msg_waiting
118     FUNCTION BPRH_set_call_state_change
119     FUNCTION BPRH_chk_conf
120     TRANS <BPRH_MSG_chk_confdrop>
121
122 DONE BPRH_MSG_active.tde
123
124 /*****
125
126 State Name: BPRH_MSG_chk_msg
127
128 Description:
129   *****/
```

9211109
12:53:08

/tmp/61441

129 This is a transitional state which will determine if there are message
130 waiting (ON) or not (OFF). If ON, the message LED will start flashing
131 red, otherwise, it will be turned off.

132 Resources Active:

133 None.

134 Critical Dependencies:

135 None.

43 141 STATE BPRH_MSG_chk_msg

44 142 EVENT PRH_I_MESSAGE_ON

45 143 FUNCTION BPRH_snd_nl_info_msg_flash_red

46 144 TRANS <BPRH_MSG_msg_on>

47 145 EVENT PRH_I_MESSAGE_OFF

48 146 FUNCTION BPRH_snd_nl_info_msg_off

49 147 FUNCTION BPRH_set_call_state_idle

50 148 TRANS <BPRH_MSG_idle>

51 149 EVENT BPRH_I_RECOVERY

52 150 FUNCTION BPRH_clear_msg_waiting

53 151 FUNCTION BPRH_snd_nl_info_msg_off

54 152 FUNCTION BPRH_set_call_state_idle

55 153 TRANS <BPRH_MSG_idle>

56 154 DONE BPRH_MSG_chk_msg.tde

162 State Name: BPRH_MSG_chk_confdrop

163 Description:

164 This is a transitional state which will determine if a there is another
165 call reference that is already active (PRH_I_ACTIVE) on the console, or
166 not (PRH_I_IDLE). If no other active calls, then the call reference will
167 be disconnected and released. If another call is already active, the
168 message call appearance will be dropped from the conference with the active
169 call reference.

170 Resources Active:

171 None.

172 Critical Dependencies:

173 None.

57 180 STATE BPRH_MSG_chk_confdrop

58 181 EVENT PRH_I_IDLE

59 182 FUNCTION BPRH_snd_nl_disc_req

60 183 TRANS <BPRH_MSG_wt_rel_ind>

61 184 EVENT PRH_I_ACTIVE

62 185 FUNCTION BPRH_snd_nl_drop_ack_req

63 186 FUNCTION BPRH_set_call_state_idle

64 187 TRANS <BPRH_MSG_chk_msgs>

65 188 TRANS <BPRH_MSG_chk_msgs>

66 189 TRANS <BPRH_MSG_chk_msgs>

67 190 TRANS <BPRH_MSG_chk_msgs>

68 191 TRANS <BPRH_MSG_chk_msgs>

69 192 TRANS <BPRH_MSG_chk_msgs>

67 193 EVENT PRH_I_RECOVERY

68 194 FUNCTION BPRH_clear_msg_waiting

69 195 FUNCTION BPRH_snd_nl_info_msg_off

70 196 FUNCTION BPRH_snd_nl_drop_ack_req

71 197 FUNCTION BPRH_set_call_state_idle

72 198 TRANS <BPRH_MSG_idle>

73 199 DONE BPRH_MSG_chk_confdrop.tde

204 State Name: BPRH_MSG_chk_confdrop

205 Description:

206 This is a transitional state which will determine if a there is another
207 call reference that is already active (PRH_I_ACTIVE) on the console, or
208 not (PRH_I_IDLE). If no other active calls, the protocol to setup the
209 first active call on the console will be performed. If another call is
210 already active, the message call appearance will be conferenced with the
211 active call reference.

212 Resources Active:

213 None.

214 Critical Dependencies:

215 None.

222 STATE BPRH_MSG_chk_confdrop

223 EVENT PRH_I_IDLE

224 FUNCTION BPRH_alloc_call_ref

225 FUNCTION BPRH_snd_fastconn_nl_setup_req

226 TRANS <BPRH_MSG_wt_setup_conf>

227 EVENT PRH_I_ACTIVE

228 FUNCTION BPRH_snd_nl_conf_ack_own_f1

229 FUNCTION BPRH_snd_msg_setup_ack

230 FUNCTION BPRH_set_call_state_active

231 TRANS <BPRH_MSG_active>

232 EVENT PRH_I_RECOVERY

233 FUNCTION BPRH_clear_msg_waiting

234 FUNCTION BPRH_snd_nl_info_msg_off

235 FUNCTION BPRH_set_call_state_idle

236 TRANS <BPRH_MSG_idle>

237 DONE BPRH_MSG_chk_confdrop.tde

245 State Name: BPRH_MSG_wt_msgack

246 Description:

247 This is a stimulus state which is waiting for an acknowledgement to the
248 LDR_MESSAGES_REQ sent when the message key was pressed.

249 Resources Active:

250 None.

251 Critical Dependencies:

252 None.

2103204

92/11/09
12:53:00

/tmp/61441

3

2103204

```
*****
257 STATE BPRH_MSG_wt_msgack
258
259 STATE BPRH_MSG_wt_msgack
260
261 EVENT PRH_MESSAGES_ACK BPRH_save_bmu_cap_id
262 FUNCTION BPRH_chk_other_active_call
263 TRANS <BPRH_MSG_chk_confrig>
264
265
266 EVENT PRH_MESSAGE_DENIED BPRH_set_call_state_idle
267 FUNCTION BPRH_MSG_idle>
268 TRANS
269
270 EVENT PRH_I_RECOVERY BPRH_set_call_state_idle
271 FUNCTION
272 TRANS <BPRH_MSG_idle>
273
274 DONE BPRH_MSG_wt_msgack.tde
275
276 /
277
278 State Name: BPRH_MSG_wt_msgackon
279
280 Description:
281 This is a stimulus state which is waiting for an acknowledgement to the
282 LDH_MESSAGES_req sent when the message key was pressed.
283
284 Resources Active:
285 Message Key LED is ON.
286
287 Critical Dependencies:
288 None.
289
290 *****
291 STATE BPRH_MSG_wt_msgackon
292
293 EVENT PRH_MESSAGES_ACK BPRH_save_bmu_cap_id
294 FUNCTION BPRH_chk_other_active_call
295 TRANS <BPRH_MSG_chk_confrig>
296
297 EVENT PRH_MESSAGE_DENIED BPRH_MSG_msg_on>
298 TRANS
299
300 EVENT PRH_I_RECOVERY BPRH_snd_nl_info_msg_off
301 FUNCTION BPRH_clear_msg_waiting
302 TRANS <BPRH_MSG_idle>
303
304
305 STATE BPRH_MSG_wt_msgackon.tde
306
307 /
308
309 State Name: BPRH_MSG_wt_setup_conf
310
311 Description:
312 This is a stimulus state which is waiting for confirmation (CONNECT) to a
313 SETUP request to terminate a Nailed Up call. The console will be sent
314 a RELEASE on an early disconnect.
315
316 Resources Active:
317 None.
318
319 *****
320
321 Critical Dependencies:
322 None.
323
324
325 *****
326 STATE BPRH_MSG_wt_setup_conf
327
328 EVENT NL_SETUP_CONF BPRH_snd_nl_info_msg_off
329 FUNCTION BPRH_snd_nl_info_msg_waiting
330 TRANS <BPRH_MSG_active>
331
332 EVENT NL_SETUP_ERR_CONF BPRH_set_call_state_idle
333 FUNCTION
334 TRANS <BPRH_MSG_idle>
335
336 EVENT NL_REJECT_IND BPRH_set_call_state_idle
337 FUNCTION
338 TRANS <BPRH_MSG_idle>
339
340 EVENT NL_DISC_IND BPRH_snd_nl_rel_req
341 FUNCTION BPRH_set_call_state_idle
342 TRANS <BPRH_MSG_idle>
343
344
345 EVENT PRH_I_RECOVERY BPRH_clear_msg_waiting
346 FUNCTION BPRH_snd_nl_info_msg_off
347 TRANS <BPRH_MSG_idle>
348
349
350 STATE BPRH_MSG_wt_setup_conf.tde
351
352 /
353
354 State Name: BPRH_MSG_wt_rel_ind
355
356 Description:
357 This is a stimulus state which is waiting for a release indication to a
358 DISCONNECT primitive that was sent to the console.
359
360 Resources Active:
361 None.
362
363 Critical Dependencies:
364 None.
365
366 *****
367 STATE BPRH_MSG_wt_rel_ind
368
369 EVENT NL_REL_IND BPRH_set_call_state_idle
370 FUNCTION BPRH_chk_messages
371 TRANS <BPRH_MSG_chk_msg>
372
373
374 EVENT NL_REL_CONF BPRH_set_call_state_idle
375 FUNCTION BPRH_chk_messages
376 TRANS <BPRH_MSG_chk_msg>
377
378
379
380
381
382
383
384
```

92/11/09
12:53:00

/tmp/61441

2103204

```

-47 385      EVENT  NL_REL_ERR_CONF
-48 386      FUNCTION  BPRH_set_call_state_idle
-49 387      FUNCTION  BPRH_chk_messages
-50 388      TRANS   <BPRH_MSG_chk_msg>
-51 389
-52 390      EVENT  PPRH : RECOVERY
-53 391      FUNCTION  BPRH_snd_nl_info_msg_off
-54 392      FUNCTION  BPRH_clear_msg_waiting
-55 393      FUNCTION  BPRH_set_call_state_idle
-56 394      TRANS   <BPRH_MSG_idle>
-57 395
-58 396 DONE  BPRH_MSG_wt_rel_include
-59 397
-60 398      /** SCCS UNPROCESSED COMMENTS BEGIN **
-61 399      SCCS UNPROCESSED COMMENTS END **/
-62 400
-63 401      /** SCCS PROCESSED COMMENTS BEGIN **
-64 402 03/10/92 corrigian File created.
-65 403 04/14/92 corrigian Added changes for PC Recovery.
-66 404 05/01/92 corrigian Made changes per review.
-67 405 05/03/92 corrigian Made changes for PRAM 13707 (connection port = logical port)
-68 406      by removing functions to allocate and free B channel.
-69 407      SCCS PROCESSED COMMENTS END **/

```

92/11/09
12:52:49

/tmp/61373

54 186 STATE BPPH_MH_AVS_acd_ack
27 135 STATE BPPH_MH_AVS_acd_pa_stim
14 89 STATE BPPH_MH_AVS_acd_stim
5 61 STATE BPPH_MH_AVS_chk_seat_status
1 35 STATE BPPH_MH_AVS_idle
77 220 STATE BPPH_MH_AVS_wt_sub

2103204

92/11/09
13:52:48

/tmp/61371

Part III

2103204

```
1 / *****
2
3 FSM Name: BPRH_MH_AVS - CAP_ID_MH_AVS
4
5 Description:
6   This is the PRH Agent Voice Services Capability ID state machine.
7
8 Initial State:
9   BPRH_MH_AVS_idle
10
11 Post Processing State:
12   BPRH_MH_POST_abyss
13
14 Notes:
15   None.
16
17 *****/
18 / *****
19 / *****
20
21 State Name: BPRH_MH_AVS_idle
22
23 Description:
24   This is the initial state which will handle events to activate this menu
25   and display the main menu.
26
27 Resources Active:
28   None.
29
30 Critical Dependencies:
31   None.
32
33 *****/
34
35 STATE BPRH_MH_AVS_idle
36
37 EVENT BPRH_MH_AVS_activate
38   FUNCTION BPRH_chk_pa_feat_status
39   TRANS <BPRH_MH_AVS_chk_feat_status>
40
41 DONE BPRH_MH_AVS_idle.tde
42
43 / *****
44
45 State Name: BPRH_MH_AVS_chk_feat_status
46
47 Description:
48   This is a transitional state which will determine if the Personal
49   Announcement feature has been provisioned, and display the appropriate
50   menu and transition to the appropriate stimulus state based on the internal
51   event.
52
53 Resources Active:
54   None.
55
56 Critical Dependencies:
57   None.
58
59 *****/
60
61 STATE BPRH_MH_AVS_chk_feat_status
62
63 EVENT BPRH_I_PA_NOT_PROVISIONED
64   FUNCTION BPRH_snd_nl_inf_avs_menu
65
66
67
68
69
70
71
72
73 / *****
74
75 State Name: BPRH_MH_AVS_acd_stim
76
77 Description:
78   This is a stimulus state which is waiting for either the ACD Mail,
79   Soft key 1 or an Exit, Soft key 4, to be selected.
80
81 Resources Active:
82   None.
83
84 Critical Dependencies:
85   None.
86
87 *****/
88
89 STATE BPRH_MH_AVS_acd_stim
90
91 EVENT PRH_SOFT_KEY_1
92   FUNCTION BPRH_snd_vms_call_req
93   FUNCTION BPRH_set_ack_timer
94   TRANS <BPRH_MH_AVS_acd_ack>
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119 / *****
120
121 State Name: BPRH_MH_AVS_acd_pa_stim
122
123 Description:
124   This is a stimulus state which is waiting for either the ACD Mail,
125   Soft key 1, Personal, Soft key 2 or an Exit, Soft key 4, to be selected.
126
127 Resources Active:
128   None.
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
```

92/11/09
13:32:48

/tmp/61371

2103204

```
129 Critical Dependencies:
130 None.
131
132
133
134
135 STATE BPRH_MH_AVS_acd_pa_stim
136
137 EVENT PRH_I_SOFT_KEY_1
138 FUNCTION BPRH_snd_vms_call_req
139 FUNCTION BPRH_snd_ack_timer
140 TRANS <BPRH_MH_AVS_acd_ack>
141
142
143 EVENT PRH_I_SOFT_KEY_2
144 FUNCTION BPRH_snd_vms_pa_req
145 FUNCTION BPRH_snd_mhi_ap_activate
146 TRANS <BPRH_MH_AVS_wc_sub>
147
148
149 EVENT PRH_I_SOFT_KEY_4
150 FUNCTION BPRH_snd_vms_done_req
151 FUNCTION BPRH_clear_menu_handler_block
152 FUNCTION BPRH_snd_ni_inf_clear_display
153 TRANS <BPRH_MH_AVS_idle>
154
155
156 EVENT PRH_MENU_REFRESH
157 FUNCTION BPRH_check_pa_feat_status
158 TRANS <BPRH_MH_AVS_chk_feat_status>
159
160
161 EVENT PRH_MH_EXIT
162 FUNCTION BPRH_clear_menu_handler_block
163 FUNCTION BPRH_snd_ni_inf_clear_display
164 TRANS <BPRH_MH_AVS_idle>
165
166
167 EVENT PRH_I_PLUGIN
168 FUNCTION BPRH_set_plugin
169 FUNCTION BPRH_snd_ldh_plugin
170 FUNCTION BPRH_check_pa_feat_status
171 TRANS <BPRH_MH_AVS_chk_feat_status>
172
173
174 STATE Name: BPRH_MH_AVS_acd_ack
175
176 Description:
177 This is a stimulus state which is waiting for an acknowledgement from ACD
178 Mail to a VMS_CALL_REQ.
179
180 Resources Active:
181 None.
182
183 Critical Dependencies:
184 None.
185
186
187 STATE BPRH_MH_AVS_acd_ack
188
189 EVENT PRH_CALLBACK_WAITING
190 FUNCTION BPRH_cancel_ack_timer
191 FUNCTION BPRH_save_callback_waiting_data
192 TRANS <BPRH_MH_AVS_wc_sub>
193
194
195 EVENT PPH_VMS_FAILURE
196 FUNCTION BPRH_cancel_ack_timer
197 TRANS <BPRH_MH_AVS_acd_stim>
198
199
200 EVENT PRH_I_TIMEOUT
201 FUNCTION BPRH_ack_timeout
202 TRANS <BPRH_MH_AVS_idle>
203
204
205 DONE BPRH_MH_AVS_acd_ack.tde
206
207
208 State Name: BPRH_MH_AVS_wc_sub
209 Description:
210 This is a stimulus state which is waiting for the sub menu that was
211 activated to exit. It will then redisplay the main Message Selection Menu.
212
213 Resources Active:
214 None.
215
216 Critical Dependencies:
217 None.
218
219
220 STATE BPRH_MH_AVS_wc_sub
221
222 EVENT PRH_I_SUB_MENU_EXIT
223 FUNCTION BPRH_check_pa_feat_status
224 TRANS <BPRH_MH_AVS_chk_feat_status>
225
226
227 DONE BPRH_MH_AVS_wc_sub.tde
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
```

92/11/09

12:52:31

/tmp/61273

```
121 293 STATE BPRH_MHC_PP_all_stimulus
112 262 STATE BPRH_MHC_PP_essced_timeout
213 464 STATE BPRH_MHC_PP_failure_timeout
1 36 STATE BPRH_MHC_PP_idle
7 62 STATE BPRH_MHC_PP_start_stimulus
55 157 STATE BPRH_MHC_PP_stop_stimulus
156 356 STATE BPRH_MHC_PP_wt_record_ack
35 114 STATE BPRH_MHC_PP_wt_start_record_ack
88 215 STATE BPRH_MHC_PP_wt_stop_record_ack
189 416 STATE BPRH_MHC_PP_wt_store_complete
```

2103204

92/11/09
12:52:38

/tmp/61271

2103204

```
1 /*****
2
3 FSM Name: BPRH_MHC_PP - CAP_ID_SMH_RECORD_PERS_ANM
4
5 Description:
6   This is the PRH Record Personal Announcement Sub Menu Capability ID
7   state machine.
8
9 Initial State:
10  BPRH_MHC_PP_idle
11
12 Post Processing State:
13  BPRH_MH_POST_ack:33
14
15 Notes:
16  None.
17
18 *****/
19
20 /*****/
21
22 State Name: BPRH_MHC_PP_idle
23
24 Description:
25   This is the initial state which will handle events to the activation of this
26   state machine.
27
28 Resources Active:
29  None.
30
31 Critical Dependencies:
32  None.
33
34 *****/
35
36 STATE BPRH_MHC_PP_idle
37
38   EVENT PRH_I_MHC_PP_ACTIVATE
39     FUNCTION BPRH_save_msg_status
40     FUNCTION BPRH_snd_ni_info_record_pa_start_menu
41   TRANS <BPRH_MHC_PP_start_stimulus>
42
43 DONE BPRH_MHC_PP_idle.tde
44
45 /*****/
46
47 State Name: BPRH_MHC_PP_start_stimulus
48
49 Description:
50   This is a stimulus state that is waiting for an event indicating a Soft key
51   press indicating the start of a recording (Soft Key 1) or an Exit from the
52   sub menu (Soft Key 4).
53
54 Resources Active:
55  None.
56
57 Critical Dependencies:
58  None.
59
60 *****/
61
62 STATE BPRH_MHC_PP_start_stimulus
63
64   EVENT PRH_I_SOFT_KEY_1
65     FUNCTION BPRH_snd_pac_start_record
66     FUNCTION BPRH_set_ack_timer
67   TRANS <BPRH_MHC_PP_start_record_ack>
68
69   EVENT PRH_I_SOFT_KEY_4
70     FUNCTION BPRH_restore_msg_status
71     FUNCTION BPRH_snd_pac_record_exit
72     FUNCTION BPRH_snd_ni_info_clear_display
73     FUNCTION BPRH_snd_ni_info_record_pa_start_menu
74   TRANS <BPRH_MHC_PP_idle>
75
76   EVENT PRH_PA_INFO
77     FUNCTION BPRH_save_pa_info
78     FUNCTION BPRH_snd_ni_info_record_pa_start_menu
79   TRANS <BPRH_MHC_PP_start_stimulus>
80
81   EVENT PRH_PA_FAILURE
82     FUNCTION BPRH_snd_ni_info_pa_failure
83     FUNCTION BPRH_set_display_timer
84   TRANS <BPRH_MHC_PP_failure_timeout>
85
86   EVENT PRH_I_PLUGIN
87     FUNCTION BPRH_set_plugin
88     FUNCTION BPRH_snd_ldh_plugin
89     FUNCTION BPRH_snd_ni_info_record_pa_start_menu
90   TRANS <BPRH_MHC_PP_start_stimulus>
91
92   EVENT PRH_MENU_REFRESH
93     FUNCTION BPRH_snd_ni_info_record_pa_start_menu
94   TRANS <BPRH_MHC_PP_start_stimulus>
95
96 DONE BPRH_MHC_PP_start_stimulus.tde
97
98 /*****/
99
100 State Name: BPRH_MHC_PP_start_record_ack
101
102 Description:
103   This is a stimulus state which is waiting for an acknowledgement event
104   indicating that recording has started.
105
106 Resources Active:
107  None.
108
109 Critical Dependencies:
110  None.
111
112 *****/
113
114 STATE BPRH_MHC_PP_start_record_ack
115
116   EVENT PRH_START_RECORD_ACK
117     FUNCTION BPRH_cancel_timer
118     FUNCTION BPRH_snd_ni_info_record_pa_stop_menu
119   TRANS <BPRH_MHC_PP_start_stimulus>
120
121   EVENT PRH_PA_INFO
122     FUNCTION BPRH_save_pa_info
123     FUNCTION BPRH_snd_ni_info_record_ack
124   TRANS <BPRH_MHC_PP_start_record_ack>
125
126   EVENT PRH_PA_FAILURE
127     FUNCTION BPRH_cancel_timer
128     FUNCTION BPRH_snd_ni_info_pa_failure
129     FUNCTION BPRH_set_display_timer
130   TRANS <BPRH_MHC_PP_start_record_ack>
```

92/11/89
12:52:38

/tmp/61271

```
47 129 TRANS <BPRH_MHC_RP_failure_timeout>
48 130
49 131 EVENT PRH_I_TIMEOUT
50 132 FUNCTION BPRH_restore_msg_status
51 133 FUNCTION BPRH_snd_nl_info_clear_display
52 134 FUNCTION BPRH_snd_l1_sub_menu_exit
53 135 FUNCTION BPRH_snd_l0_sub_menu_exit
54 136 TRANS <BPRH_MHC_RP_idle>
55 137
56 138 DONE BPRH_MHC_RP_wt_start_record_ack.tde
57 139
58 140 /
59 141
60 142 State Name: BPRH_MHC_RP_stop_stimulus
61 143
62 144 Description:
63 145 This is a stimulus state that is waiting for an event indicating a Soft key
64 146 press indicating the Stoppage of a Recording (Soft Key 1) or an Exit from
65 147 the sub menu (Soft Key 4).
66 148
67 149 Resources Active:
68 150 None.
69 151
70 152 Critical Dependencies:
71 153 None.
72 154
73 155
74 156
75 157 STATE BPRH_MHC_RP_stop_stimulus
76 158
77 159 EVENT PRH_I_SOFT_KEY_1
78 160 FUNCTION BPRH_snd_pac_stop_record
79 161 FUNCTION BPRH_set_ack_timer
80 162 TRANS <BPRH_MHC_RP_wt_stop_record_ack>
81 163
82 164 EVENT PRH_I_SOFT_KEY_4
83 165 FUNCTION BPRH_restore_msg_status
84 166 FUNCTION BPRH_snd_pac_record_exit
85 167 FUNCTION BPRH_snd_nl_info_clear_display
86 168 FUNCTION BPRH_snd_l1_sub_menu_exit
87 169 TRANS <BPRH_MHC_RP_idle>
88 170
89 171 EVENT ANY_EXCEED_MSG_SIZE
90 172 FUNCTION BPRH_set_msg_recorded
91 173 FUNCTION BPRH_snd_nl_info_pa_failure
92 174 TRANS <BPRH_MHC_RP_failure_timeout>
93 175
94 176 EVENT PRH_PA_FAILURE
95 177 FUNCTION BPRH_cancel_timer
96 178 FUNCTION BPRH_snd_nl_info_pa_failure
97 179 TRANS <BPRH_MHC_RP_failure_timeout>
98 180
99 181 EVENT PRH_I_TIMEOUT
100 182 FUNCTION BPRH_restore_msg_status
101 183 FUNCTION BPRH_snd_nl_info_clear_display
102 184 TRANS <BPRH_MHC_RP_sub_menu_exit>
103 185
104 186 EVENT PRH_PA_FAILURE
105 187 FUNCTION BPRH_cancel_timer
106 188 FUNCTION BPRH_snd_nl_info_pa_failure
107 189 TRANS <BPRH_MHC_RP_failure_timeout>
108 190
109 191 EVENT PRH_I_TIMEOUT
110 192 FUNCTION BPRH_restore_msg_status
111 193 FUNCTION BPRH_snd_nl_info_clear_display
112 194 TRANS <BPRH_MHC_RP_sub_menu_exit>
113 195
114 196 EVENT PRH_PA_FAILURE
115 197 FUNCTION BPRH_cancel_timer
116 198 FUNCTION BPRH_snd_nl_info_pa_failure
117 199 TRANS <BPRH_MHC_RP_failure_timeout>
118 200
119 201 STATE BPRH_MHC_RP_wt_stop_record_ack
120 202
121 203 EVENT PRH_STOP_RECORD_ACK
122 204 FUNCTION BPRH_set_msg_recorded
123 205 TRANS <BPRH_MHC_RP_all_stimulus>
124 206
125 207 STATE BPRH_MHC_RP_wt_stop_record_ack
126 208
127 209 EVENT PRH_STOP_RECORD_ACK
128 210 FUNCTION BPRH_cancel_timer
129 211 FUNCTION BPRH_set_msg_recorded
130 212 TRANS <BPRH_MHC_RP_all_stimulus>
131 213
132 214
133 215 STATE BPRH_MHC_RP_wt_stop_record_ack
134 216
135 217 EVENT PRH_STOP_RECORD_ACK
136 218 FUNCTION BPRH_cancel_timer
137 219 FUNCTION BPRH_set_msg_recorded
138 220 TRANS <BPRH_MHC_RP_all_stimulus>
139 221
140 222
141 223 EVENT ANY_EXCEED_MSG_SIZE
142 224 FUNCTION BPRH_cancel_timer
143 225 FUNCTION BPRH_set_msg_recorded
144 226 TRANS <BPRH_MHC_RP_all_stimulus>
145 227
146 228 EVENT PRH_PA_FAILURE
147 229 FUNCTION BPRH_cancel_timer
148 230 FUNCTION BPRH_snd_nl_info_pa_failure
149 231 TRANS <BPRH_MHC_RP_failure_timeout>
150 232
151 233 EVENT PRH_I_TIMEOUT
152 234 FUNCTION BPRH_restore_msg_status
153 235 FUNCTION BPRH_snd_nl_info_clear_display
154 236 TRANS <BPRH_MHC_RP_sub_menu_exit>
155 237
156 238 EVENT PRH_PA_FAILURE
157 239 FUNCTION BPRH_cancel_timer
158 240 FUNCTION BPRH_snd_nl_info_pa_failure
159 241 TRANS <BPRH_MHC_RP_failure_timeout>
160 242
161 243 STATE BPRH_MHC_RP_wt_stop_record_ack.tde
162 244
163 245 /
164 246
165 247 State Name: BPRH_MHC_RP_exceed_timeout
166 248
167 249 Description:
168 250 This is a stimulus state which is waiting for a timeout indication.
169 251 When the timeout event is received, the recording sub menu with the
170 252 Start, Play, Store and Exit Soft keys will be displayed.
171 253
172 254 Resources Active:
173 255 Display timer.
```

2103204

92/11/09
12:52:38

/tmp/61271

2103204

```
257 Critical Dependencies:
258 None.
259
260 *****
261 STATE BPPH_MHC_PP_exceed_timeout
262
263
264 EVENT PRH_I_TIMEOUT
265 FUNCTION BPPH_snd_nl_info_record_pa_menu
266 TRANS <BPPH_MHC_PP_all_stimulus>
267
268
269 EVENT PRH_PA_FAILURE
270 FUNCTION BPPH_snd_nl_info_pa_failure
271 TRANS <BPPH_MHC_PP_failure_timeout>
272
273 DONE BPPH_MHC_PP_exceed_timeout.tde
274
275 *****
276
277 State Name: BPPH_MHC_PP_all_stimulus
278
279 Description:
280 This is a stimulus state that is waiting for a Soft key press indicating
281 the Start of a recording (Soft Key 1), the Playing of an announcement
282 (Soft Key 2), the acceptance, or Store, of an announcement (Soft Key 3)
283 or an Exit from the sub menu (Soft Key 4).
284
285 Resources Active:
286 None.
287
288 Critical Dependencies:
289 None.
290
291 *****
292
293 STATE BPPH_MHC_PP_all_stimulus
294
295 EVENT PRH_I_SOFT_KEY_1
296 FUNCTION BPPH_snd_pac_record_req
297 TRANS <BPPH_MHC_PP_wt_record_ack>
298
299
300 EVENT PRH_I_SOFT_KEY_2
301 FUNCTION BPPH_snd_pac_play_req
302 TRANS <BPPH_MHC_PP_all_stimulus>
303
304
305 EVENT PRH_I_SOFT_KEY_3
306 FUNCTION BPPH_snd_pac_store_req
307 TRANS <BPPH_MHC_PP_wt_store_complete>
308
309
310 EVENT PRH_I_SOFT_KEY_4
311 FUNCTION BPPH_restore_msg_status
312 FUNCTION BPPH_snd_pac_record_exit
313 FUNCTION BPPH_snd_nl_info_clear_display
314 TRANS <BPPH_MHC_PP_idle>
315
316
317 EVENT PRH_PA_INFO
318 FUNCTION BPPH_save_pa_info
319 TRANS <BPPH_MHC_PP_all_stimulus>
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336 DONE BPPH_MHC_PP_all_stimulus.tde
337
338 *****
339
340 State Name: BPPH_MHC_PP_wt_record_ack
341
342 Description:
343 This is a stimulus state which is waiting for an acknowledgement event
344 from the Personal Announcement Controller (PAC) on the BMC. If a Recorder
345 Full event is received, a Waiting for Recorder Resources menu will be
346 displayed, and continue to wait for an acknowledgement event to proceed.
347
348 Resources Active:
349 None.
350
351 Critical Dependencies:
352 None.
353
354 *****
355
356 STATE BPPH_MHC_PP_wt_record_ack
357
358 EVENT PRH_RECORD_ACK
359 FUNCTION BPPH_cancel_timer
360 TRANS <BPPH_MHC_PP_wt_start_record>
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
```

/tmp/61271

```

449 State Name: BPRH_MHC_PP_failure_timeout
450
451 Description:
452 This is a stimulus state which is waiting for a timeout indication.
453 When the timeout event is received, this menu will be edited, and the
454 main level menu reactivated.
455
456 Resources Active:
457 Display timer.
458
459 Critical Dependencies:
460 None.
461
462 *****
463
464 STATE BPRH_MHC_PP_failure_timeout
465
466 EVENT PRH_I_TIMEOUT
467 FUNCTION BPRH_snd_pac_pa_failure_ack
468 FUNCTION BPRH_restore_msg_status
469 FUNCTION BPRH_snd_nl_info_clear_display
470 FUNCTION BPRH_snd_ll_sub_menu_exit
471 FUNCTION BPRH_snd_lc_sub_menu_exit
472 TRANS <BPRH_MHC_PP_idle>
473
474 DONE BPRH_MHC_PP_failure_timeout.tde
475
476
477 /** SCCS UNPROCESSED COMMENTS BEGIN **
478 SCCS UNPROCESSED COMMENTS END **
479
480 /** SCCS PROCESSED COMMENTS BEGIN **
481 Corrigan File created.
482 Corrigan Added enhancements based on re-inspection
483 Corrigan Made changes per review.
484 Corrigan Call BPRH_set_msg_recorded on PRH_STORE_COMPLETE
485 kamdar added BPRH_cancel_timer in wait start record ack &
486 wait stop record ack states.
487 kamdar added BPRH_snd_pac_pa_failure_ack in
488 BPRH_MHC_PP_failure_timeout state.
489 Added BPRH_MHC_PP_wt_record_ack state to handle record
490 SCCS PROCESSED COMMENTS END **

```

```

/** SCSS UNPROCESSED COMMENTS BEGIN **
SCSS UNPROCESSED COMMENTS END **

/** SCSS PROCESSED COMMENTS BEGIN **
reated.
enhancements based on re-inspection
changes per review.
BPRH_set_flg recorded on PRH STORE COMP
PRH_cancel_timer in wait start record a
e record ack states.
BPRH_snd_pac_pa_failure_ack in
BPR_failure_timeout state.
BPRH_MH2_Rp wt_record_ack state to han
SCSS PROCESSED COMMENTS END **/

```

2103204

921109
12:52:14

/tmp/61183

108	286	STATE	BPRH_MHI_AP_all_stimulus
148	356	STATE	BPRH_MHI_AP_erase_confirm
31	132	STATE	BPRH_MHI_AP_failure_timeout
1	38	STATE	BPRH_MHI_AP_idle
23	103	STATE	BPRH_MHI_AP_msg_status
38	159	STATE	BPRH_MHI_AP_rec_stimulus
6	64	STATE	BPRH_MHI_AP_wc_info
71	222	STATE	BPRH_MHI_AP_wc_record_ack
170	402	STATE	BPRH_MHI_AP_wc_sub

2103204

92/11/09
12:52:13

/tmp/61181

```
1 /*****
2
3 FSM Name: BPRH_MH1_AP - CAP_ID_SMH_AGENT_PERS_ANN
4
5 Description:
6   This is the BPRH Agent Personal Announcements Sub Menu Capability ID
7   state machine.
8
9 Initial State:
10  BPRH_MH1_AP_idle
11
12 Post Processing State:
13  BPRH_MH_POST_abyss
14
15 Notes:
16  None.
17
18 *****/
19 /*****
20
21 State Name: BPRH_MH1_AP_idle
22
23 Description:
24  This is the initial state which will handle events to activate the Agent
25  Personal Announcement submenu. It will set a timer to wait for an event
26  containing Personal Announcement information, and transition to a state to
27  wait for that event.
28
29 Resources Active:
30  None.
31
32 Critical Dependencies:
33  None.
34
35 *****/
36 /*****
37
38 STATE BPRH_MH1_AP_idle
39
40 EVENT PRH_I_MH1_AP_ACTIVATE
41 FUNCTION FTNCTION BPRH_set_ack_timer
42 TRANS <BPRH_MH1_AP_wt_info>
43
44 DONE BPRH_MH1_AP_idle.tde
45
46 /*****
47
48 State Name: BPRH_MH1_AP_wt_info
49
50 Description:
51  This is a stimulus state that is waiting for an event containing Personal
52  Announcement information. A function will be called to generate an internal
53  event indicating the status of the Personal Announcement. If a timeout
54  event is received, the submenu will be exited.
55
56 Resources Active:
57  Inter processor timer active.
58
59 Critical Dependencies:
60  None.
61
62 *****/
63
64 STATE BPRH_MH1_AP_wt_inf:
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86 /*****
87
88 State Name: BPRH_MH1_AP_msg_status
89
90 Description:
91  This is a transitional state which will determine the status of a personal
92  announcement, display the a menu with the appropriate soft keys, and
93  transition to a state that will only accept the soft keys displayed.
94
95 Resources Active:
96  None.
97
98 Critical Dependencies:
99  None.
100
101 *****/
102
103 STATE BPRH_MH1_AP_msg_status
104
105 EVENT PRH_I_RECORD_ONLY
106 FUNCTION <BPRH_MH1_AP_rec_stimulus>
107 TRANS <BPRH_MH1_AP_rec_stimulus>
108
109
110
111
112
113 DONE BPRH_MH1_AP_msg_status.tde
114
115 /*****
116
117 State Name: BPRH_MH1_AP_failure_timeout
118
119 Description:
120  This is a stimulus state which is waiting for a timeout indication.
121  When the timeout event is received, this menu will be exited, and the
122  main level menu reactivated.
123
124 Resources Active:
125  Display timer.
126
127 Critical Dependencies:
128  None.
129
130 *****/
131
132 STATE BPRH_MH1_AP_failure_timeout
133
134 EVENT PRH_I_ALL_KEYS
135 FUNCTION <BPRH_MH1_AP_all_stimulus>
136 TRANS <BPRH_MH1_AP_all_stimulus>
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

2103204

92/11/09
13:52:13

/tmp/61181

```
129 *****
130 *****
131 *****
132 STATE BPRH_MH1_AP_failure_timeout
133
134 EVENT PRH_I_TIMEOUT
135 FUNCTION BPRH_snd_pac_pa_failure_ack
136 FUNCTION BPRH_snd_nl_info_clear_display
137 FUNCTION BPRH_snd_lo_sub_menu_exit
138 TRANS <BPRH_MH1_AF_idle>
139
140 DONE BPRH_MH1_AF_failure_timeout.tde
141
142 /*****
143
144 State Name: BPRH_MH1_AF_rec_stimulus
145
146 Description:
147 This is a stimulus state that is waiting for an event indicating a Soft or
148 Scroll key has been pressed.
149
150 Resources Active:
151 The RECORD and EXIT soft keys are displayed.
152 The Scroll keys may be displayed depending on the Arrow Status.
153
154 Critical Dependencies:
155 None.
156
157 *****
158
38 159 STATE BPRH_MH1_AF_rec_stimulus
159
160
161 EVENT PRH_I_SOFT_KEY_2
162 FUNCTION BPRH_snd_pac_record_req
163 FUNCTION BPRH_snd_ack_timer
164 TRANS <BPRH_MH1_AF_wt_record_ack>
165
166 EVENT PRH_I_SOFT_KEY_4
167 FUNCTION BPRH_snd_nl_info_clear_display
168 FUNCTION BPRH_snd_pac_exit
169 FUNCTION BPRH_snd_lo_sub_menu_exit
170 TRANS <BPRH_MH1_AF_idle>
171
172
173 EVENT PRH_PA_INFO
174 FUNCTION BPRH_save_pa_info
175 FUNCTION BPRH_check_pa_msg_status
176 TRANS <BPRH_MH1_AF_msg_status>
177
178 EVENT PRH_I_SCROLL_UP
179 FUNCTION BPRH_snd_pac_prev_req
180 TRANS <BPRH_MH1_AF_rec_stimulus>
181
182 EVENT PRH_I_SCROLL_DOWN
183 FUNCTION BPRH_snd_pac_next_req
184 TRANS <BPRH_MH1_AF_rec_stimulus>
185
186 EVENT PRH_PA_FAILURE
187 FUNCTION BPRH_snd_nl_info_pa_failure
188 TRANS <BPRH_MH1_AF_failure_timeout>
189
190
191 EVENT PRH_I_PLUGIN
192 FUNCTION BPRH_set_plugin
193 FUNCTION BPRH_snd_idb_plugin
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
```

2103204

```
65 193 FUNCTION BPRH_snd_nl_info_agent_pa_record_menu
66 194 TRANS <BPRH_MH1_AP_rec_stimulus>
67 195
68 196 EVENT PRH_MENU_REFRESH
69 197 FUNCTION BPRH_snd_nl_info_agent_pa_record_menu
70 198 TRANS <BPRH_MH1_AP_rec_stimulus>
71 199
72 200 DONE BPRH_MH1_AP_rec_stimulus.tde
73 201
74 202 /*****
75 203
76 204 State Name: BPRH_MH1_AF_wt_record_ack
77 205
78 206 Description:
79 This is a stimulus state which is waiting for an acknowledgement event
80 from the Personal Announcement Controller (PAC) on the BMU. When the
81 record ack event is received, the PA Record submenu will be activated,
82 and this menu will wait for the PA Record submenu to exit. If a Recorder
83 Full event is received, a Waiting for Recorder Resources menu will be
84 displayed, and continue to wait for an acknowledgement event to proceed.
85
86 Resources Active:
87 None.
88
89 Critical Dependencies:
90 None.
91
92 *****
93
94 STATE BPRH_MH1_AP_wt_record_ack
95
96
97 EVENT PRH_RECORD_ACK
98 FUNCTION BPRH_cancel_timer
99 FUNCTION BPRH_snd_mhc_fp_activate
100 TRANS <BPRH_MH1_AP_wt_sub>
101
102 EVENT PRH_PA_DAS_FULL
103 FUNCTION BPRH_cancel_timer
104 FUNCTION BPRH_snd_nl_info_recorder_busy_menu
105 TRANS <BPRH_MH1_AP_wt_record_ack>
106
107 EVENT PRH_I_SOFT_KEY_4
108 FUNCTION BPRH_cancel_timer
109 FUNCTION BPRH_snd_nl_info_clear_display
110 FUNCTION BPRH_snd_pac_abort_record
111 FUNCTION BPRH_snd_lo_sub_menu_exit
112 TRANS <BPRH_MH1_AP_idle>
113
114
115 EVENT PRH_PA_INFO
116 FUNCTION BPRH_save_pa_info
117 TRANS <BPRH_MH1_AP_wt_record_ack>
118
119 EVENT PRH_I_TIMEOUT
120 FUNCTION BPRH_snd_pac_exit
121 FUNCTION BPRH_snd_lo_sub_menu_exit
122 TRANS <BPRH_MH1_AP_idle>
123
124
125 EVENT PRH_PA_FAILURE
126 FUNCTION BPRH_cancel_timer
127 FUNCTION BPRH_snd_nl_info_pa_failure
128 TRANS <BPRH_MH1_AF_failure_timeout>
129
130
```

9211109
12:52:13

/tmp/61181

```
99 257 EVENT PRH_I_PLUGIN BPRH_snd_nl_info_recorder_busy_menu
100 258 FUNCTION BPRH_snd_nl_info_recorder_busy_menu
101 259 FUNCTION BPRH_snd_nl_info_recorder_busy_menu
102 260 FUNCTION BPRH_snd_nl_info_recorder_busy_menu
103 261 TRANS <BPRH_MH1_AP_wt_record_ack>
104 262
105 263 EVENT PRH_MENU_REFRESH
106 264 FUNCTION BPRH_snd_nl_info_recorder_busy_menu
107 265 TRANS <BPRH_MH1_AP_wt_record_ack>
108 266
109 267 DONE BPRH_MH1_AP_wt_record_ack.tde
110 268
111 269 /*****
112 270 State Name: BPRH_MH1_AP_all_stimulus
113 271
114 272 Description:
115 273 This is a stimulus state that is waiting for an event indicating a Soft or
116 274 Scroll key has been pressed.
117 275
118 276 Resources Active:
119 277 The PLAY, RECORD, ERASE and EXT soft keys are displayed.
120 278 The Scroll keys may be displayed depending on the Arrow Status.
121 279
122 280 Critical Dependencies:
123 281 None.
124 282
125 283
126 284
127 285
128 286 STATE BPRH_MH1_AP_all_stimulus
129 287
130 288 EVENT PRH_I_SOFT_KEY_1 BPRH_snd_pac_play_req
131 289 FUNCTION BPRH_snd_pac_play_req
132 290 TRANS <BPRH_MH1_AP_all_stimulus>
133 291
134 292 EVENT PRH_I_SOFT_KEY_2 BPRH_snd_pac_record_req
135 293 FUNCTION BPRH_snd_pac_record_req
136 294 TRANS <BPRH_MH1_AP_wt_record_ack>
137 295
138 296 EVENT PRH_I_SOFT_KEY_3 BPRH_snd_pac_stop_req
139 297 FUNCTION BPRH_snd_pac_stop_req
140 298 TRANS <BPRH_MH1_AP_erase_confirm>
141 299
142 300 EVENT PRH_I_SOFT_KEY_4 BPRH_snd_nl_info_clear_display
143 301 FUNCTION BPRH_snd_pac_exit
144 302 TRANS <BPRH_MH1_AP_idle>
145 303
146 304 EVENT PRH_PA_INFO BPRH_save_pa_info
147 305 FUNCTION BPRH_check_pa_msg_status
148 306 TRANS <BPRH_MH1_AP_msg_status>
149 307
150 308 EVENT PRH_I_SCROLL_UP BPRH_snd_pac_prev_req
151 309 FUNCTION BPRH_snd_pac_prev_req
152 310 TRANS <BPRH_MH1_AP_all_stimulus>
153 311
154 312 EVENT PRH_I_SCROLL_DOWN BPRH_snd_pac_next_req
155 313 FUNCTION BPRH_snd_pac_next_req
156 314 TRANS <BPRH_MH1_AP_all_stimulus>
157 315
158 316
159 317
160 318
161 319
162 320
```

```
135 321 EVENT PRH_PA_FAILURE BPRH_snd_nl_info_pa_failure
136 322 FUNCTION BPRH_snd_nl_info_pa_failure
137 323 TRANS <BPRH_MH1_AP_failure_timeout>
138 324
139 325
140 326 EVENT PRH_I_PLUGIN BPRH_set_plugin
141 327 FUNCTION BPRH_snd_idh_plugin
142 328 TRANS <BPRH_MH1_AP_all_stimulus>
143 329
144 330
145 331
146 332
147 333
148 334
149 335
150 336 DONE BPRH_MH1_AP_all_stimulus.tde
151 337
152 338 /*****
153 339 State Name: BPRH_MH1_AP_erase_confirm
154 340
155 341 Description:
156 342 This is a stimulus state which is waiting for a soft key press indicating
157 343 whether a Personal Announcement should be erased, or not. Soft key 1 will
158 344 be treated as an affirmative response, and Soft key 4 as a cancellation
159 345 response.
160 346
161 347 Resources Active:
162 348 None.
163 349
164 350 Critical Dependencies:
165 351 None.
166 352
167 353
168 354
169 355
170 356 STATE BPRH_MH1_AP_erase_confirm
171 357
172 358 EVENT PRH_I_SOFT_KEY_1 BPRH_snd_pac_delete_req
173 359 FUNCTION BPRH_snd_pac_delete_req
174 360 TRANS <BPRH_MH1_AP_msg_status>
175 361
176 362 EVENT PRH_I_SOFT_KEY_4 BPRH_check_pa_msg_status
177 363 FUNCTION BPRH_check_pa_msg_status
178 364 TRANS <BPRH_MH1_AP_erase_confirm>
179 365
180 366 EVENT PRH_PA_FAILURE BPRH_snd_nl_info_pa_failure
181 367 FUNCTION BPRH_snd_nl_info_pa_failure
182 368 TRANS <BPRH_MH1_AP_failure_timeout>
183 369
184 370
185 371
186 372
187 373
188 374
189 375
190 376
191 377
192 378
193 379
194 380
195 381
196 382
197 383 DONE BPRH_MH1_AP_erase_confirm.tde
198 384
```

2103204

92/11/09
12:52:13

/tmp/61181

2103204

```
385 /*****
386
387 State Name: BPRH_MH1_AP_wt_sub
388
389 Description:
390 This is a stimulus state which is waiting for a submenu that it activated,
391 to exit. When the submenu exit event is received, the appropriate menu
392 will be displayed.
393
394 Resources Active:
395 None.
396
397 Critical Dependencies:
398 None.
399
400 *****/
401
402 STATE BPRH_MH1_AP_wt_sub
403 13
404 EVENT PPH_I_SUB_MENU_EXIT
405 FUNCTION BPRH_check_pa_msg_status
406 TRANS <BPRH_MH1_AP_msg_status>
407
408 DONE BPRH_MH1_AP_wt_sub.tde
409
410 ** SCCS UNPROCESSED COMMENTS BEGIN **
411 SCCS UNPROCESSED COMMENTS END **/
412
413 ** SCCS PROCESSED COMMENTS BEGIN **
414 03/09/92 corrigian File created.
415 04/24/92 corrigian Added enhancements based on re-inspection
416 05/05/92 corrigian Made changes per review.
417 08/05/92 kamdar Added BPRH_snd_pac_pa_failure_ack in
418 BPRH_MH1_AP_failure_timeout state.
419 SCCS PROCESSED COMMENTS END **/
```

92/11/09
12:52:01

/tmp/61103

```
1 1 STATE BPRH_MH1_AM_cb1_stm
28 37 STATE BPRH_MH1_AM_cb1_wait
48 64 STATE BPRH_MH1_AM_cb2_data
58 78 STATE BPRH_MH1_AM_cb2_stm
85 114 STATE BPRH_MH1_AM_cb3_stm
125 165 STATE BPRH_MH1_AM_cb4_stm
157 206 STATE BPRH_MH1_AM_cb5_stm
177 233 STATE BPRH_MH1_AM_cb_resume
182 241 STATE BPRH_MH1_AM_cb_suspend
188 250 STATE BPRH_MH1_AM_fwd_time
196 262 STATE BPRH_MH1_AM_fwd_endc
205 275 STATE BPRH_MH1_AM_idle
210 282 STATE BPRH_MH1_AM_suspend_data
223 301 STATE BPRH_MH1_AM_wt_nhc
```

2103204

92/11/09
12:52:00

/tmp/61101

2103204

1	1	STATE	BPRH_MH1_AM_cbl_stm	65
2	2			66
3	3	EVENT	PRH_I_SOFT_KEY_1	49
4	4	FUNCTION	BPRH_snd_vms_replay_req	50
5	5	TRANS	<BPRH_MH1_AM_cbl_stm>	51
6	6			52
7	7	EVENT	PRH_I_SOFT_KEY_2	53
8	8	FUNCTION	BPRH_snd_vms_redirect_req	71
9	9	FUNCTION	BPRH_snd_nl_info_avs_cbl_submenu	72
10	10	TRANS	<BPRH_MH1_AM_cbl_stm>	73
11	11			74
12	12	EVENT	PRH_I_SOFT_KEY_3	75
13	13	FUNCTION	BPRH_snd_vms_ent_num_req	57
14	14	TRANS	<BPRH_MH1_AM_wt_ndc>	76
15	15			77
16	16	EVENT	PRH_I_SOFT_KEY_4	78
17	17	FUNCTION	BPRH_snd_vms_done_req	79
18	18	TRANS	<BPRH_MH1_AM_idle>	80
19	19			81
20	20	EVENT	PRH_VMS_FAILURE	82
21	21	FUNCTION	BPRH_snd_l0_sub_menu_exit	83
22	22	TRANS	<BPRH_MH1_AM_idle>	84
23	23			85
24	24	EVENT	PRH_I_PLUGIN	86
25	25	FUNCTION	BPRH_set_plugin	87
26	26	TRANS	<BPRH_MH1_AM_cbl_stm>	88
27	27			89
28	28	EVENT	PRH_MENU_REFRESH	90
29	29	FUNCTION	BPRH_snd_nl_info_avs_cbl_submenu	91
30	30	TRANS	<BPRH_MH1_AM_cbl_stm>	92
31	31			93
32	32	EVENT	PRH_VMS_FAILURE	94
33	33	FUNCTION	BPRH_snd_l0_sub_menu_exit	95
34	34	TRANS	<BPRH_MH1_AM_idle>	96
35	35			97
36	36	EVENT	PRH_I_PLUGIN	98
37	37	FUNCTION	BPRH_set_plugin	99
38	38	TRANS	<BPRH_MH1_AM_cbl_stm>	100
39	39			101
40	40	EVENT	PRH_MENU_REFRESH	102
41	41	FUNCTION	BPRH_snd_nl_info_avs_cbl_submenu	103
42	42	TRANS	<BPRH_MH1_AM_cbl_stm>	104
43	43			105
44	44	EVENT	PRH_VMS_FAILURE	106
45	45	FUNCTION	BPRH_snd_l0_sub_menu_exit	107
46	46	TRANS	<BPRH_MH1_AM_idle>	108
47	47			109
48	48	EVENT	PRH_I_PLUGIN	110
49	49	FUNCTION	BPRH_set_plugin	111
50	50	TRANS	<BPRH_MH1_AM_cbl_stm>	112
51	51			113
52	52	EVENT	PRH_MENU_REFRESH	114
53	53	FUNCTION	BPRH_snd_nl_info_avs_cbl_submenu	115
54	54	TRANS	<BPRH_MH1_AM_cbl_wait>	116
55	55			117
56	56	EVENT	PRH_VMS_FAILURE	118
57	57	FUNCTION	BPRH_snd_l0_sub_menu_exit	119
58	58	TRANS	<BPRH_MH1_AM_idle>	120
59	59			121
60	60	EVENT	PRH_I_PLUGIN	122
61	61	FUNCTION	BPRH_set_plugin	123
62	62	TRANS	<BPRH_MH1_AM_cbl_wait>	124
63	63			125
64	64	EVENT	PRH_MENU_REFRESH	126
65	65	FUNCTION	BPRH_snd_nl_info_avs_wait_submenu	127
66	66	TRANS	<BPRH_MH1_AM_cbl_wait>	128
67	67			129
68	68	EVENT	PRH_VMS_FAILURE	130
69	69	FUNCTION	BPRH_snd_l0_sub_menu_exit	131
70	70	TRANS	<BPRH_MH1_AM_idle>	132
71	71			133
72	72	EVENT	PRH_I_PLUGIN	134
73	73	FUNCTION	BPRH_set_plugin	135
74	74	TRANS	<BPRH_MH1_AM_cbl_stm>	136
75	75			137
76	76	EVENT	PRH_MENU_REFRESH	138
77	77	FUNCTION	BPRH_snd_nl_info_avs_cbl_submenu	139
78	78	TRANS	<BPRH_MH1_AM_cbl_stm>	140
79	79			141
80	80	EVENT	PRH_VMS_FAILURE	142
81	81	FUNCTION	BPRH_snd_l0_sub_menu_exit	143
82	82	TRANS	<BPRH_MH1_AM_idle>	144
83	83			145
84	84	EVENT	PRH_I_PLUGIN	146
85	85	FUNCTION	BPRH_set_plugin	147
86	86	TRANS	<BPRH_MH1_AM_cbl_stm>	148
87	87			149
88	88	EVENT	PRH_MENU_REFRESH	150
89	89	FUNCTION	BPRH_snd_nl_info_avs_wait_submenu	151
90	90	TRANS	<BPRH_MH1_AM_cbl_wait>	152
91	91			153
92	92	EVENT	PRH_VMS_FAILURE	154
93	93	FUNCTION	BPRH_snd_l0_sub_menu_exit	155
94	94	TRANS	<BPRH_MH1_AM_idle>	156
95	95			157
96	96	EVENT	PRH_I_PLUGIN	158
97	97	FUNCTION	BPRH_set_plugin	159
98	98	TRANS	<BPRH_MH1_AM_cbl_stm>	160
99	99			161
100	100	EVENT	PRH_MENU_REFRESH	162
101	101	FUNCTION	BPRH_snd_nl_info_avs_cbl_submenu	163
102	102	TRANS	<BPRH_MH1_AM_cbl_stm>	164
103	103			165
104	104	EVENT	PRH_VMS_FAILURE	166
105	105	FUNCTION	BPRH_snd_l0_sub_menu_exit	167
106	106	TRANS	<BPRH_MH1_AM_idle>	168
107	107			169
108	108	EVENT	PRH_I_PLUGIN	170
109	109	FUNCTION	BPRH_set_plugin	171
110	110	TRANS	<BPRH_MH1_AM_cbl_stm>	172
111	111			173
112	112	EVENT	PRH_MENU_REFRESH	174
113	113	FUNCTION	BPRH_snd_nl_info_avs_cbl_submenu	175
114	114	TRANS	<BPRH_MH1_AM_cbl_stm>	176
115	115			177
116	116	EVENT	PRH_VMS_FAILURE	178
117	117	FUNCTION	BPRH_snd_l0_sub_menu_exit	179
118	118	TRANS	<BPRH_MH1_AM_idle>	180
119	119			181
120	120	EVENT	PRH_I_PLUGIN	182
121	121	FUNCTION	BPRH_set_plugin	183
122	122	TRANS	<BPRH_MH1_AM_cbl_stm>	184
123	123			185
124	124	EVENT	PRH_MENU_REFRESH	186
125	125	FUNCTION	BPRH_snd_nl_info_avs_wait_submenu	187
126	126	TRANS	<BPRH_MH1_AM_cbl_wait>	188
127	127			189
128	128	EVENT	PRH_VMS_FAILURE	190
129	129	FUNCTION	BPRH_snd_l0_sub_menu_exit	191
130	130	TRANS	<BPRH_MH1_AM_idle>	192
131	131			193
132	132	EVENT	PRH_I_PLUGIN	194
133	133	FUNCTION	BPRH_set_plugin	195
134	134	TRANS	<BPRH_MH1_AM_cbl_stm>	196
135	135			197
136	136	EVENT	PRH_MENU_REFRESH	198
137	137	FUNCTION	BPRH_snd_nl_info_avs_cbl_submenu	199
138	138	TRANS	<BPRH_MH1_AM_cbl_stm>	200
139	139			201
140	140	EVENT	PRH_VMS_FAILURE	202
141	141	FUNCTION	BPRH_snd_l0_sub_menu_exit	203
142	142	TRANS	<BPRH_MH1_AM_idle>	204
143	143			205
144	144	EVENT	PRH_I_PLUGIN	206
145	145	FUNCTION	BPRH_set_plugin	207
146	146	TRANS	<BPRH_MH1_AM_cbl_stm>	208
147	147			209
148	148	EVENT	PRH_MENU_REFRESH	210
149	149	FUNCTION	BPRH_snd_nl_info_avs_cbl_submenu	211
150	150	TRANS	<BPRH_MH1_AM_cbl_stm>	212
151	151			213
152	152	EVENT	PRH_VMS_FAILURE	214
153	153	FUNCTION	BPRH_snd_l0_sub_menu_exit	215
154	154	TRANS	<BPRH_MH1_AM_idle>	216
155	155			217
156	156	EVENT	PRH_I_PLUGIN	218
157	157	FUNCTION	BPRH_set_plugin	219
158	158	TRANS	<BPRH_MH1_AM_cbl_stm>	220
159	159			221
160	160	EVENT	PRH_MENU_REFRESH	222
161	161	FUNCTION	BPRH_snd_nl_info_avs_cbl_submenu	223
162	162	TRANS	<BPRH_MH1_AM_cbl_stm>	224
163	163			225
164	164	EVENT	PRH_VMS_FAILURE	226
165	165	FUNCTION	BPRH_snd_l0_sub_menu_exit	227
166	166	TRANS	<BPRH_MH1_AM_idle>	228
167	167			229
168	168	EVENT	PRH_I_PLUGIN	230
169	169	FUNCTION	BPRH_set_plugin	231
170	170	TRANS	<BPRH_MH1_AM_cbl_stm>	232
171	171			233
172	172	EVENT	PRH_MENU_REFRESH	234
173	173	FUNCTION	BPRH_snd_nl_info_avs_cbl_submenu	235
174	174	TRANS	<BPRH_MH1_AM_cbl_stm>	236
175	175			237
176	176	EVENT	PRH_VMS_FAILURE	238
177	177	FUNCTION	BPRH_snd_l0_sub_menu_exit	239
178	178	TRANS	<BPRH_MH1_AM_idle>	240
179	179			241
180	180	EVENT	PRH_I_PLUGIN	242
181	181	FUNCTION	BPRH_set_plugin	243
182	182	TRANS	<BPRH_MH1_AM_cbl_stm>	244
183	183			245
184	184	EVENT	PRH_MENU_REFRESH	246
185	185	FUNCTION	BPRH_snd_nl_info_avs_cbl_submenu	247
186	186	TRANS	<BPRH_MH1_AM_cbl_stm>	248
187	187			249
188	188	EVENT	PRH_VMS_FAILURE	250
189	189	FUNCTION	BPRH_snd_l0_sub_menu_exit	251
190	190	TRANS	<BPRH_MH1_AM_idle>	252
191	191			253
192	192	EVENT	PRH_I_PLUGIN	254
193	193	FUNCTION	BPRH_set_plugin	255
194	194	TRANS	<BPRH_MH1_AM_cbl_stm>	256
195	195			257
196	196	EVENT	PRH_MENU_REFRESH	258
197	197	FUNCTION	BPRH_snd_nl_info_avs_cbl_submenu	259
198	198	TRANS	<BPRH_MH1_AM_cbl_stm>	260
199	199			261
200	200	EVENT	PRH_VMS_FAILURE	262
201	201	FUNCTION	BPRH_snd_l0_sub_menu_exit	263
202	202	TRANS	<BPRH_MH1_AM_idle>	264
203	203			265
204	204	EVENT	PRH_I_PLUGIN	266
205	205	FUNCTION	BPRH_set_plugin	267
206	206	TRANS	<BPRH_MH1_AM_cbl_stm>	268
207	207			269
208	208	EVENT	PRH_MENU_REFRESH	270
209	209	FUNCTION	BPRH_snd_nl_info_avs_cbl_submenu	271
210	210	TRANS	<BPRH_MH1_AM_cbl_stm>	272
211	211			273
212	212	EVENT	PRH_VMS_FAILURE	274
213	213	FUNCTION	BPRH_snd_l0_sub_menu_exit	275
214	214	TRANS	<BPRH_MH1_AM_idle>	276
215	215			277
216	216	EVENT	PRH_I_PLUGIN	278
217	217	FUNCTION	BPRH_set_plugin	279
218	218	TRANS	<BPRH_MH1_AM_cbl_stm>	280
219	219			281
220	220	EVENT	PRH_MENU_REFRESH	282
221	221	FUNCTION	BPRH_snd_nl_info_avs_cbl_submenu	283
222	222	TRANS	<BPRH_MH1_AM_cbl_stm>	284
223	223			285
224	224	EVENT	PRH_VMS_FAILURE	286
225	225	FUNCTION	BPRH_snd_l0_sub_menu_exit	287
226	226	TRANS	<BPRH_MH1_AM_idle>	288
227	227			289
228	228	EVENT	PRH_I_PLUGIN	290
229	229	FUNCTION	BPRH_set_plugin	291
230	230	TRANS	<BPRH_MH1_AM_cbl_stm>	292
231	231			293
232	232	EVENT	PRH_MENU_REFRESH	294
233	233	FUNCTION	BPRH_snd_nl_info_avs_cbl_submenu	295
234	234	TRANS	<BPRH_MH1_AM_cbl_stm>	296
235	235			297
236	236	EVENT	PRH_VMS_FAILURE	298
237	237	FUNCTION	BPRH_snd_l0_sub_menu_exit	299
238	238	TRANS	<BPRH_MH1_AM_idle>	300
239	239			301
240	240	EVENT	PRH_I_PLUGIN	302
241	241	FUNCTION	BPRH_set_plugin	303
242	242	TRANS	<BPRH_MH1_AM_cbl_stm>	304
243	243			305
244	244	EVENT	PRH_MENU_REFRESH	306
245	245	FUNCTION	BPRH_snd_nl_info_avs_cbl_submenu	307
246	246	TRANS	<BPRH_MH1_AM_cbl_stm>	308
247	247			309
248	248	EVENT	PRH_VMS_FAILURE	310
249	249	FUNCTION	BPRH_snd_l0_sub_menu_exit	311
250	250	TRANS	<BPRH_MH1_AM_idle>	312
251	251			313
252	252	EVENT	PRH_I_PLUGIN	314
253	253	FUNCTION	BPRH_set_plugin	315
254	254	TRANS	<BPRH_MH1_AM_cbl_stm>	316
255	255			317
256	256	EVENT	PRH_MENU_REFRESH	318
257	257	FUNCTION	BPRH_snd_nl_info_avs_cbl_submenu	319
258	258	TRANS	<BPRH_MH1_AM_cbl_stm>	320
259	259			321
260	260	EVENT	PRH_VMS_FAILURE	322
261	261	FUNCTION	BPRH_snd_l0_sub_menu_exit	323
262	262	TRANS	<BPRH_MH1_AM_idle>	324
263	263			325
264	264	EVENT	PRH_I_PLUGIN	326
265	265	FUNCTION	BPRH_set_plugin	327
266	266	TRANS	<BPRH_MH1_AM_cbl_stm>	328
267	267			329
268	268	EVENT	PRH_MENU_REFRESH	330
269	269	FUNCTION	BPRH_snd_nl_info_avs_cbl_submenu	331
270	270	TRANS	<BPRH_MH1_AM_cbl_stm>	332
271	271			333
272	272	EVENT	PRH_VMS_FAILURE	334
273	273	FUNCTION	BPRH_snd_l0_sub_menu_exit	335
274	274	TRANS	<BPRH_MH1_AM_idle>	336
275	275			337
276	276	EVENT	PRH_I_PLUGIN	338
277	277	FUNCTION	BPRH_set_plugin	339
278	278	TRANS	<BPRH_MH1_AM_cbl_stm>	340
279	279			341
280	280	EVENT	PRH_MENU_REFRESH	342
281	281	FUNCTION	BPRH_snd_nl_info_avs_cbl_submenu	343

92/11/09
12:52:08

/tmp/61101

```
97 129 FUNCTION BPRH_snd_vms_dial_req
98 130 FUNCTION BPRH_disable_digit_collector
99 131 FUNCTION BPRH_set_ack_timer
100 132 TRANS <BPRH_MH1_AM_sus_data>
101 133
102 134 PRH_I_SOFT_KEY_4
103 135 FUNCTION BPRH_snd_vms_done_req
104 136 FUNCTION BPRH_disable_digit_collector
105 137 FUNCTION BPRH_snd_nl_info_avs_cbl_submenu
106 138 TRANS <BPRH_MH1_AM_cbl_stm>
107 139
108 140 PRH_CALLBACK_DATA
109 141 FUNCTION BPRH_save_callback_data
110 142 TRANS <BPRH_MH1_AM_cb3_stm>
111 143
112 144 PRH_DIGIT_TIMEOUT
113 145 FUNCTION BPRH_disable_digit_collector
114 146 FUNCTION BPRH_snd_nl_info_avs_cbl_submenu
115 147 TRANS <BPRH_MH1_AM_cbl_stm>
116 148
117 149 PRH_VMS_FAILURE
118 150 FUNCTION BPRH_snd_10_sub_menu_exit
119 151 TRANS <BPRH_MH1_AM_idle>
120 152
121 153 PRH_I_PLUGIN
122 154 FUNCTION BPRH_set_plugin
123 155 FUNCTION BPRH_snd_ldh_plugin
124 156 FUNCTION BPRH_snd_nl_info_avs_cb3_submenu
125 157 TRANS <BPRH_MH1_AM_cb3_stm>
126 158
127 159 PRH_MENU_REFRESH
128 160 FUNCTION BPRH_snd_nl_info_avs_cb3_submenu
129 161 TRANS <BPRH_MH1_AM_cb3_stm>
130 162
131 163 DONE BPRH_MH1_AM_cb3_stm.tde
132 164
133 165 STATE BPRH_MH1_AM_cb4_stm
134 166
135 167 PRH_CALLBACK_VALID_Q
136 168 FUNCTION BPRH_disable_digit_collector
137 169 FUNCTION BPRH_snd_10_sub_menu_exit
138 170 TRANS <BPRH_MH1_AM_idle>
139 171
140 172 PRH_CALLBACK_INVALID_Q
141 173 FUNCTION BPRH_disable_digit_collector
142 174 FUNCTION BPRH_snd_nl_info_avs_fwd_invalid_entry
143 175 FUNCTION BPRH_set_display_timer
144 176 TRANS <BPRH_MH1_AM_fwd_time>
145 177
146 178 PRH_I_SOFT_KEY_4
147 179 FUNCTION BPRH_disable_digit_collector
148 180 FUNCTION BPRH_snd_vms_done_req
149 181 FUNCTION BPRH_snd_nl_info_avs_cb2_submenu
150 182 TRANS <BPRH_MH1_AM_cb2_stm>
151 183
152 184 PRH_DIGIT_TIMEOUT
153 185 FUNCTION BPRH_disable_digit_collector
154 186 FUNCTION BPRH_snd_nl_info_avs_fwd_invalid_entry
155 187 FUNCTION BPRH_set_display_timer
156 188 TRANS <BPRH_MH1_AM_fwd_time>
157 189
158 190 PRH_VMS_FAILURE
159 191 FUNCTION BPRH_snd_10_sub_menu_exit
160 192 TRANS <BPRH_MH1_AM_idle>
```

```
148 193
149 194 PRH_I_PLUGIN
150 195 FUNCTION BPRH_set_plugin
151 196 FUNCTION BPRH_snd_ldh_plugin
152 197 FUNCTION BPRH_snd_nl_info_avs_cb4_submenu
153 198 TRANS <BPRH_MH1_AM_cb4_stm>
154 199
155 200 PRH_MENU_REFRESH
156 201 FUNCTION BPRH_snd_nl_info_avs_cb4_submenu
157 202 TRANS <BPRH_MH1_AM_cb4_stm>
158 203
159 204 DONE BPRH_MH1_AM_cb4_stm.tde
160 205
161 206 STATE BPRH_MH1_AM_cb5_stm
162 207
163 208 PRH_I_SOFT_KEY_1
164 209 FUNCTION BPRH_snd_vms_delete_req
165 210 FUNCTION BPRH_snd_10_sub_menu_exit
166 211 TRANS <BPRH_MH1_AM_idle>
167 212
168 213 PRH_I_SOFT_KEY_4
169 214 FUNCTION BPRH_snd_nl_info_avs_cb2_submenu
170 215 TRANS <BPRH_MH1_AM_cb2_stm>
171 216
172 217 PRH_VMS_FAILURE
173 218 FUNCTION BPRH_snd_10_sub_menu_exit
174 219 TRANS <BPRH_MH1_AM_idle>
175 220
176 221 PRH_I_PLUGIN
177 222 FUNCTION BPRH_set_plugin
178 223 FUNCTION BPRH_snd_ldh_plugin
179 224 FUNCTION BPRH_snd_nl_info_avs_cb5_submenu
180 225 TRANS <BPRH_MH1_AM_cb5_stm>
181 226
182 227 PRH_MENU_REFRESH
183 228 FUNCTION BPRH_snd_nl_info_avs_cb5_submenu
184 229 TRANS <BPRH_MH1_AM_cb5_stm>
185 230
186 231 DONE BPRH_MH1_AM_cb5_stm.tde
187 232
188 233 STATE BPRH_MH1_AM_cb_resume
189 234
190 235 PRH_MENU_RESUME
191 236 FUNCTION BPRH_snd_nl_info_avs_cbl_submenu
192 237 TRANS <BPRH_MH1_AM_cbl_stm>
193 238
194 239 DONE BPRH_MH1_AM_cb_resume.tde
195 240
196 241 STATE BPRH_MH1_AM_cb_susp
197 242
198 243 PRH_MENU_SUSPEND
199 244 FUNCTION BPRH_snd_nl_info_clear_display
200 245 FUNCTION BPRH_toggle_active_menu
201 246 TRANS <BPRH_MH1_AM_cb_resume>
202 247
203 248 DONE BPRH_MH1_AM_cb_susp.tde
204 249
205 250 STATE BPRH_MH1_AM_fwd_time
206 251
207 252 PRH_I_TIMEOUT
208 253 FUNCTION BPRH_snd_nl_info_avs_cb2_submenu
209 254 TRANS <BPRH_MH1_AM_cb2_stm>
210 255
211 256 PRH_VMS_FAILURE
```

2103204

92/11/09
12:52:00

/tmp/61101

```
193 257 FUNCTION BPRH_snd_10_sub_menu_exit
194 258 TRANS <BPRH_MH1_AM_idle>
195 259
196 260 DONE BPRH_MH1_AM_fwd_time.tde
197 261
198 262 STATE BPRH_MH1_AM_fwd_time.tde
199 263
200 264 EVENT PRH_NDC_ACTIVATED
201 265 FUNCTION BPRH_avs_cb4_enable_digit_collector
202 266 FUNCTION BPRH_snd_n1_info_avs_cb4_submenu
203 267 TRANS <BPRH_MH1_AM_cb4_stm>
204 268
205 269 EVENT PRH_VMS_FAILURE
206 270 FUNCTION BPRH_snd_10_sub_menu_exit
207 271 TRANS <BPRH_MH1_AM_idle>
208 272
209 273 DONE BPRH_MH1_AM_fwd_time.tde
210 274
211 275 STATE BPRH_MH1_AM_idle
212 276
213 277 EVENT PRH_MH1_AM_ACTIVATED
214 278 FUNCTION BPRH_snd_n1_info_avs_wait_submenu
215 279 TRANS <BPRH_MH1_AM_cb1_wait>
216 280
217 281 DONE BPRH_MH1_AM_idle.tde
218 282
219 283 STATE BPRH_MH1_AM_suspend_data
220 284
221 285 EVENT PRH_CALLBACK_DATA
222 286 FUNCTION BPRH_save_callback_data
223 287 FUNCTION BPRH_cancel_ack_timer
224 288 TRANS <BPRH_MH1_AM_cancel_suspend>
225 289
226 290 EVENT PRH_VMS_FAILURE
227 291 FUNCTION BPRH_snd_10_sub_menu_exit
228 292 TRANS <BPRH_MH1_AM_idle>
229 293
230 294 EVENT PRH_DIGIT_TIMEOUT
231 295 FUNCTION BPRH_cancel_ack_timer
232 296 FUNCTION BPRH_snd_n1_info_avs_cb1_submenu
233 297 TRANS <BPRH_MH1_AM_cb1_stm>
234 298
235 299 DONE BPRH_MH1_AM_suspend_data.tde
236 300
237 301 STATE BPRH_MH1_AM_waiting_data
238 302
239 303 EVENT PRH_NDC_ACTIVATED
240 304 FUNCTION BPRH_avs_cb3_cont_enable_digit_collector
241 305 FUNCTION BPRH_snd_n1_info_avs_cb3_submenu
242 306 TRANS <BPRH_MH1_AM_cb3_stm>
243 307
244 308 EVENT PRH_VMS_FAILURE
245 309 FUNCTION BPRH_snd_10_sub_menu_exit
246 310 TRANS <BPRH_MH1_AM_idle>
247 311
248 312 DONE BPRH_MH1_AM_waiting_data.tde
249 313
250 314
251 315
252 316
253 317
254 318
255 319 03/10/92 Corrigan File created.
256 320 04/20/92 Corrigan Added changes for RC per PRAM 13567.
```

/** SCCS UNPROCESSED COMMENTS BEGIN **
SCCS UNPROCESSED COMMENTS END **/

/** SCCS PROCESSED COMMENTS BEGIN **

added PRH_DIGIT_TIMEOUT event in BPRH_MH1_AM_cb1_
Added the BPRH_snd_vms_play_stop_req func to
BPRH_MH1_AM_cb2_stm.tde state to stop msg play.

SCCS PROCESSED COMMENTS END **/

321 08/13/92 chung
322 08/25/92 lavigne
323
324
325

2103204

92/11/09
12:21:14

/tmp/59783

2103204

```
302 1520 STATE AMD_chk_if_download
360 754 STATE AMD_chk_if_idle_pseudo
357 548 STATE AMD_chk_if_upload
46 142 STATE AMD_chk_msg_num
380 597 STATE AMD_chk_msg_status
322 1574 STATE AMD_chk_msgq_pbk_tag_ld_msg
409 850 STATE AMD_chk_msgq_playbk_tag
22 97 STATE AMD_chk_msgq_record_tag
425 901 STATE AMD_chk_tn_only
2 48 STATE AMD_idle
334 1405 STATE AMD_move_file_to_new_queue
345 1096 STATE AMD_play
308 1031 STATE AMD_play_wt_port
83 229 STATE AMD_rec_wait_port
147 343 STATE AMD_record
377 1296 STATE AMD_rmv_deall_msg_ack
70 1352 STATE AMD_rmv_msg_from_disk
65 186 STATE AMD_tn_chk_msg_num
102 18 STATE AMD_wait_record
71 170 STATE AMD_wt_load_comp
314 1202 STATE AMD_wt_next_req
460 954 STATE AMD_wt_next_tn_req
374 789 STATE AMD_wt_playback
384 411 STATE AMD_wt_stop_rec
319 685 STATE AMD_wt_update_tn_comp
220 481 STATE AMD_wt_upload_comp
196 638 STATE AMD_wt_upload_tn
```

REL C ASCII STATE MACHINE

92/11/09
12:21:13

/tmp/59781

2103204

```
1 /*****
2
3
4 REL 2 ASCII STATE MACHINE
5
6 ****
7
8
9 State Machine File Prologue
10
11 ****
12 /*****
13
14
15 ACD-MAIL LDH (AMD) State Machine
16
17 SCI of DLD: G3KE-09-0192
18
19 Description:
20 The ACD-MAIL pseudo LDH controls the playing, recording,
21 uploading and downloading of the voice message and/or telephone
22 number (TN).
23
24 Design Notes of Interest:
25 ****
26
27
28
29
30 /*****
31 *
32 * AMD_idle
33 *
34 *
35 * Description:
36 * This state indicates the LDH is ready to process requests.
37 *
38 * Resources Active:
39 * none
40 *
41 * Critical Dependencies:
42 * Pseudo port is mapped by Pseudo LDH for ACD-Mail if the
43 * vector or AMG activates the AMD when recording a message
44 * or when AMG decides to pre-transfer a message. This is the
45 * initial state of AMD, which is mapped in by SDOG in this state.
46 ****
47
48 STATE AMD_idle
49
50 EVENT LDH_ACTIVATE /* a voice message */
51 FUNCTION AMD_init
52 FUNCTION AMD_save_rec_req_info
53 FUNCTION AMD_chk_msgq_rec_tag
54 TRANS <AMD_chk_msgq_rec_tag>
55
56 EVENT LDH_ACTIVATE_TN /* a TN only type of message */
57 FUNCTION AMD_init
58 FUNCTION AMD_save_rec_req_info
59 FUNCTION AMD_get_msg_num
60 TRANS <AMD_chk_chk_msg_num>
61
62 EVENT LDH_PRE_TRANS
63 FUNCTION AMD_init
64 FUNCTION AMD_save_prr_rec_info
65 FUNCTION AMD_chk_chk_msgq_rec_tag
66 TRANS <AMD_chk_chk_msgq_rec_tag>
67
68
69 EVENT ANY_STABLE_CALL
70 FUNCTION AMD_chk_chk_msgq_rec_tag
71 TRANS <AMD_chk_chk_msgq_rec_tag>
72
73 EVENT ANY_FORCE_OOS
74 FUNCTION AMD_chk_chk_msgq_rec_tag
75 TRANS <AMD_chk_chk_msgq_rec_tag>
76
77 DONE
78
```

92/11/09
12:21:13

/tmp/59781

2103204

```

79 /*****
80 *
81 * AMD_chk_msgq_record_tag
82 *
83 *
84 * Description:
85 * In this transition state the message queue record TAG ID is
86 * checked to see if one exists. If so a message number will be
87 * obtained.
88 *
89 * Resources Active:
90 * None.
91 *
92 * Critical Dependencies:
93 * None.
94 *
95 *****/
96
97 STATE AMD_chk_msgq_record_tag
98
99 EVENT AMD_I_REC_TAG_ID_OK
100 FUNCTION AMD_get_msg_num
101 TRANS <AMD_chk_msg_num>
102
103 EVENT AMD_I_NO_REC_TAG_ID
104 FUNCTION AMD_mnt_inform_rec_tag_id
105 FUNCTION AMD_send_any_activate_nack /* to vector */
106 FUNCTION AMD_2sdog_send_lgh_idle
107 TRANS <AMD_idle>
108
109 EVENT ANY_STABLE_CALL
110 FUNCTION AMD_2tmc_any_unstable_ack
111 TRANS <AMD_chk_msgq_record_tag>
112
113 EVENT ANY_FORCE_OOS
114 FUNCTION AMD_send_any_activate_nack /* to vector */
115 FUNCTION AMD_set_status_to_free
116 FUNCTION AMD_2sdog_send_lgh_idle
117 TRANS <AMD_idle>
118
119 DONE AMD_chk_msgq_record_tag.tde
120
121 /*****
122 *
123 * AMD_chk_msg_num
124 *
125 *
126 * Description:
127 * This transition state determines whether or not a new system
128 * message number can be allocated. Upon the successful allocation
129 * of system message number AMD responds to the requestor with
130 * ANY_MESSAGE_NUM and acquires DAS memory and a port, otherwise,
131 * AMD responds to the request with ANY_ACTIVATE_NACK.
132 *
133 * Resources Active:
134 * System message number, if successful.
135 *
136 * Critical Dependencies:
137 * ACD-Mail message type is not known until the vector sends an
138 * event to record, LDH_RECORD or LDH_RECORD_TN.
139 *
140 *****/
141
142 STATE AMD_chk_msg_num
143
144 EVENT AMD_I_MSG_NUM
145 FUNCTION AMD_send_any_message_num /* to vector */
146 FUNCTION AMD_2tag_lgh_acquire_mem
147 TRANS <AMD_rec_wait_port>
148
149 EVENT AMD_I_NO_MSG_NUM
150 FUNCTION AMD_send_any_activate_nack /* to vector */
151 FUNCTION AMD_2sdog_send_lgh_idle
152 TRANS <AMD_idle>
153
154 EVENT ANY_STABLE_CALL
155 FUNCTION AMD_2tmc_any_unstable_ack
156 TRANS <AMD_chk_msg_num>
157
158 EVENT ANY_FORCE_OOS
159 FUNCTION AMD_send_any_activate_nack /* to vector */
160 FUNCTION AMD_set_status_to_free
161 FUNCTION AMD_2sdog_send_lgh_idle
162 TRANS <AMD_idle>
163
164 DONE AMD_chk_msg_num.tde
165
166
```

92/11/89
12:21:13

/tmp/59781

```
166 /*****
167 *
168 * AMD_tn_chk_msg_num
169 *
170 *
171 * Description:
172 * This transition state determines whether or not a new system
173 * message number can be allocated. Upon the successful allocation
174 * of a system message number AMD responds to the requester with
175 * the ANY_MESSAGE_NUM event, otherwise, the ANY_ACTIVATE_NACK
176 * event is sent.
177 *
178 * Resources Active:
179 * System message number, if successful.
180 *
181 * Critical Dependencies:
182 * None.
183 *
184 *****/
185
55 STATE AMD_tn_chk_msg_num
187
56 EVENT AMD_1 MSG_NUM
188 FUNCTION AMD_set_msgnum_to_tn_only
57 FUNCTION AMD_send_any_message_num /* to vector */
58
59 TRANS <AMD_wt_upload_tn>
190
191
192
193 EVENT AMD_1 NO_MSG_NUM
194 FUNCTION AMD_send_any_activate_nack /* to vector */
195 <AMD_idle>
196
197
198 EVENT ANY_STABLE_CALL
199 FUNCTION <AMD_tn_chk_msg_num>
200
201
202 EVENT ANY_FORCE_OOS
203 FUNCTION AMD_send_any_activate_nack /* to vector */
204 FUNCTION AMD_set_status_to_free
205 FUNCTION AMD_2sdog_send_lgh_idle
206 <AMD_idle>
207
81 DONE AMD_tn_chk_msg_num.tde
8:
```

```
209 /*****
210 *
211 * AMD_rec_wait_port
212 *
213 *
214 * Description:
215 * A request has been made to acquire a DAS port and DAS memory,
216 * and AMD is waiting for a response from TAG LDH.
217 *
218 * Resources Active:
219 * Acquiring DAS port and DAS memory.
220 *
221 * Critical Dependencies:
222 * TAG LGH is handling the cancellation of the DAS port even if the
223 * TAG LDH has been activated and is allocating the DAS memory.
224 * The vector does the timing while waiting for the resources.
225 *
226 *****/
227
228 STATE AMD_rec_wait_port
229
230
231 EVENT ANY_ACTIVATE_ACK
232 FUNCTION AMD_save_das_port_id
233 FUNCTION AMD_send_any_activate_ack /* to vector */
234 <AMD_wait_record>
235
236
237 EVENT ANY_ACTIVATE_NACK
238 FUNCTION AMD_send_any_activate_nack /* to vector */
239 FUNCTION AMD_set_status_to_free
240 FUNCTION AMD_2sdog_lgh_idle
241 <AMD_idle>
242
243
244 EVENT LDH_IDLE /* vector timeout waiting for the resources */
245 FUNCTION AMD_2tag_lgh_cancel_acquire
246 FUNCTION AMD_set_status_to_free
247 FUNCTION AMD_2sdog_lgh_idle
248 <AMD_idle>
249
250
251 EVENT ANY_STABLE_CALL
252 FUNCTION AMD_2tmc_any_unstable_ack
253 <AMD_rec_wait_port>
254
255
256 EVENT ANY_FORCE_OOS
257 FUNCTION AMD_send_any_activate_nack /* to vector */
258 FUNCTION AMD_2tag_lgh_cancel_acquire
259 FUNCTION AMD_set_status_to_free
260 FUNCTION AMD_2sdog_send_lgh_idle
261 <AMD_idle>
262
263
264 STATE DONE
265 AMD_rec_wait_port.tde
266
```

2103204

92/11/89
12:21:13

/tmp/59781

```
261 /*****
262 *
263 * AMD_wait_record
264 *
265 *
266 * Description:
267 * This state indicates the DAS port is ready to record a message.
268 *
269 * Resources Active:
270 * das_port and DAS memory
271 *
272 * Critical Dependencies:
273 * None.
274 *
275 *****/
276
277
278 STATE AMD_wait_record
279
280 EVENT LDH_RECORD /* this message contains only the voice */
281 FUNCTION AMD_save_flash_list
282 FUNCTION AMD_tag_ldh_record
283 FUNCTION AMD_set_status_to_record
284 FUNCTION AMD_set_msgtyp_to_voice_only
285 TRANS <AMD_record>
286
287 EVENT LDH_RECORD /* this message contains both voice and TN */
288 FUNCTION AMD_save_flash_list_tn
289 FUNCTION AMD_save_tn_digits
290 FUNCTION AMD_tag_ldh_record
291 FUNCTION AMD_set_status_to_record
292 FUNCTION AMD_set_msgtyp_to_voice_tn
293 TRANS <AMD_record>
294
295 EVENT LDH_IDLE /* agent answer the phone while recording */
296 FUNCTION AMD_tag_ldh_idle
297 FUNCTION AMD_send_das_deallocate_noresp /* to DAS applic
298 FUNCTION AMD_set_status_to_free
299 FUNCTION AMD_2sdog_igh_idle
300 TRANS <AMD_idle>
301
302 EVENT ANY_MSG_FAILURE
303 FUNCTION AMD_send_any_sdo_failure /* to vector */
304 FUNCTION AMD_send_das_deallocate_noresp /* to DAS applic
305 FUNCTION AMD_set_status_to_free
306 FUNCTION AMD_2sdog_igh_idle
307 TRANS <AMD_idle>
308
309 EVENT ANY_STABLE_CALL
310 FUNCTION AMD_2tmc_any_unstable_ack
311 TRANS <AMD_wait_record>
312
313 EVENT ANY_FORCE_OOS
314 FUNCTION AMD_send_any_sdo_failure /* to vector */
315 FUNCTION AMD_tag_ldh_idle
316 FUNCTION AMD_send_das_deallocate_noresp /* to DAS applic
317 FUNCTION AMD_set_status_to_free
318 FUNCTION AMD_2sdog_send_igh_idle
319 TRANS <AMD_idle>
320
321 DONE
322
323 /*****
324 *
325 * AMD_record
326 *
327 *
328 * Description:
329 * This state indicates that the vector has initiated the recording
330 * process and is waiting for a stop record event from the vector
331 * or an exceed message event from the DAS application.
332 *
333 * Resources Active:
334 * das_port and DAS memory
335 *
336 * Critical Dependencies:
337 * The BMU uploader updates the status of msg_ctl_tbl, indicating
338 * whether it's uploading or not.
339 *
340 *****/
341
342
343 STATE AMD_record
344
345 EVENT LDH_STOP_RECORD
346 FUNCTION AMD_tag_ldh_stop_record
347 TRANS <AMD_wait_record>
348
349 EVENT ANY_EXCEED_MSG_SIZE
350 FUNCTION AMD_send_any_exceed_msg_size /* to vector */
351 FUNCTION AMD_tag_ldh_idle
352 FUNCTION AMD_send_das_upload_hdr /* to BMU uploader */
353 FUNCTION AMD_start_upload_tmr
354 TRANS <AMD_wait_upload_como>
355
356 EVENT LDH_IDLE_DELETE /* agent answer the phone while recording */
357 FUNCTION AMD_tag_ldh_stop_record
358 TRANS <AMD_record>
359
360 /* received LDH_IDLE DELETE from vector which means an agent */
361 /* has answer in middle of recording. */
362 EVENT ANY_STOP_RECORD_ACK /* get rid of resources */
363 FUNCTION AMD_tag_ldh_idle
364 FUNCTION AMD_send_das_deallocate_noresp /* to DAS applic
365 FUNCTION AMD_set_status_to_free
366 FUNCTION AMD_2sdog_igh_idle
367 TRANS <AMD_idle>
368
369 EVENT ANY_MSG_FAILURE
370 FUNCTION AMD_send_any_sdo_failure /* to vector */
371 FUNCTION AMD_set_status_to_free
372 FUNCTION AMD_send_das_deallocate_noresp /* to DAS applic
373 FUNCTION AMD_2sdog_igh_idle
374 TRANS <AMD_idle>
375
376 EVENT ANY_STABLE_CALL
377 FUNCTION AMD_2tmc_any_unstable_ack
378 TRANS <AMD_record>
379
380 EVENT ANY_FORCE_OOS
381 FUNCTION AMD_send_any_sdo_failure /* to vector */
382 FUNCTION AMD_tag_ldh_idle
383 FUNCTION AMD_set_status_to_free
384 FUNCTION AMD_send_das_deallocate_noresp /* to DAS applic
385 FUNCTION AMD_2sdog_send_igh_idle
```

2103204

92/11/09
12:21:13

181 386 TRANS <amd_idle>
182 387
183 388 DONE AMD_record.tde
184 389

/tmp/59781

```
390 /*****
391 *
392 *
393 * AMD_wt_stop_rec
394 *
395 * Description:
396 * This state indicates that the vector has stopped the recording
397 * process and is waiting for a stop record ack from TAG LDH.
398 *
399 * Resources Active:
400 *   das_port and DAS memory
401 *
402 * Critical Dependencies:
403 *   AMG should send either an idle or a ready event to indicate
404 *   whether the message should be removed from DAS. The BMU
405 *   uploader updates the status of msg_ctl_tbl, indicating whether
406 *   it's uploading or not.
407 *
408 * *****/
409
410
411 STATE AMD_wt_stop_rec
412
413 EVENT ANY_STOP_RECORD_ACK
414   FUNCTION AMD_2tag_ldh_idle
415   FUNCTION AMD_send_das_upload_hdr
416   FUNCTION AMD_start_upload_amd_tm
417   TRANS <AMD_wt_upload_comp>
418
419 EVENT LDH_IDLE
420   FUNCTION AMD_set_i_event
421   TRANS <AMD_wt_stop_rec>
422
423 EVENT LDH_READY
424   FUNCTION AMD_set_i_event
425   TRANS <AMD_wt_stop_rec>
426
427 /* Either RC requested to delete the queue or an agent answer the */
428 /* while recording */
429 EVENT LDH_IDLE_DELETE
430   FUNCTION AMD_set_status_to_free
431   FUNCTION AMD_2tag_ldh_idle
432   FUNCTION AMD_send_das_deallocate_noresp /* to DAS applic
433   FUNCTION AMD_2sdog_ldh_idle
434   TRANS <AMD_idle>
435
436 EVENT ANY_MSG_FAILURE
437   FUNCTION AMD_set_status_to_free
438
439 /* the following function may or may not be appropriate
440 need new function to check if I event set first?? */
441 FUNCTION AMD_2amg_ldh_remove_msg
442
443 FUNCTION AMD_send_das_deallocate_noresp /* to DAS applic
444   FUNCTION AMD_2sdog_ldh_idle
445   TRANS <AMD_idle>
446
447 EVENT ANY_STABLE_CALL
448   FUNCTION AMD_2tmc_any_unstable_ack
449   TRANS <AMD_wt_stop_rec>
450
451 EVENT ANY_FORCE_OCS
452   FUNCTION AMD_2tag_ldh_idle
```

2103204

92/11/09
12:21:13

/tmp/59781

13 453
14 454
15 455
16 456
17 457
18 458
19 459 DONE
20 460

FUNCTION
FUNCTION
FUNCTION
FUNCTION
TRANS <AMD_idle>
AMD_wt_stop_rec.tde

AMD_set_status_to_free
AMD_2amg_lgh_remove_mesg
AMD_send_das_deallocate_noresp /* to DAS applic
AMD_2sdog_send_lgh_idle

461 /*****
462 *
463 *
464 * AMD_wt_upload_comp
465 *
466 * Description:
467 * This state indicates the that recorded message is currently on
468 * the disk and is waiting for a response from message uploader.
469 *
470 * Resources Active:
471 * DAS memory, timer waiting for upload complete.
472 *
473 * Critical Dependencies:
474 * AMG should send either an idle or a ready event to indicate
475 * whether the message should be removed from the DAS or not.
476 *
477 *****/

220 481 STATE AMD_wt_upload_comp
221 482
222 483
223 484
224 485
225 486
226 487
227 488
228 489
229 490
230 491
231 492
232 493
233 494
234 495
235 496
236 501
237 502
238 503
239 504
240 505
241 506
242 507
243 508
244 509
245 510
246 511
247 512
248 513
249 514
250 515
251 516
252 517
253 518
254 519
255 520
256 521
257 522
258 523

EVENT LDH_UPLOAD_COMPLETE
FUNCTION AMD_chk_status_rdy_play
FUNCTION AMD_stop_timer
FUNCTION AMD_2amg_lgh_upload_comp
FUNCTION AMD_gen_i_event
TRANS <AMD_chk_msg_status>

EVENT LDH_IDLE
FUNCTION AMD_set_i_event
TRANS <AMD_wt_upload_comp>

EVENT LDH_READY
FUNCTION AMD_set_i_event
TRANS <AMD_wt_upload_comp>

EVENT ANY_UPLOAD_FAILURE
FUNCTION AMD_stop_timer
FUNCTION AMD_chk_upload_flag
TRANS <AMD_chk_if_upload>

EVENT AMD_I_UPLOAD_COMPLETE
FUNCTION AMD_mnt_inform_upload_timeout
FUNCTION AMD_set_status_to_free
FUNCTION AMD_clr_ready_to_play
FUNCTION AMD_2amg_lgh_remove_mesg
FUNCTION AMD_send_das_deallocate_noresp /* to DAS applic
FUNCTION AMD_2sdog_lgh_idle
TRANS <AMD_idle>

EVENT ANY_STABLE_CALL
FUNCTION AMD_2tmc_any_unstable_ack
TRANS <AMD_wt_upload_comp>

EVENT ANY_FORCE_OOS
FUNCTION AMD_stop_timer
FUNCTION AMD_2amg_lgh_remove_mesg
FUNCTION AMD_send_das_deallocate_noresp /* to DAS applic
FUNCTION AMD_chk_to_set_free
FUNCTION AMD_2sdog_send_lgh_idle
TRANS <AMD_idle>

2103204

92/11/09
12:21:13

55 524 DONE AMD_wt_upload_comp.tde
56 525

/tmp/59781

```
526 /*****  
527 *  
528 *  
529 * AMD_chk_if_upload  
530 *  
531 * Description:  
532 * This state indicates that the message has been recorded and  
533 * uploading failed. It will decide to either deallocate the DAS  
534 * memory or upload the message again depending on the internal  
535 * event it receives.  
536 *  
537 * Resources Active:  
538 * DAS memory.  
539 *  
540 * Critical Dependencies:  
541 * The BMU uploader updates the status of msg_ctl_tbl, indicating  
542 * whether it's uploading or not.  
543 *  
544 *****/  
545  
546  
547  
548 STATE AMD_chk_if_upload  
549  
550 EVENT AMD_I_UPLOAD  
551 FUNCTION AMD_send_das_upload_hdr  
552 FUNCTION AMD_start_upload_comp_tm  
553 TRANS <AMD_wt_upload_comp>  
554  
555 EVENT AMD_I_NO_UPLOAD  
556 FUNCTION AMD_set_status_to_free  
557 FUNCTION AMD_clr_ready_to_play  
558 FUNCTION AMD_cancel_remove_msg  
559 FUNCTION AMD_send_das_deallocate_noresp /* DAS applic */  
560 FUNCTION AMD_lsdog_gh_idle  
561 TRANS <AMD_idle>  
562  
563 EVENT ANY_STABLE_CALL  
564 FUNCTION AMD_tmrc_any_unstable_ack  
565 TRANS <AMD_chk_if_upload>  
566  
567 EVENT ANY_FORCE_OOS  
568 FUNCTION AMD_set_status_to_free  
569 FUNCTION AMD_cancel_remove_msg  
570 FUNCTION AMD_send_das_deallocate_noresp /* DAS applic */  
571 FUNCTION AMD_lsdog_send_gh_idle  
572 TRANS <AMD_idle>  
573  
574 DONE  
575
```

2103204

92/11/09
12:21:13

/tmp/59781

```
576 *****
577 /*****
578 *
579 *   AMD_chk_msg_status
580 *
581 *   Description:
582 *       This state indicates that the message has been recorded and
583 *       uploaded. It will decide either to deallocate the DAS memory
584 *       or retain it based on the internal event.
585 *
586 *   Resources Active:
587 *       DAS memory.
588 *
589 *   Critical Dependencies:
590 *       At this point, AMD has received either LDH_IDLE or LDH_READY in
591 *       the previous state.
592 *
593 *****
594
595
596
597 STATE   AMD_chk_msg_status
598
599 EVENT   AMD_IDLE
600 FUNCTION
601 FUNCTION
602 TRANS   <AMD_idle>
603
604 EVENT   AMD_READY
605 TRANS   <AMD_wt_playback>
606
607 EVENT   ANY_STABLE_CALL
608 FUNCTION
609 TRANS   <AMD_chk_msg_status>
610
611 EVENT   ANY_FORCE_OOS
612 FUNCTION
613 FUNCTION
614 TRANS   <AMD_idle>
615
616 DONE
617
618 *****
619 /*****
620 *
621 *   AMD_wt_upload_tn
622 *
623 *   Description:
624 *       This state indicates that the telephone number (TN) has been
625 *       recorded and is waiting for an event from the vector to save
626 *       the number on the disk.
627 *
628 *   Resources Active:
629 *       None.
630 *
631 *   Critical Dependencies:
632 *       None.
633 *
634 *****
635
636
637
638 STATE   AMD_wt_upload_tn
639
640 EVENT   LDH_UPLOAD_TN
641 FUNCTION
642 FUNCTION
643 FUNCTION
644 FUNCTION
645 TRANS   <AMD_wt_update_tn_comp>
646
647
648 EVENT   LDH_IDLE_DELETE /* Agent has answer the phone while recording */
649 FUNCTION
650 FUNCTION
651 TRANS   <AMD_idle>
652
653 EVENT   ANY_STABLE_CALL
654 FUNCTION
655 TRANS   <AMD_wt_upload_tn>
656
657 EVENT   ANY_FORCE_OOS
658 FUNCTION
659 FUNCTION
660 FUNCTION
661 FUNCTION
662 TRANS   <AMD_idle>
663
664 DONE
665
```

2103204

92/11/09
12:21:13

/tmp/59781

2103204

```
666 / *****
667 * AMD_wt_update_tn_comp *****
668 * Description: *****
669 * This state is waiting for a response from the message uploader *****
670 * Indicating that the telephone number has been saved on the disk. *****
671 * Resources Active: *****
672 * A timer waiting for the update header complete event. *****
673 * Critical Dependencies: *****
674 * AMG should send either an idle or ready event to indicate *****
675 * whether the message should be removed from the DAS or not. *****
676 * *****
677 * *****
678 * *****
679 * *****
680 * *****
681 * *****
682 * *****
683 * *****
684 * *****
685 * *****
686 * *****
687 * *****
688 * *****
689 * *****
690 * *****
691 * *****
692 * *****
693 * *****
694 * *****
695 * *****
696 * *****
697 * *****
698 * *****
699 * *****
700 * *****
701 * *****
702 * *****
703 * *****
704 * *****
705 * *****
706 * *****
707 * *****
708 * *****
709 * *****
710 * *****
711 * *****
712 * *****
713 * *****
714 * *****
715 * *****
716 * *****
717 * *****
718 * *****
719 * *****
720 * *****
721 * *****
722 * *****
723 * *****
724 * *****
725 * *****
726 * *****
727 * *****
728 * *****
729 * *****
730 * *****
731 * *****
732 * *****
733 * *****
734 * *****
735 * *****
736 * *****
737 * *****
738 * *****
739 * *****
740 * *****
741 * *****
742 * *****
743 * *****
744 * *****
745 * *****
746 * *****
747 * *****
748 * *****
749 * *****
750 * *****
751 * *****
752 * *****
753 * *****
754 * *****
755 * *****
756 * *****
757 * *****
758 * *****
759 * *****
760 * *****
761 * *****
762 * *****
763 * *****
764 * *****
765 * *****
766 * *****
767 * *****
768 * *****
769 * *****
770 * *****
771 * *****
772 * *****
773 * *****
774 * *****
775 * *****
776 * *****
777 * *****
778 * *****
779 * *****
780 * *****
781 * *****
782 * *****
783 * *****
784 * *****
785 * *****
786 * *****
787 * *****
788 * *****
789 * *****
790 * *****
791 * *****
792 * *****
793 * *****
794 * *****
795 * *****
796 * *****
797 * *****
798 * *****
799 * *****
800 * *****
801 * *****
802 * *****
803 * *****
804 * *****
805 * *****
806 * *****
807 * *****
808 * *****
809 * *****
810 * *****
811 * *****
812 * *****
813 * *****
814 * *****
815 * *****
816 * *****
817 * *****
818 * *****
819 * *****
820 * *****
821 * *****
822 * *****
823 * *****
824 * *****
825 * *****
826 * *****
827 * *****
828 * *****
829 * *****
830 * *****
831 * *****
832 * *****
833 * *****
834 * *****
835 * *****
836 * *****
837 * *****
838 * *****
839 * *****
840 * *****
841 * *****
842 * *****
843 * *****
844 * *****
845 * *****
846 * *****
847 * *****
848 * *****
849 * *****
850 * *****
851 * *****
852 * *****
853 * *****
854 * *****
855 * *****
856 * *****
857 * *****
858 * *****
859 * *****
860 * *****
861 * *****
862 * *****
863 * *****
864 * *****
865 * *****
866 * *****
867 * *****
868 * *****
869 * *****
870 * *****
871 * *****
872 * *****
873 * *****
874 * *****
875 * *****
876 * *****
877 * *****
878 * *****
879 * *****
880 * *****
881 * *****
882 * *****
883 * *****
884 * *****
885 * *****
886 * *****
887 * *****
888 * *****
889 * *****
890 * *****
891 * *****
892 * *****
893 * *****
894 * *****
895 * *****
896 * *****
897 * *****
898 * *****
899 * *****
900 * *****
901 * *****
902 * *****
903 * *****
904 * *****
905 * *****
906 * *****
907 * *****
908 * *****
909 * *****
910 * *****
911 * *****
912 * *****
913 * *****
914 * *****
915 * *****
916 * *****
917 * *****
918 * *****
919 * *****
920 * *****
921 * *****
922 * *****
923 * *****
924 * *****
925 * *****
926 * *****
927 * *****
928 * *****
929 * *****
930 * *****
931 * *****
932 * *****
933 * *****
934 * *****
935 * *****
936 * *****
937 * *****
938 * *****
939 * *****
940 * *****
941 * *****
942 * *****
943 * *****
944 * *****
945 * *****
946 * *****
947 * *****
948 * *****
949 * *****
950 * *****
951 * *****
952 * *****
953 * *****
954 * *****
955 * *****
956 * *****
957 * *****
958 * *****
959 * *****
960 * *****
961 * *****
962 * *****
963 * *****
964 * *****
965 * *****
966 * *****
967 * *****
968 * *****
969 * *****
970 * *****
971 * *****
972 * *****
973 * *****
974 * *****
975 * *****
976 * *****
977 * *****
978 * *****
979 * *****
980 * *****
981 * *****
982 * *****
983 * *****
984 * *****
985 * *****
986 * *****
987 * *****
988 * *****
989 * *****
990 * *****
991 * *****
992 * *****
993 * *****
994 * *****
995 * *****
996 * *****
997 * *****
998 * *****
999 * *****
1000 * *****
```

```
734 /*****
735 *
736 *
737 *   AMD_chk_if_idle_pseudo
738 *
739 *   Description:
740 *   In this transition state it has been determined whether or not a
741 *   pseudo port should be put in the idle list based on the internal
742 *   event it receives.
743 *
744 *   Resources Active:
745 *   None.
746 *
747 *   Critical Dependencies:
748 *   At this point AMD has received either LDH_IDLE_FORWARD, LDH_IDLE,
749 *   LDH_READY or LDH_READY_FORWARD in the previous state.
750 *
751 * *****/
752
753 STATE AMD_chk_if_idle_pseudo
754
755 EVENT AMD_IDLE
756 FUNCTION AMD_2sdog_lgh_idle
757 TRANS <AMD_idle>
758
759 EVENT AMD_READY
760 FUNCTION AMD_wt_playback
761 TRANS <AMD_idle>
762
763 EVENT ANY_STABLE_CALL
764 FUNCTION AMD_2tmc_any_unstable_ack
765 TRANS <AMD_chk_if_idle_pseudo>
766
767 EVENT ANY_FORCE_OOS
768 FUNCTION AMD_2sdog_send_lgh_idle
769 TRANS <AMD_idle>
770
771 DONE AMD_chk_if_idle_pseudo.tde
772
```

```
773 /*****
774 *
775 *
776 *   AMD_wt_playback
777 *
778 *   Description:
779 *   This state is ready to playback a message.
780 *
781 *   Resources Active:
782 *   DAS memory.
783 *
784 *   Critical Dependencies:
785 *   None.
786 *
787 * *****/
788
789 STATE AMD_wt_playback
790
791 EVENT LDH_PLAYBACK
792 FUNCTION AMD_set_status_to_offer
793 FUNCTION AMD_clr_event
794 FUNCTION AMD_save_play_req_info
795 FUNCTION AMD_chk_msgc_pbk_tag
796 TRANS <AMD_chk_msgc_playbk_tag>
797
798 EVENT LDH_IDLE
799 FUNCTION AMD_set_status_to_free
800 FUNCTION AMD_send_das_deallocate_noresp /* to DAS appli:
801 FUNCTION AMD_2sdog_lgh_idle
802 TRANS <AMD_idle>
803
804 EVENT LDH_READY
805 FUNCTION AMD_set_status_to_rdy_play
806 TRANS <AMD_wt_playback>
807
808 EVENT LDH_IDLE_DELETE /* When RC wants to remove the queue AMG sends th
809 FUNCTION AMD_set_status_to_free
810 FUNCTION AMD_send_das_remove_file_noresp /* to BMU uploade
811 FUNCTION AMD_send_das_deallocate_noresp /* to DAS appli:
812 FUNCTION AMD_2sdog_lgh_idle
813 TRANS <AMD_idle>
814
815 EVENT ANY_STABLE_CALL
816 FUNCTION AMD_2tmc_any_stable_ack
817 TRANS <AMD_wt_playback>
818
819 EVENT LDH_RESTART
820 FUNCTION AMD_2tmc_restart_ack
821 TRANS <AMD_wt_playback>
822
823 EVENT ANY_FORCE_OOS
824 FUNCTION AMD_2amg_lgh_clear_sdo_port
825 FUNCTION AMD_clr_ready_to_play
826 FUNCTION AMD_clr_msg_info_tbl
827 FUNCTION AMD_2sdog_lgh_idle
828 TRANS <AMD_idle>
829
830 DONE AMD_wt_playback.tde
831
```

92/11/09
13:21:13

/tmp/59781

```
832 /*****
833 *
834 * AMD_chk_msgq_playbk_tag
835 *
836 *
837 * Description:
838 * In this transition state the message queue playback TAG ID
839 * has been checked to see if one exists. If so, the message type
840 * will be checked.
841 *
842 * Resources Active:
843 * None.
844 *
845 * Critical Dependencies:
846 * None.
847 *
848 *****/
19
409 30 STATE AMD_chk_msgq_playbk_tag
851
410 851 EVENT AMD_I_PBK_TAG_ID_OK
411 852 FUNCTION AMD_chk_msg_type
412 853 TRANS <AMD_chk_tn_only>
413 854
414 855
415 856 EVENT AMD_I_NC_PBK_TAG_ID
416 857 FUNCTION AMD_mnt_inform_pbk_tag_id /* to VMS */
417 858 FUNCTION AMD_send_any_activate_nack
418 859 FUNCTION AMD_set_status_to_rdy_play
419 860 TRANS <AMD_wt_playback>
420 861
421 862 ANY_STABLE_CALL
422 863 FUNCTION AMD_2tmc_any_stable_ack
423 864 TRANS <AMD_chk_msgq_playbk_tag>
424 865
425 866 LDH_RESTART
426 867 FUNCTION AMD_2tmc_restart_ack
427 868 FUNCTION AMD_send_any_activate_nack /* to VMS */
428 869 FUNCTION AMD_set_status_to_rdy_play
429 870 TRANS <AMD_wt_playback>
430 871
431 872 ANY_FORCE_OOS
432 873 FUNCTION AMD_send_any_activate_nack /* to VMS */
433 874 FUNCTION AMD_2amg_lgh_clear_sdo_port
434 875 FUNCTION AMD_clr_ready_to_play
435 876 FUNCTION AMD_clr_msg_info_tbl
436 877 FUNCTION AMD_2sdoq_lgh_idle
437 878 TRANS <AMD_idle>
438 879
439 880 DONE
440 881 AMD_chk_msgq_playbk_tag.tde
```

```
882 /*****
883 *
884 * AMD_chk_tn_only
885 *
886 *
887 * Description:
888 * In this transition state an internal event will be received to
889 * indicate that the message contains only the TN or voice.
890 *
891 * Resources Active:
892 * DAS memory if voice exists.
893 *
894 * Critical Dependencies:
895 * None.
896 *
897 *****/
898
899
900 435 901 STATE AMD_chk_tn_only
902 902
903 436 903 EVENT AMD_I_HAVE_VOICE
904 437 904 FUNCTION AMD_2vms_activate_inform /* to VMS */
905 438 905 FUNCTION AMD_2tag_lgh_acquire_das
906 439 906 TRANS <AMD_play_wt_port>
907 907
908 440 908 EVENT AMD_I_HAVE_TN_ONLY
909 441 909 FUNCTION AMD_2vms_activate_inform_tn /* to VMS */
910 442 910 TRANS <AMD_wt_next_tn_req>
911 911
912 443 912 EVENT ANY_STABLE_CALL
913 444 913 FUNCTION AMD_2tmc_any_stable_ack
914 445 914 TRANS <AMD_chk_tn_only>
915 915
916 446 916 LDH_RESTART
917 447 917 FUNCTION AMD_2tmc_restart_ack
918 448 918 FUNCTION AMD_send_any_activate_nack /* to VMS */
919 449 919 FUNCTION AMD_set_status_to_rdy_play
920 450 920 TRANS <AMD_wt_playback>
921 921
922 451 922 EVENT ANY_FORCE_OOS
923 452 923 FUNCTION AMD_send_any_activate_nack /* to VMS */
924 453 924 FUNCTION AMD_2amg_lgh_clear_sdo_port
925 454 925 FUNCTION AMD_clr_ready_to_play
926 455 926 FUNCTION AMD_clr_msg_info_tbl
927 456 927 FUNCTION AMD_2sdoq_lgh_idle
928 457 928 TRANS <AMD_idle>
929 929
930 458 930 DONE
931 459 931 AMD_chk_tn_only.tde
```

2103204

92/11/09
12:21:13

/tmp/59781

12

2103204

```
932 /*****
933 *
934 *
935 *   AMD_wt_next_tn_req
936 *
937 *   Description:
938 *   This state indicates that VMS has requested a TN message be
939 *   displayed. This message may be requested, removed from the disk
940 *   or moved to a new queue.
941 *
942 *   Resources Active:
943 *   None.
944 *
945 *   Critical Dependencies:
946 *   When an event is received from VMS to remove a message, AMD will
947 *   send a DAS_REMOVE_FILE event to BMD uploader and will assume
948 *   that the uploader frees the system message number by setting the
949 *   status of msg_ctl_tbl to FREE in the memory and on the disk.
950 *
951 * *****/
952
953
954 STATE AMD_wt_next_tn_req
955
956 EVENT LDH_IDLE_DELETE
957 FUNCTION AMD_clr_lgh_remove_msg
958 FUNCTION AMD_send_das_remove_file
959 FUNCTION AMD_start_remove_file_cmp_tmr
960 TRANS <AMD_rm_mv_msg_from_disk>
961
962 EVENT LDH_IDLE
963 FUNCTION AMD_clr_status_offer
964 FUNCTION AMD_clrmsg_lgh_idle
965 TRANS <AMD_idle>
966
967 EVENT LDH_READY
968 FUNCTION AMD_set_status_to_rdy_play
969 TRANS <AMD_wt_playback>
970
971 EVENT LDH_IDLE_FORWARD
972 FUNCTION AMD_clr_status_offer
973 FUNCTION AMD_set_i_event
974 FUNCTION AMD_send_das_move_file
975 FUNCTION AMD_save_new_msg_queue_id
976 FUNCTION AMD_start_move_file_cmp_tmr
977 TRANS <AMD_move_file_to_new_queue>
978
979 EVENT LDH_READY_FORWARD
980 FUNCTION AMD_set_status_to_rdy_play
981 FUNCTION AMD_set_i_event
982 FUNCTION AMD_send_das_move_file
983 FUNCTION AMD_save_new_msg_queue_id
984 FUNCTION AMD_start_move_file_cmp_tmr
985 TRANS <AMD_move_file_to_new_queue>
986
987 EVENT LDH_REPLACE_TN
988 FUNCTION AMD_save_tn_digits
989 FUNCTION AMD_send_das_update_hdr_noresp
990 TRANS <AMD_wt_next_tn_req>
991
992 EVENT ANY_STABLE_CALL
993 FUNCTION AMD_clrmsg_lgh_idle
994 TRANS <AMD_wt_next_tn_req>
```

```
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
```

```
LDH_RESTART
FUNCTION
FUNCTION
FUNCTION
<AMD_wt_playback>

ANY_FORCE_OOS
FUNCTION
FUNCTION
FUNCTION
FUNCTION
FUNCTION
<AMD_idle>

AMD_wt_next_tn_req.tde
DONE
507 1011
```

```
AMD_2tmc_restart_ack
AMD_send_any_sdc_failure /* to VMS */
AMD_set_status_to_rdy_play
```

```
AMD_send_any_sdc_failure /* to VMS */
AMD_clrmsg_lgh_clear_sdc_port
AMD_clr_ready_to_play
AMD_clrmsg_info_tbl
AMD_clrmsg_lgh_idle
```

92/11/09
13:21:13

/tmp/59781

544 1075

```
1012 /*****
1013 /*****
1014 *
1015 *   AMD_play_wt_port
1016 *
1017 *   Description:
1018 *   This state is waiting for a DAS port from TAG LDH in order to
1019 *   play the message.
1020 *
1021 *   Resources Active:
1022 *   Pseudo port, DAS memory.
1023 *
1024 *   Critical Dependencies:
1025 *   None.
1026 *
1027 /*****
1028
1029
1030
1031 STATE   AMD_play_wt_port
1032
1033 EVENT   AMT_ACTIVATE_ACK
1034 FUNCTION AMD_save_das_port_id
1035 FUNCTION AMD_send_any_activate_ack /* to VMS */
1036 TRANS   <AMD_play>
1037
1038 EVENT   AMT_ACTIVATE_NACK
1039 FUNCTION AMD_send_any_activate_nack /* to VMS */
1040 TRANS   <AMD_wt_playback>
1041
1042 EVENT   LDH_IDLE
1043 FUNCTION AMD_clr_status_offer
1044 FUNCTION AMD_tag_lgh_cancel_acquire
1045 FUNCTION AMD_send_das_deallocate_noresp /* to DAS applic
1046 FUNCTION AMD_tag_lgh_idle
1047 TRANS   <AMD_idle>
1048
1049 EVENT   LDH_READY
1050 FUNCTION AMD_set_status_to_rdy_play
1051 FUNCTION AMD_tag_lgh_cancel_acquire
1052 TRANS   <AMD_wt_playback>
1053
1054 EVENT   ANY_STABLE_CALL
1055 FUNCTION AMD_tag_any_stable_ack
1056 TRANS   <AMD_play_wt_port>
1057
1058 EVENT   LDH_RESTART
1059 FUNCTION AMD_tag_restart_ack
1060 FUNCTION AMD_send_any_activate_nack /* to VMS */
1061 FUNCTION AMD_tag_lgh_cancel_acquire
1062 FUNCTION AMD_tag_lgh_cancel_acquire
1063 FUNCTION AMD_set_status_to_rdy_play
1064 TRANS   <AMD_wt_playback>
1065
1066 EVENT   ANY_FORCE_OOS
1067 FUNCTION AMD_send_any_activate_nack /* to VMS */
1068 FUNCTION AMD_tag_lgh_cancel_acquire
1069 FUNCTION AMD_tag_lgh_cancel_acquire
1070 FUNCTION AMD_clr_ready_to_play
1071 FUNCTION AMD_clr_msg_info_tbl
1072 FUNCTION AMD_tag_lgh_idle
1073 TRANS   <AMD_idle>
1074
1075 DONE   AMD_play_wt_port.tdo
```

2103204

13

92/11/89
12:21:13

/tmp/59781

```
1076 /*****
1077 /*****
1078 *
1079 *   AMD_play
1080 *
1081 *   Description:
1082 *   This state indicates that the message is playing and is waiting
1083 *   for end of message from TAG LDH or that VMS decided to stop
1084 *   the message from playing.
1085 *
1086 *   Resources Active:
1087 *   DAS port, DAS memory.
1088 *
1089 *   Critical Dependencies:
1090 *   None.
1091 *
1092 *****/
1093
1094 STATE AMD_play
1095
1096 145 1096 STATE AMD_play
1097
1098 146 1098 EVENT ANY_END_OF_MESSAGE
1099 147 1099 FUNCTION AMD_send_any_end_of_message /* to VMS */
1100 148 1100 TRANS <AMD_wt_next_req>
1101
1102 149 1102 EVENT LDH_PLAY
1103 150 1103 FUNCTION AMD_ctag_ldh_play
1104 151 1104 TRANS <AMD_play>
1105
1106 152 1106 EVENT LDH_STOP_PLAY
1107 153 1107 FUNCTION AMD_ctag_ldh_stop_play
1108 154 1108 TRANS <AMD_wt_next_req>
1109
1110 155 1110 EVENT LDH_IDLE
1111 156 1111 FUNCTION AMD_clr_status_offer
1112 157 1112 FUNCTION AMD_ctag_ldh_idle
1113 158 1113 FUNCTION AMD_send_das_deallocate_noresp /* to DAS applic
1114 159 1114 FUNCTION AMD_2sdoq_lgh_idle
1115 160 1115 TRANS <AMD_idle>
1116
1117 161 1117 EVENT LDH_READY
1118 162 1118 FUNCTION AMD_set_status_to_rdy_play
1119 163 1119 FUNCTION AMD_ctag_ldh_idle
1120 164 1120 TRANS <AMD_wt_playback>
1121
1122 /* When agent pressed direct soft key while playing flash ann */
1123 EVENT LDH_IDLE_DELETE
1124 165 1123 FUNCTION AMD_2amg_lgh_remove_msg
1125 166 1124 FUNCTION AMD_ctag_ldh_idle
1126 167 1125 FUNCTION AMD_send_das_deallocate_msg /* to DAS applic */
1127 168 1126 FUNCTION AMD_start_deallocate_msg_tmr
1128 169 1127 TRANS <AMD_rm_deall_msg_ack>
1129
1130 /* When agent pressed direct soft key while playing flash ann */
1131 EVENT LDH_IDLE_FORWARD
1132 170 1128 FUNCTION AMD_clr_status_offer
1133 171 1129 FUNCTION AMD_set_event
1134 172 1130 FUNCTION AMD_ctag_ldh_idle
1135 173 1131 FUNCTION AMD_send_das_deallocate_noresp /* to DAS applic
1136 174 1132 FUNCTION AMD_send_new_move_file
1137 175 1133 FUNCTION AMD_start_move_file_cmp_tmr
1138 176 1134 TRANS <AMD_move_file_to_new_queue>
1139
1140 /* When agent pressed direct soft key while playing flash ann */
1141 EVENT LDH_READY_FORWARD
1142 177 1135 FUNCTION AMD_set_status_to_rdy_play
1143 178 1136 FUNCTION AMD_set_event
1144 179 1137 FUNCTION AMD_ctag_ldh_idle
1145 180 1138 FUNCTION AMD_send_das_move_file
1146 181 1139 FUNCTION AMD_save_new_msg_queue_id
1147 182 1140 FUNCTION AMD_start_move_file_cmp_tmr
1148 183 1141 TRANS <AMD_move_file_to_new_queue>
1149
1150 /* When agent pressed dial # soft key while playing flash ann */
1151 EVENT LDH_REPLACE_TN
1152 184 1142 FUNCTION AMD_save_tn_digits
1153 185 1143 FUNCTION AMD_send_das_update_hdr_noresp
1154 186 1144 TRANS <AMD_wt_next_req>
1155
1156 187 1145 EVENT ANY_MSG_FAILURE
1157 188 1146 FUNCTION AMD_send_any_sdc_failure /* to VMS */
1158 189 1147 TRANS <AMD_wt_playback>
1159
1160 190 1148 EVENT ANY_STABLE_CALL
1161 191 1149 FUNCTION AMD_2tmc_any_stable_ack
1162 192 1150 TRANS <AMD_play>
1163
1164 193 1151 EVENT LDH_RESTART
1165 194 1152 FUNCTION AMD_2tmc_restart_ack
1166 195 1153 FUNCTION AMD_send_any_sdc_failure /* to VMS */
1167 196 1154 FUNCTION AMD_set_status_to_rdy_play
1168 197 1155 FUNCTION AMD_ctag_ldh_idle
1169 198 1156 TRANS <AMD_wt_playback>
1170
1171 199 1157 EVENT ANY_FORCE_OOS
1172 200 1158 FUNCTION AMD_send_any_sdc_failure /* to VMS */
1173 201 1159 FUNCTION AMD_2tag_ldh_idle
1174 202 1160 FUNCTION AMD_2amg_lgh_clear_sdo_port
1175 203 1161 FUNCTION AMD_clr_ready_to_play
1176 204 1162 FUNCTION AMD_clr_msg_inf_tbl
1177 205 1163 FUNCTION AMD_2sdoq_lgh_idle
1178 206 1164 TRANS <AMD_idle>
1179
1180 207 1165 TRANS <AMD_play.tde>
1181 208 1166 DONE
1182 209 1167
```

2103204

14

THE UNIVERSITY OF CHICAGO

```

1183 /*****  

1184 *  

1185 * AMD_wt_next_req  

1186 *  

1187 * Description:  

1188 * This state indicates that the voice message has stopped playing  

1189 * and is waiting for the next action from VMS.  

1190 *  

1191 * Resources Active:  

1192 * DAS port, Pseudo port and DAS memory.  

1193 *  

1194 * Critical Dependencies:  

1195 * When an event is received from VMS to remove a message, AMD will  

1196 * send a DAS_REMOVE_FILE event to BMO uploader and assumes that  

1197 * the uploader frees up the system message number by setting the  

1198 * status of msg_atl_tbl to FREE in the memory and on the disk.  

1199 *  

1200 *****/  

1201  

1202 STATE AMD_wt_next_req  

1203  

1204 EVENT LDH_PLAY AMD_tag_ldh_play  

1205 FUNCTION  

1206 TRANS <AMD_play>  

1207  

1208 EVENT LDH_IDLE_DELETE  

1209 FUNCTION AMD_clr_ldh_remove_msg  

1210 AMD_tag_ldh_idle  

1211 FUNCTION AMD_send_das_deallocate_msg /* to DAS applic */  

1212 FUNCTION AMD_start_deallocate_msg_tmr  

1213 TRANS <AMD_tm_deall_msg_ack>  

1214  

1215 EVENT LDH_IDLE  

1216 FUNCTION AMD_clr_status_offer  

1217 FUNCTION AMD_tag_ldh_idle  

1218 FUNCTION AMD_send_das_deallocate_noresp /* to DAS applic  

1219 FUNCTION AMD_clr_ldh_idle  

1220 TRANS <AMD_idle>  

1221  

1222 EVENT LDH_READY  

1223 FUNCTION AMD_set_status_to_rdy_play  

1224 FUNCTION AMD_tag_ldh_idle  

1225 TRANS <AMD_wt_playback>  

1226  

1227 EVENT LDH_IDLE_FORWARD  

1228 FUNCTION AMD_clr_status_offer  

1229 FUNCTION AMD_set_i_event  

1230 FUNCTION AMD_tag_ldh_idle  

1231 FUNCTION AMD_send_das_deallocate_noresp /* to DAS applic  

1232 FUNCTION AMD_send_new_move_file  

1233 FUNCTION AMD_save_new_msg_queue_id  

1234 FUNCTION AMD_start_move_file_tmr  

1235 TRANS <AMD_move_file_to_new_queue>  

1236  

1237 EVENT LDH_READY_FORWARD  

1238 FUNCTION AMD_set_status_to_rdy_play  

1239 FUNCTION AMD_set_i_event  

1240 FUNCTION AMD_tag_ldh_idle  

1241 FUNCTION AMD_send_das_move_file  

1242 FUNCTION AMD_save_new_msg_queue_id  

1243 FUNCTION AMD_start_move_file_tmr  

1244 TRANS <AMD_move_file_to_new_queue>  

1245

```

92/11/09
12:21:13

/tmp/59781

16

2103204

```
1277 /*****
1278 *
1279 *
1280 * AMD_rmrv_deall_msg_ack
1281 *
1282 * Description:
1283 * In this state the DAS application has been requested to remove
1284 * the message from DAS memory, and AMD is waiting for a response.
1285 *
1286 * Resources Active:
1287 * A timer for deallocate message ack.
1288 *
1289 * Critical Dependencies:
1290 * None.
1291 *
1292 *****/
1293
1294 14
1295 35
1296 STATE AMD_rmrv_deall_msg_ack
1297
1298 EVENT ANY_DEALLOCATE_MSG_ACK
1299 FUNCTION AMD_stop_timer
1300 FUNCTION AMD_send_das_remove_file
1301 FUNCTION AMD_start_remove_file_tmp_tmr
1302 TRANS <AMD_rmrv_msg_from_disk>
1303
1304 EVENT ANY_DEALLOCATE_MSG_NACK
1305 FUNCTION AMD_stop_timer
1306 FUNCTION AMD_send_das_remove_file
1307 FUNCTION AMD_start_remove_file_tmp_tmr
1308 TRANS <AMD_rmrv_msg_from_disk>
1309
1310 EVENT AMD_I_DEALLOCATE_MSG_TMOUT
1311 FUNCTION AMD_mnt_inform_deall_tmout
1312 FUNCTION AMD_send_das_remove_file
1313 FUNCTION AMD_start_remove_file_cmp_tmr
1314 TRANS <AMD_rmrv_msg_from_disk>
1315
1316 EVENT ANY_STABLE_CALL
1317 FUNCTION AMD_2tmc_any_stable_ack
1318 TRANS <AMD_rmrv_deall_msg_ack>
1319
1320 EVENT LDH_RESTART
1321 FUNCTION AMD_2tmc_restart_ack
1322 FUNCTION AMD_send_das_deallocate_msg
1323 FUNCTION AMD_start_deallocate_msg_tmr
1324 TRANS <AMD_rmrv_deall_msg_ack>
1325
1326 EVENT ANY_FORCE_OOS
1327 FUNCTION AMD_stop_timer
1328 FUNCTION AMD_2sdcg_ign_idle
1329 TRANS <AMD_idle>
1330
1331 AMD_rmrv_deall_msg_ack.tde
1332
```

```
1332 /*****
1333 *
1334 *
1335 *
1336 * AMD_rmrv_msg_from_disk
1337 *
1338 * Description:
1339 * This state indicates that the voice message has been removed
1340 * from the DAS memory if it existed, and a request has been made
1341 * to the message uploader to remove the message from the disk.
1342 *
1343 * Resources Active:
1344 * Timer for removing the file.
1345 *
1346 * Critical Dependencies:
1347 * BMD uploader will free the system message before removing the
1348 * file.
1349 *
1350 *****/
1351
1352 STATE AMD_rmrv_msg_from_disk
1353
1354 EVENT ANY_REMOVE_FILE_COMPLETE
1355 FUNCTION AMD_stop_timer
1356 FUNCTION AMD_2sdcg_ign_idle
1357 TRANS <AMD_idle>
1358
1359 EVENT ANY_UPLOAD_FAILURE
1360 FUNCTION AMD_stop_timer
1361 FUNCTION AMD_2sdcg_ign_idle
1362 TRANS <AMD_idle>
1363
1364 EVENT AMD_I_REMOVE_FILE_TMOUT
1365 FUNCTION AMD_mnt_inform_remove_tmout
1366 FUNCTION AMD_2sdcg_ign_idle
1367 TRANS <AMD_idle>
1368
1369 EVENT ANY_STABLE_CALL
1370 FUNCTION AMD_2tmc_any_stable_ack
1371 TRANS <AMD_rmrv_msg_from_disk>
1372
1373 EVENT LDH_RESTART
1374 FUNCTION AMD_2tmc_restart_ack
1375 FUNCTION AMD_send_das_remove_file
1376 FUNCTION AMD_start_remove_file_cmp_tmr
1377 TRANS <AMD_rmrv_msg_from_disk>
1378
1379 EVENT ANY_FORCE_OOS
1380 FUNCTION AMD_stop_timer
1381 FUNCTION AMD_2sdcg_ign_idle
1382 TRANS <AMD_idle>
1383
1384 AMD_rmrv_msg_from_disk.tde
1385
```

92/11/09
12:21:13

/tmp/59781

17

2103204

```
1386 /*****
1387 *
1388 *
1389 * AMD_move_file_to_new_queue
1390 *
1391 * Description:
1392 * This state indicates that a request has been made to the message
1393 * uploader to move the message to another queue.
1394 *
1395 * Resources Active:
1396 * Pseudo port and timer for moved file.
1397 *
1398 * Critical Dependencies:
1399 * BMD uploader will free the system message before removing the
1400 * file.
1401 *
1402 *****/
1403 STATE AMD_move_file_to_new_queue
1404
1405
1406
1407 EVENT ANY_MOVE_FILE_COMPLETE
1408 FUNCTION AMD_stop_timer
1409 FUNCTION AMD_2amg_lgh_upload_tmp
1410 FUNCTION AMD_gen_event
1411 TRANS <AMD_chk_if_idle_pseudo>
1412
1413 EVENT ANY_UPLOAD_FAILURE
1414 FUNCTION AMD_stop_timer
1415 FUNCTION AMD_send_das_remove_file_noresp
1416 FUNCTION AMD_clr_ready_to_play
1417 FUNCTION AMD_2amg_lgh_remove_msg
1418 FUNCTION AMD_2sdog_lgh_idle
1419 TRANS <AMD_idle>
1420
1421 EVENT AMD_MOVE_FILE_TIMEOUT
1422 FUNCTION AMD_mnt_inform_move_timeout
1423 FUNCTION AMD_send_das_remove_file_noresp
1424 FUNCTION AMD_clr_ready_to_play
1425 FUNCTION AMD_2amg_lgh_remove_msg
1426 FUNCTION AMD_2sdog_lgh_idle
1427 TRANS <AMD_idle>
1428
1429
1430 EVENT ANY_STABLE_CALL
1431 FUNCTION AMD_2tmc_any_stable_ack
1432 TRANS <AMD_move_file_to_new_queue>
1433
1434
1435 EVENT LDH_RESTART
1436 FUNCTION AMD_2tmc_restart_ack
1437 FUNCTION AMD_send_das_move_file
1438 FUNCTION AMD_start_move_file_cmp_tmr
1439 TRANS <AMD_move_file_to_new_queue>
1440
1441
1442 EVENT ANY_FORCE_OOS
1443 FUNCTION AMD_stop_timer
1444 FUNCTION AMD_2amg_lgh_clear_sdo_port
1445 FUNCTION AMD_clr_ready_to_play
1446 FUNCTION AMD_clr_msg_info_tbl
1447 FUNCTION AMD_2sdog_lgh_idle
1448 TRANS <AMD_idle>
1449
1450
1451 AMD_move_file_to_new_queue.tde
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3090
3091
3092
3093
3094
3095
3096
3097
3098
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3110
3111
3112
3113
3114
3115
3116
3117
3118
3119
3120
3121
3122
3123
3124
3125
3126
3127
3128
3129
3130
3131
3132
3133
3134
3135
3136
3137
3138
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3150
3151
3152
3153
3154
3155
3156
3157
3158
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3190
3191
3192
3193
3194
3195
3196
3197
3198
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3210
3211
3212
3213
3214
3215
3216
3217
3218
3219
3220
3221
3222
3223
3224
3225
3226
3227
3228
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238
3239
3240
3241
3242
3243
3244
3245
3246
3247
3248
3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269
3270
3271
3272
3273
3274
3275
3276
3277
3278
3279
3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3290
3291
3292
3293
3294
3295
3296
3297
3298
3299
3300
3301
3302
3303
3304
3305
3306
3307
3308
3309
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388
3389
3390
3391
3392
3393
3394
3395
3396
3397
3398
3399
3400
3401
3402
3403
3404
3405
3406
3407
3408
3409
3410
3411
3412
3413
3414
3415
3416
3417
3418
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3479
3480
3481
3482
3483
3484
3485
3486
3487
3488
3489
3490
3491
3492
3493
3494
3495
3496
3497
3498
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578
3579
3580
3581
3582
3583
3584
3585
3586
3587
3588
3589
3590
3591
3592
3593
3594
3595
3596
3597
3598
3599
3600
3601
3602
3603
3604
3605
3606
3607
3608
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619
3620
3621
3622
3623
3624
3625
3626
3627
3628
3629
3630
3631
3632
3633
3634
3635
3636
3637
3638
3639
3640
3641
3642
3643
3644
3645
3646
3647
3648
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658
3659
3660
3661
3662
3663
3664
3665
3666
3667
3668
3669
3670
3671
3672
3673
3674
3675
3676
3677
3678
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3699
3700
3701
3702
3703
3704
3705
3706
3707
3708
3709
3710
3711
3712
3713
3714
3715
3716
3717
3718
3719
3720
3721
3722
3723
3724
3725
3726
3727
3728
3729
3730
3731
3732
3733
3734
3735
3736
3737
3738
3739
3740
3741
3742
3743
3744
3745
3746
3747
3748
3749
3750
3751
3752
3753
3754
3755

```

92/11/09
12:21:13

```

1507 /*****
1508
1509 *
1510 * AMD_chk_if_download
1511 *
1512 * Description:
1513 * This state indicates that the message needs to be downloaded to
1514 * the DAS memory. It will decide either to idle the pseudo port
1515 * or download the message again depending on the internal event
1516 * it receives.
1517 *
1518 * Resources Active:
1519 * None.
1520 *
1521 * Critical Dependencies:
1522 * When the downloading has failed, the downloader deallocates
1523 * the DAS memory. The BMD downloader updates the status of
1524 * msg_ctl_tbl indicating whether it's downloading or not, and
1525 * allocates the DAS memory.
1526 *
1527 *****/
1528
1529 STATE AMD_chk_if_download
1530
1531 EVENT AMD_DOWNLOAD
1532 FUNCTION AMD_chk_msgq_pbx_tag
1533 TRANS AMD_chk_msgq_pbx_tag_id_msg>
1534
1535 EVENT AMD_NC_DOWNLOAD
1536 FUNCTION AMD_chk_msgq_pbx_tag_id_msg>
1537 FUNCTION AMD_chk_msgq_pbx_tag_id_msg>
1538 FUNCTION AMD_chk_msgq_pbx_tag_id_msg>
1539 FUNCTION AMD_chk_msgq_pbx_tag_id_msg>
1540 TRANS <AMD_idle>
1541
1542 EVENT ANY_STABLE_CALL
1543 FUNCTION AMD_chk_if_download>
1544 TRANS <AMD_chk_if_download>
1545
1546 EVENT ANY_FORCE_GOS
1547 FUNCTION AMD_chk_if_download>
1548 FUNCTION AMD_chk_if_download>
1549 FUNCTION AMD_chk_if_download>
1550 FUNCTION AMD_chk_if_download>
1551 TRANS <AMD_idle>
1552
1553 DONE AMD_chk_if_download.tde
1554

```

```

1555 /*****
1556 *
1557 *
1558 * AMD_chk_msgq_pbk_tag_ld_msg
1559 *
1560 * Description:
1561 * In this transition state the message queue playback TAG ID
1562 * has been checked to see if one exists. If so, message
1563 * loading proceeds.
1564 *
1565 * Resources Active:
1566 * None.
1567 *
1568 * Critical Dependencies:
1569 * None.
1570 *
1571 *****/
1572
1573
1574 STATE AMD_chk_msgq_pbk_tag_ld_msg
1575
1576 EVENT AMD_i_pbk_tag_id OK
1577 FUNCTION AMD_send_has_load_msg_hdr
1578 FUNCTION AMD_start_load_msg_hdr_tm
1579 TRANS <AMD_wt_load_comp>
1580
1581 EVENT AMD_i_no_pbk_tag_id
1582 FUNCTION AMD_mnt_inform_pbk_tag_id
1583 FUNCTION AMD_set_status_to_rdy_play
1584 TRANS <AMD_wt_playback>
1585
1586 EVENT ANY_STABLE_CALL
1587 FUNCTION AMD_tmrc_any_unstable_ack
1588 TRANS <AMD_chk_msgq_pbk_tag_ld_msg>
1589
1590 EVENT ANY_FORCE_OOS
1591 FUNCTION AMD_cancel_clear_sdc_port
1592 FUNCTION AMD_clr_ready_to_play
1593 FUNCTION AMD_clr_msg_info_tbl
1594 FUNCTION AMD_sdog_send_ign_idle
1595 TRANS <AMD_idle>
1596
1597 DONE AMD_chk_msgq_pbk_tag_ld_msg.tde
1598
1599 /** SCCS UNPROCESSED COMMENTS BEGIN **
1600 SCCS UNPROCESSED COMMENTS END **/
1601 /** SCCS PROCESSED COMMENTS BEGIN **
1602 02/11/92 chung created
1603 03/26/92 comstock removed_oos state, changed to go to_idle and
1604 send_sdog_ign_idle on ANY_FORCE_OOS, updated
1605 prolog for_idle state.
1606 03/27/92 comstock added HOT init (ANY_STABLE_CALL) handling
1607 03/31/92 comstock "fixed" HOT init handling
1608 04/10/92 comstock "fixed" HOT init handling again for re-inspection
1609 04/30/92 zito Added states to check recording and playback
1610 ID's and fired prologue wording.
1611 05/29/92 paney terminated prologue for state AMD_chk_if_idle_pseudo
1612 06/19/92 jackson Resolve comment/propagation errors from new state m
1613 06/22/92 chung moved the setting of ready to play status to
1614 uploading has completed successfully and removed AMD_
1615 state.
1616 06/24/92 chung call AMD_cancel_ign_upload_cmp function when uploading or

```

2103204

92/11/09
13:21:13

/tmp/59781

19

2103204

```
1618      updating or move file completed successfully.
1619      call AMD_cmg_any_load_fail function when download has
1620      failed.
1621      call AMD_cmg_any_upload_fail function when uploading or
1622      updating or move file failed.
1623      06/26/92      chung      added event LDH_IDLE_DELETE to handle the case
1624      when an agent answer the phone while recording
1625      has not yet completed (this event is sent by vector).
1626      06/30/92      chung      change from calling AMD_cmg_any_upload_fail to
1627      AMD_clr_ready_to_play function since AMG checks ready-to-play
1628      status to determine msg_avail_count.
1629      also added AMD_save_new_msg_queue_id function when receives
1630      LDH_IDLE_FORWARD event.
1631      07/22/92      chung      handle LDH_READY and LDH_IDLE event in AMD_wt_upl
1632      state, and forward/delete/dialt while agent is playing fl
1633      announcement. And clear msg_info_tbi and ready to play b
1634      when AMD gets any force oos while message is on DAS.
1635      08/04/92      chung      removed setting of ready-to-play on ANY_EXCEPT_MSG.
1636      08/04/92      chung      added AMD_clr_download_bit in AMD_wt_load_comp state.
1637      change from AMD_set_status_to_free to AMD_chk_to_set_free upon
1638      receiving ANY_FORCE_OOS in AMD_wt_upload_comp state.
1639      08/10/92      chung      send DAS_DEALLOCATE send with no response upon receiving LDH_IDLE
1640      in AMD_wt_next_req state.
1641      09/01/92      chung      changed LDH_IDLE_DELETE to LDH_IDLE in AMD_wait_r
1642      and change to not wait for das deallocate response when
1643      upon receiving LDH_IDLE_FORWARD event.
1644      09/18/92      chung      call AMD_set_status_to_free upon receiving
1645      ANY_STOP_RECORD_ACK in AMD_record state.
1646      SOCS PROCESSED COMMENTS END **/
```

22 1. INTRODUCTION

23 This document provides Detailed Level Design (DLD) information for the Personal Announcement Pseudo LDH
24 (PAD) state machine also referred to as a Detailed Level Design Unit or DLDU. This document contains four
25 sections:

26 — Introduction - which contains general information about the PAD DLD.

27 — Functional Overview - which describes the operation of the PAD DLDU.

28 — Packaging - which contains the SCI numbers assigned to source and support files.

29 — Unit Testing - which provides the unit test strategy, including a high level description of the tests to be run and
30 where unit testing will be done.

31 The purpose of this DLD is to provide information to allow the implementation of the DLDU defined here and to
32 provide a reference for the state machine once the code is implemented. This document is to be usable by designers
33 and anyone interested in learning the functionality of the defined DLDU. This DLD will provide an overview of the
34 DLDU functionality. If a more detailed view is required, the prologues in the source code and the state definition
35 directories should be referenced.

36 The inspection of this DLD should include the review of the code prologues and state definition files.

38 1.1 Parent Documents

39 This document is required by the Personal Announcements HLD, see reference 4.

41 1.2 DLDU Source File Organization

42 The organization of the source files for this DLDU are:

43 prologues and source code is in:

44 <"FCS ROOT">/cp/call_service/log_dev_hand/pers_ann/common/_src

45 state definition file in:

46 <"FCS ROOT">/cp/call_service/log_dev_hand/pers_ann/common/tde_files

47 global header/structure definition files in:

48 <"SFDS ROOT">/_hdr

49 local header/structure definition files in:

50 <"FCS ROOT">/cp/call_service/log_dev_hand/pers_ann/_hdr

59 2. FUNCTIONAL OVERVIEW (G3KE-20-1456, Parent: G3KE-20-0240)

60 Personal Announcement LDH (PAD) controls the uploading and downloading of personal announcements. It
61 interacts with Pseudo LGH (SDOG), and Personal Announcements Cap (PAC), TAG Logical Device Handler
62 (TAG) and TAG Logical Group Handler (TAGH). PAC will acquire a Personal Announcement LDH in order to
63 download pre-recorded messages or to upload just recorded messages. PAC will send a request to SDOG to acquire
64 an available PA pseudo port. SDOG will choose an available pseudo port and map in PAD. PAC will acquire a
65 PAD for either uploading or downloading as necessary. If PAC makes a download request, PAD will acquire
66 memory for downloading and a tag port for PAC for Personal Announcement (PA) Message playback. PAD will
67 also handle upload requests from PAC, but each PAD will be acquired separately for these purposes. i.e. each
68 instance of PAD will either do uploading or downloading during a single activation, but not both.

25 1. INTRODUCTION

26 This document provides Detailed Level Design (DLD) information for the Personal Announcement Capability
27 (PAC) state machine also referred to as a Detailed Level Design Unit or DLDU. This document contains three
28 sections:

29 — Introduction - which contains general information about the PAC DLD.

30 — Functional Overview - which describes the operation of the PAC DLDU.

31 — Packaging - which contains the SCI numbers assigned to source and support files.

32 The purpose of this DLD is to provide information to allow the implementation of the DLDU defined here and to
33 provide a reference for the state machine once the code is implemented. This document is to be usable by designers
34 and anyone interested in learning the functionality of the defined DLDU. This DLD will provide an overview of the
35 DLDU functionality. If a more detailed view is required, the prologues in the source code and the state definition
36 directories should be referenced.

37 The inspection of this DLD should include the review of the code prologues and state definition files.

39 1.1 Parent Documents

40 This document is required by the Personal Announcements HLD, see reference 4.

42 1.2 DLDU Source File Organization

43 The organization of the source files for this DLDU are:

44 prologues and source code is in:

45 <"FCS ROOT">/cp/call_service/cap/pac/common/_src

46 state definition file in:

47 <"FCS ROOT">/cp/call_service/cap/pac/common/tde_files

48 global header/structure definition files in:

49 <"FCS ROOT">/_hdr

50 local header/structure definition files in:

51 <"FCS ROOT">/cp/call_service/cap/pac/common/_hdr

60 2. FUNCTIONAL OVERVIEW (G3KE-20-1464, Parent: G3KE-20-1447)

61 The Personal Announcements Capability Handler (PAC) controls the playing of the Personal Announcement (PA)
62 messages for the Application and Position calls. It also controls recording, reviewing and deletion of the PA
63 messages by an agent and uploading and downloading of the PA messages.

65 2.1 Feature Activation

66 Upon signin, PAC is activated by the Agent LDH. PAC then checks the current status of the PA messages for that
67 agent and informs PRH of the feature status. It checks for the following conditions:

- 68 • If the primary agent group assignment for the agent does not have any PA names defined, then the condition is
69 treated as if the feature is not provisioned.
- 70 • If one or more PA messages are being uploaded from the previous signin, then PAC waits until all the uploads
71 are complete.
- 72 • If no PA messages have been recorded by the agent for the current primary agent group assignment, then PAC
73 acquires a TAG port and the feature becomes *active*.
- 74 • If one or more PA messages have been recorded, then PAC acquires a Personal Announcement Pseudo LDH
75 (PAD) port from the Pseudo LGH. PAC then tells PAD to download all the recorded messages and acquire a
76 TAG port for the feature.

78 2.2 Playing PA Messages

79 Once the feature is active, PAC will honor the requests from the AAC and PCH capabilities to play PA messages for
80 the Application and Position calls, respectively.

82 2.3 Accessing PA Messages from PA Menus

83 Agent can enter the PA menu from the Voice Messaging main menu. Agent can scroll through the PA messages and
84 perform any of these actions:

- 85 • Review the PA message. PAC will ask the TAG LDH to play the message.
- 86 • Record a PA message. If the PAD is currently performing message downloading, then PAC will notify PAD to
87 abort the downloading process and delete the PA messages from both, DAS and disk. PAC will allocate memory
88 for the recording and ask TAG LDH to start recording. When the recording is complete and the agent stores the
89 message, PAC replaces the existing PA message (if it exists) by the new recording. PAC acquires a PAD and
90 tells it to upload the recorded message. PAC also updates the staff table to store the system message number of
91 the recorded message in the entry associated with the PA name. Once PAD is told to upload the message, PAC
92 keeps no knowledge about the existence of the PAD until agent signs out. At this time PAC checks if one or
93 more PADs are involved with the uploading activity for the agent's recorded PA messages. It notifies each one
94 of them about the signout.
- 95 • Delete an existing PA message. When an agent decides to delete a PA message, PAC checks if any PAD is
96 involved with the uploading of that message. If a PAD is uploading that message, PAC tells it abort the upload
97 and delete the message from disk and DAS. If no PAD is involved with the message upload, then PAC deletes
98 the message from DAS and disk. PAC removes the system message # for the deleted PA message from the Staff
99 table entry, to mark it as *not recorded*.

29 1. INTRODUCTION

30 This document provides the Detailed Level Design information for the message loading DLDU. This document
31 contains three sections:

32 • Introduction - which contains general information about the DLD.

33 • Design - which describes the operation of the DLDU.

34 • Packaging - which contains the SCI numbers assigned to source and support files.

35 The purpose of this DLD is to provide information to allow the implementation of the task defined here and to
36 provide a reference for the DLDU once the code is implemented. This document is used by designers and those
37 interested in learning the functionality of the defined DLDU.

38 The inspection of this DLD may include the review of code prologues and header files.

48 1.2 References

50 2. DESIGN

52 2.1 Function Flow For Message Loader Creation

53 The diagram labeled **Message Loader Creation** shows the processes, functions, tables, mail boxes, and flow of
54 control involved in creating slave loader processes. Although this diagram shows a generic "loader" master and
slave process, it applies to the creation of both uploader and downloader slave processes.

ldr_func_flow
Created 2/27/92 CBD

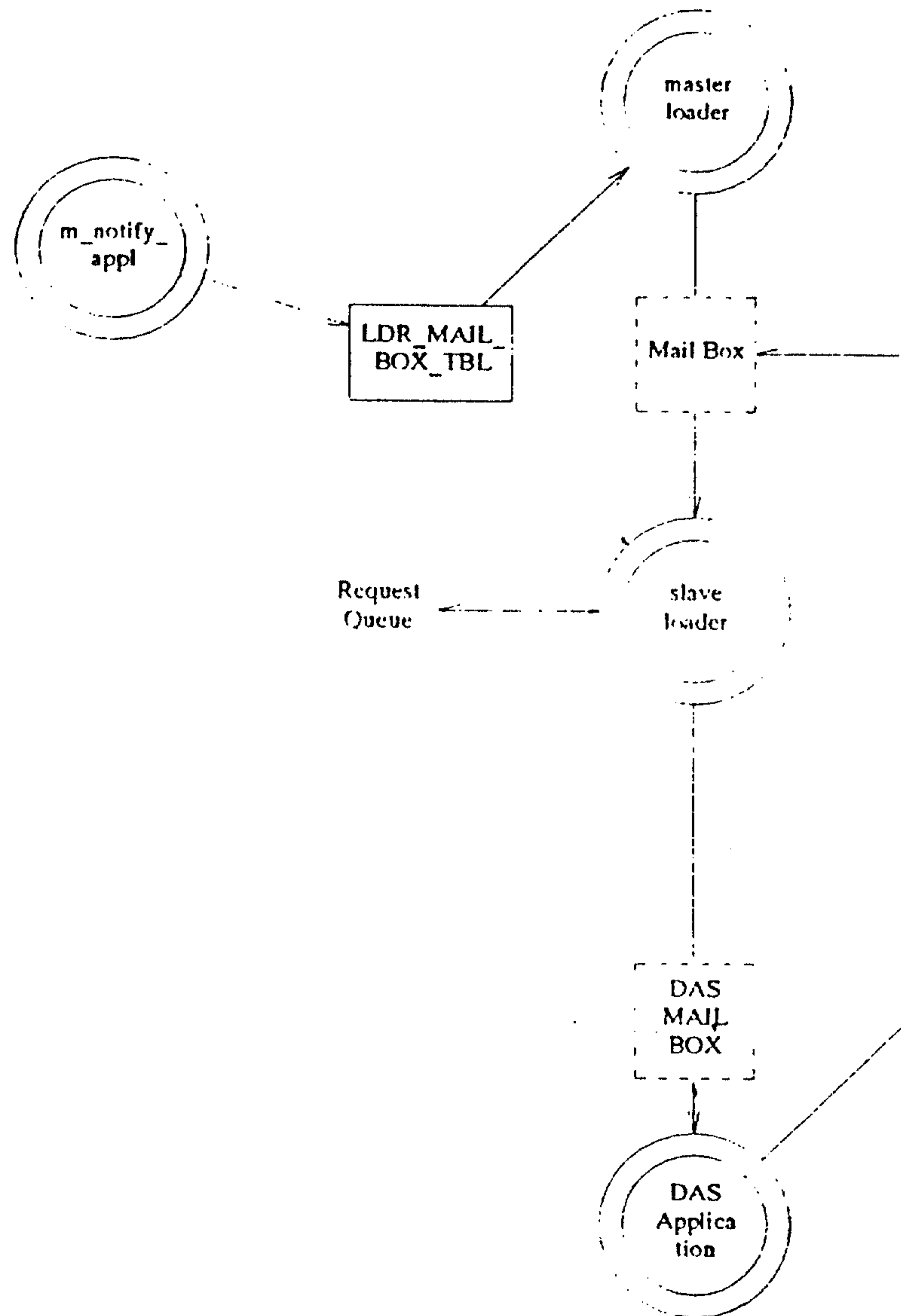


Figure 1. Message Loader Creation

55 The slave processes for a DAS card are started when the card comes into service. The sequence of steps involved in
56 creating a slave process is as follows:

57 • The m_notify_appl process calls a software configuration function, das_notify_appl(), upon receiving an restore
58 indication of the DAS card. The software configuration function will, among other things, create a mail box for
59 the slave process and put the mail box id into the LDR_MAIL_BOX_TBL.

60 • The software configuration function will start the slave process with the following parameters:

61 1. DAS unit number

62 2. mail box id

63 • The slave process will allocate a queuing area and will set itself up to receive events from it's mail box, and then
64 wait for events. The slave process is an essential process, and, as such, will remain active even in the presence of
65 exceptions. Thus, it needs no clean-up routine for fatal exceptions.

66 • The master process is an essential process, and is started by Call Processing subsystem initialization. Any other
67 entity in the system that wants service from the DAS will send a request to the master process.

68 • When the master process receives a request for the DAS that was just put into service, it will search the
69 LDR_MAIL_BOX_TBL for the slave process's mail box id.

70 • If the mail box id is not found (table entry is null), a failure indication is sent back to the requestor.

71 • If the mail box id is found, the request is sent to the slave process through the specified mail box.

72 • To service the request, the slave process communicates with the DAS application through the appropriate mail
73 box (DAS_LD_MBOX or DAS_ULD_MBOX) on the DAS.

74 • The DAS application sends responses back to the slave process through the same mail box that the master
75 process uses to send requests to the slave process.

76 • If the slave process receives a request that it can't handle immediately (because an upload/download is in
77 progress), it will put the request in a queue. After the current load is complete, the slave process will take the
78 request out of the queue and service it.

79

81 2.2 Message Uploader (G3KE-XX-0001, Parent: G3KE-XX-XXXX)

82 The message uploader moves voice messages from DAS memory to the hard disk. This process provides services
83 to capabilities such as Personal Announcements, ACD Mail, and others.

84 The goals of this Detailed Level Design Unit are:

85 1. Achieve a 2:1 ratio of upload time to message length. In other words, a voice message that takes 1 minute to
86 record would take 2 minutes to move from DAS memory to the disk.

87 2. Support uploading multiple DAS cards. Two or more cards may be uploaded in parallel, but only one message
88 at a time can be uploaded from a single card. It is intended that a single DAS card allow both uploading and
89 downloading simultaneously.

90 3. Support new events for Release 2.

92 2.2.1 Uploader On BMU

93 The BMU portion of the uploader runs as a collection of essential processes. The main process, also called the
94 master process, receives requests from any other entity on the BMU that wants a message moved from DAS
95 memory to hard disk. The master process receives requests through the mailbox CP_DAS_ULDR_MBOX.

96 The BMU has a subordinate process (also called a slave process) for every DAS that exists on the system.
97 Whenever the master process receives a request to upload a message, it will pass the request to the slave process for
98 that DAS. That slave process will handle all the details of the upload.

100 2.2.1.1 Multiple DAS Uploader

101 The purpose of the master and multiple slave arrangement on the BMU is to prevent the bottleneck that would result
102 from having a single BMU process handle the upload from all DAS cards. Having a separate process to handle each
103 DAS allows more than one DAS card to upload a message at the same time.

104 Each slave process will have it's own mailbox to which the master process sends requests. The slave process will
 105 handle the request, and send back to the original requestor an event indicating the result of the request. Responses
 106 from the DAS to the slave process will also go through this mailbox.

107 It is possible that a slave process will receive requests from the master process while in the middle of an upload. The
 108 slave process will queue such requests. After an upload is complete, the slave process will de-queue and process any
 109 requests.

111 2.2.1.2 New Events

112 The message uploader must handle new events used by new Release 2 features, or changes to existing events. Two
 113 of these new Release 2 features are ACD Mail and Personal Announcements.

115 2.2.1.2.1 DAS_UPLOAD_HDR

116 This event directs the uploader to write a message to a file, and provides data to be put into the file header. The
 117 difference between this event and DAS_UPLOAD is that this event provides more data for the header.

119 2.2.1.2.2 DAS_UPDATE_HDR

120 The uploader's reaction to this event depends on whether the file for this message exists. If that file exists, the
 121 uploader will replace the contents of the file header with data from the event. If the file does not exist, the uploader
 122 will create it with only a file header; no PCM data will be given. The uploader will return
 123 ANY_UPDATE_COMPLETE_HDR to the original requestor.

125 2.2.1.2.3 DAS_REMOVE_FILE

126 This event directs the uploader to remove a file from the disk. After the file is removed, the uploader will send back
 127 ANY_REMOVE_FILE_COMPLETE event to the requestor.

129 2.2.1.2.4 DAS_MOVE_FILE

130 This event directs the uploader to move a file from one directory to another. This event includes the old (current)
 131 queue id, the new queue id, and the system message number. The queue id identifies one part of the path for the file
 132 to be moved. The system message number is the simple file name. After the file has been moved, the uploader
 133 returns ANY_MOVE_FILE_COMPLETE.

135 2.2.1.2.5 DAS_DEALLOCATE_MSG

136 This event is sent from Personal Announcements to deallocate a message that was just uploaded. The master
 137 uploader will pass this event to the slave process for the appropriate DAS; the slave process will queue the message
 138 until the upload is done, and then send it to the DAS application. Routines to handle this event already exist in the
 139 DAS application.

141 2.2.1.3 Uploader Performance

142 The performance goal for the uploader is to achieve a 2:1 ratio of uploading time to message length. In other words,
 143 a message that takes 1 minute to record would take 2 minutes to upload. To achieve this goal, the following steps
 144 will be taken:

- 145 1. Data will be transferred in 2048-byte blocks.
- 146 2. The file header will be 2048 bytes long, which will force the data to start on a disk sector boundary. This
 147 makes a disk access quicker.
- 148 3. Three blocks of data will be kept 'in flight' all the time. This will reduce the time spent waiting to receive a
 149 block of data.
- 150 4. The priority of data packets will be lowered. While this will not directly affect uploader performance, it will
 151 contribute to overall system performance by allowing other events to move between the DAS and the BMU
 152 with minimum delay.

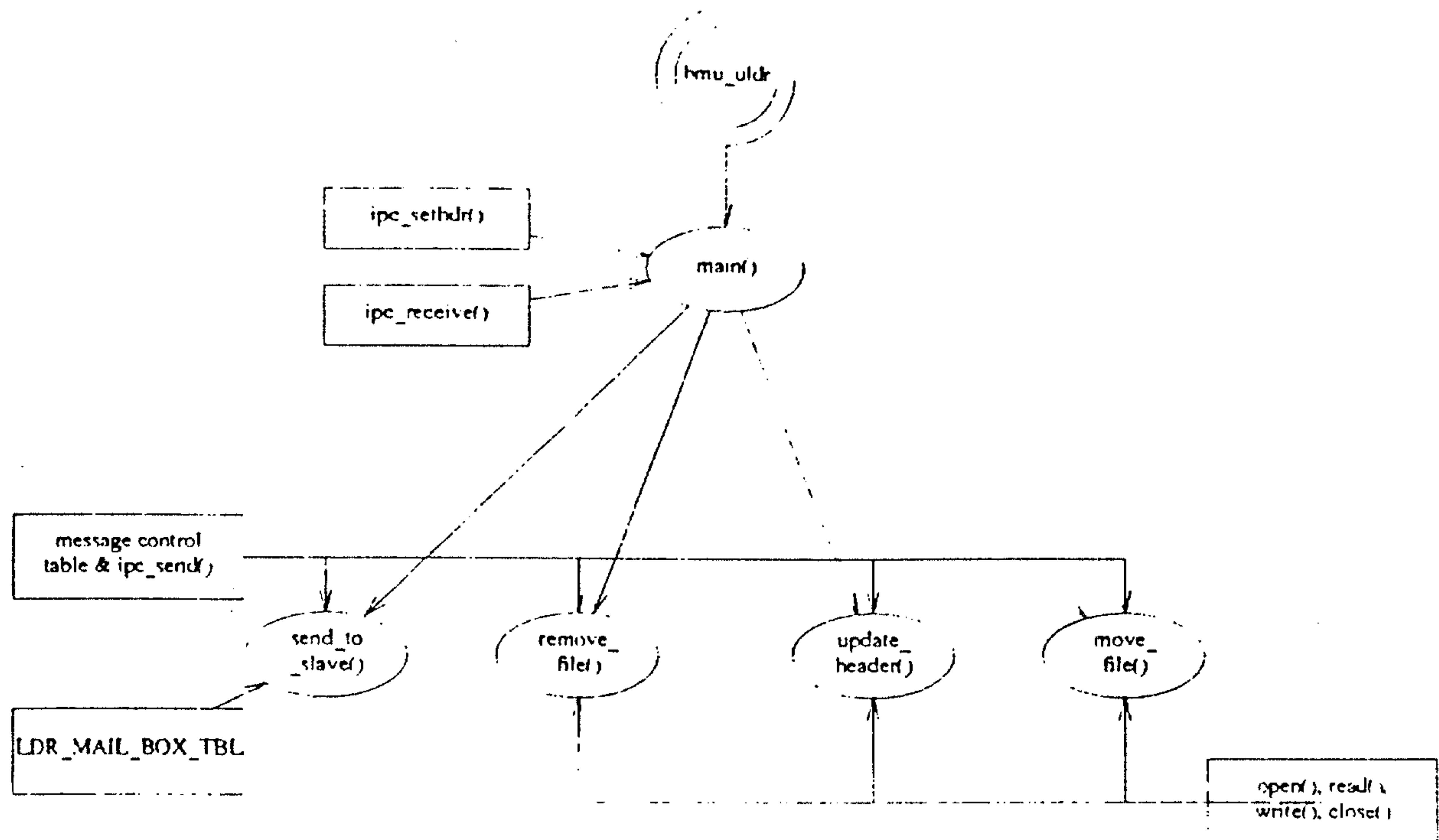


Figure 2. Message Uploader Master Process

154 2.2.2 Message Uploader Master Process

155 The bubble diagram labeled **Message Uploader Master Process** shows the main components of the master
 156 uploader process and the system services the uploader uses.

157 All of the functions called from `main()` access the message control table for information about the message. They
 158 will also use the `ipc_send()` primitive to send events.

159 All of the functions except `send_to_slave()` will use the standard file access routines such as `open()`, `read()`, `write()`,
 160 and `close()`.

162 2.2.2.1 Master Uploader Main()

163 The master uploader process is responsible for receiving and processing requests from any other entity in the
 164 system. The main routine sets up the mechanism for receiving events. It then decides which event it has received,
 165 and handles it accordingly.

166 In doing this work, it uses '`ipc_sethdr()`' to prepare to receive events, and then '`ipc_receive()`' to actually receive the
 167 events. Events are received from other entities in the system that require service.

169 2.2.2.2 `send_to_slave()`

170 When the master uploader process receives a request to upload a message, it will pass that request off to a slave
 171 process. In this function, the master uploader will decide which DAS the request will affect, and send the request to
 172 the slave process for that DAS.

173 This function uses the `LDR_MAIL_BOX_TBL` to find the mail box id for the slave process of interest.

175 2.2.2.3 remove_file()

176 This function will be entered when the master uploader process receives a DAS_REMOVE_FILE event. This
177 function will form the full path name of the file, and call the 'unlink()' function to remove the file. This function will
178 then send ANY_REMOVE_FILE_COMPLETE to the original requestor. If an error is encountered, this function
179 will send ANY_UPLOAD_FAILURE to the requestor.

181 2.2.2.4 update_header()

182 This function will be entered when the master uploader process receives a DAS_UPDATE_HDR event. There are
183 two possibilities to consider then:

- 184 • The file already exists, in which case this function will open that file, read the header, enter the new data in the
185 header, rewind the file to the beginning, write the new header, and close the file.
- 186 • The file does not exist, in which case this function will create the file, write only a file header to the file, and
187 close the file. This will happen in the case when a TN-only ACD Mail message is being written to disk.

188 If no error is encountered, this function will send ANY_UPDATE_COMPLETE_HDR to the original requestor. If
189 an error is encountered, this function will send ANY_UPLOAD_FAILURE to the requestor.

191 2.2.2.5 move_file()

192 This function will be entered when the master uploader process receives a DAS_MOVE_FILE event. This function
193 will form the full path name for the old (current) file, the path name for the new file, open the old file, create the new
194 file, read from the old file, write to the new file, close both files, and unlink the old file. This function will then send
195 ANY_MOVE_FILE_COMPLETE to the original requestor. If an error is encountered, this function will send
ANY_UPLOAD_FAILURE to the original requestor.

hmu_slave
Created 2/18/92 CBJD

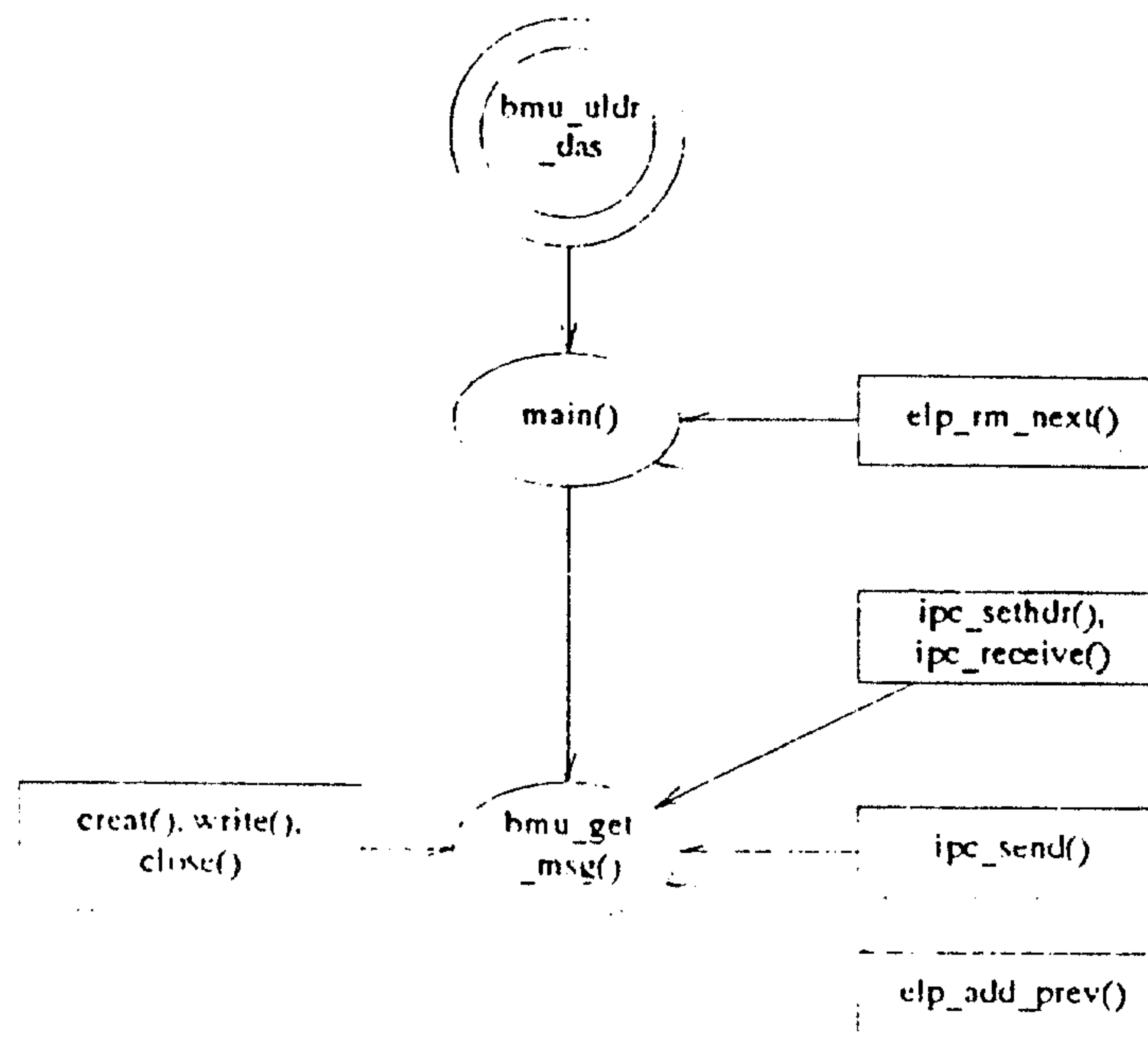


Figure 3. Message Uploader Slave Process

197 2.2.3 Message Uploader Slave Process

198 The bubble diagram labeled **Message Uploader Slave Process** shows the main components of the slave uploader
199 process and the system services the uploader uses.

200 For each DAS provisioned in the system, there will exist on the BMU one such dedicated essential process to handle
201 the details of uploading a message from that DAS. This process is 'called a 'slave' process because it accepts
202 requests from the master process (discussed above).

203 The message uploader slave process will set the status of the message in the message control table to **uploading**
204 before an upload starts, and to **on disk** after uploading is complete. If there is an error encountered in uploading, the
205 status will be set to **recording**. The slave process will also set the appropriate bit in the unit bitmap to denote that the
206 message is on this DAS.

208 2.2.3.1 Slave Uploader Main()

209 The slave uploader process expects to receive events from the master uploader process. The slave process expects
210 either **DAS_UPLOAD** or **DAS_UPLOAD_HDR** events, which tell it to upload a message from it's DAS. The
211 difference between the two events is that **DAS_UPLOAD_HDR** contains more data to be put into the file header.
212 The receipt of any other event at this point is considered an error.

213 It is possible for the slave process to receive an upload request from the master while an upload is in progress. When
214 this happens, the request will be queued. When the upload is finished and the function controlling the upload returns
215 to main (the calling function), main will check the queue for requests. All queued requests will be handled before
216 any new requests are taken.

217 The slave process will concern itself only with the first entry in the queue. There is never a need for the slave
218 process to scan the queue looking for an element that meets certain criteria. Also, there is no need for the slave

hmu_slave
Created 2/18/92 CBD

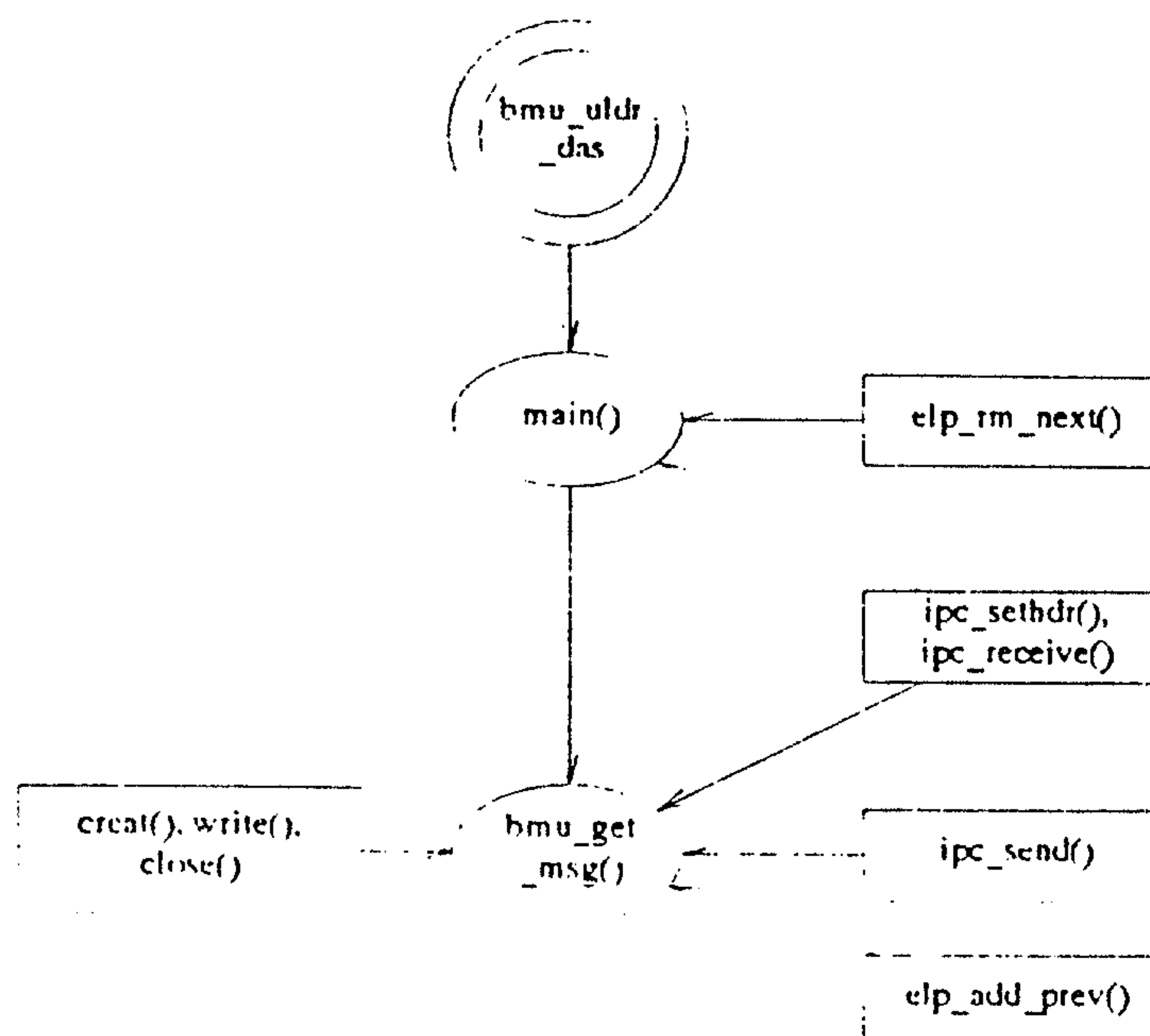


Figure 3. Message Uploader Slave Process

197 2.2.3 Message Uploader Slave Process

198 The bubble diagram labeled **Message Uploader Slave Process** shows the main components of the slave uploader
199 process and the system services the uploader uses.

200 For each DAS provisioned in the system, there will exist on the BMU one such dedicated essential process to handle
201 the details of uploading a message from that DAS. This process is 'called a 'slave' process because it accepts
202 requests from the master process (discussed above).

203 The message uploader slave process will set the status of the message in the message control table to **uploading**
204 before an upload starts, and to **on disk** after uploading is complete. If there is an error encountered in uploading, the
205 status will be set to **recording**. The slave process will also set the appropriate bit in the unit bitmap to denote that the
206 message is on this DAS.

208 2.2.3.1 Slave Uploader Main()

209 The slave uploader process expects to receive events from the master uploader process. The slave process expects
210 either **DAS_UPLOAD** or **DAS_UPLOAD_HDR** events, which tell it to upload a message from it's DAS. The
211 difference between the two events is that **DAS_UPLOAD_HDR** contains more data to be put into the file header.
212 The receipt of any other event at this point is considered an error.

213 It is possible for the slave process to receive an upload request from the master while an upload is in progress. When
214 this happens, the request will be queued. When the upload is finished and the function controlling the upload returns
215 to main (the calling function), main will check the queue for requests. All queued requests will be handled before
216 any new requests are taken.

217 The slave process will concern itself only with the first entry in the queue. There is never a need for the slave
218 process to scan the queue looking for an element that meets certain criteria. Also, there is no need for the slave

Access to abort an upload before it's complete.

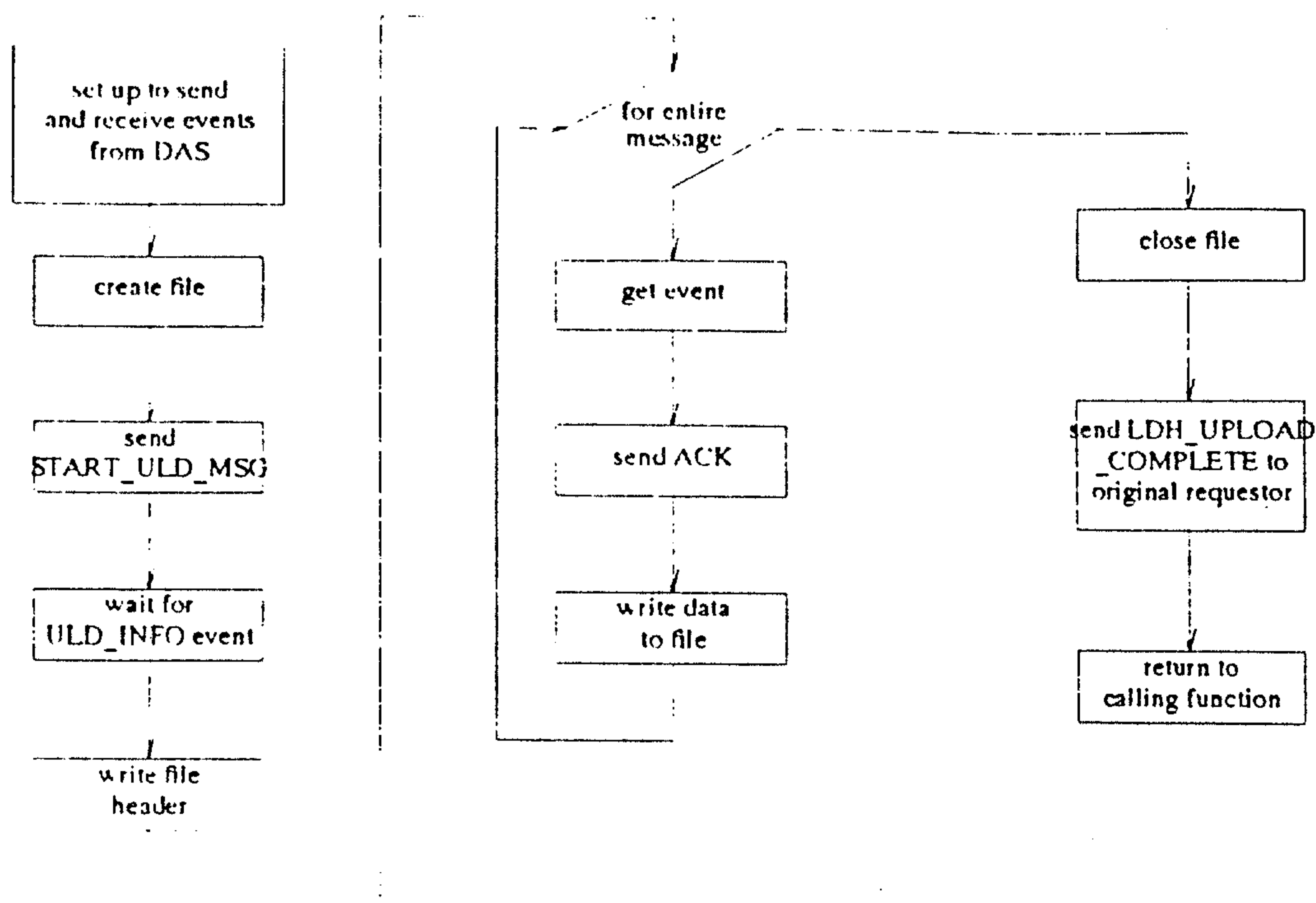


Figure 4. Message Uploader Algorithm

220 2.2.3.2 bmu_get_msg()

221 This function is called by the slave uploader main routine to handle the details of moving a message from DAS
222 memory to disk. The figure labeled **Message Uploader Algorithm** shows the steps that will happen to upload a
223 message.

224 The steps are as follows:

- 225 1. Use ipc calls to get set up to receive events from, and send events to, the DAS application.
- 226 2. Create the file in which the message will be stored, and open it for writing.
- 227 3. Send START_ULD_MSG to tell the DAS application to start uploading the message.
- 228 4. Wait (at ipc_receive) for a ULD_INFO event from the DAS application. This event gives some information
229 about the message that goes in the file header.
- 230 5. When the ULD_INFO event is received, the file header is filled in and written to disk.
- 231 6. At this point, the data has to be received and put into the file. So, until all bytes have been received, do the
232 following:
 - 233 • receive an event
 - 234 • send an acknowledgement for the event
 - 235 • write the data from that event to the file
- 236 7. When the uploader receives a ULD_END event, it closes the file.
- 237 8. The uploader sends LDH_UPLOAD_COMPLETE to the original requestor.

9. This function returns to the calling function.

2103204

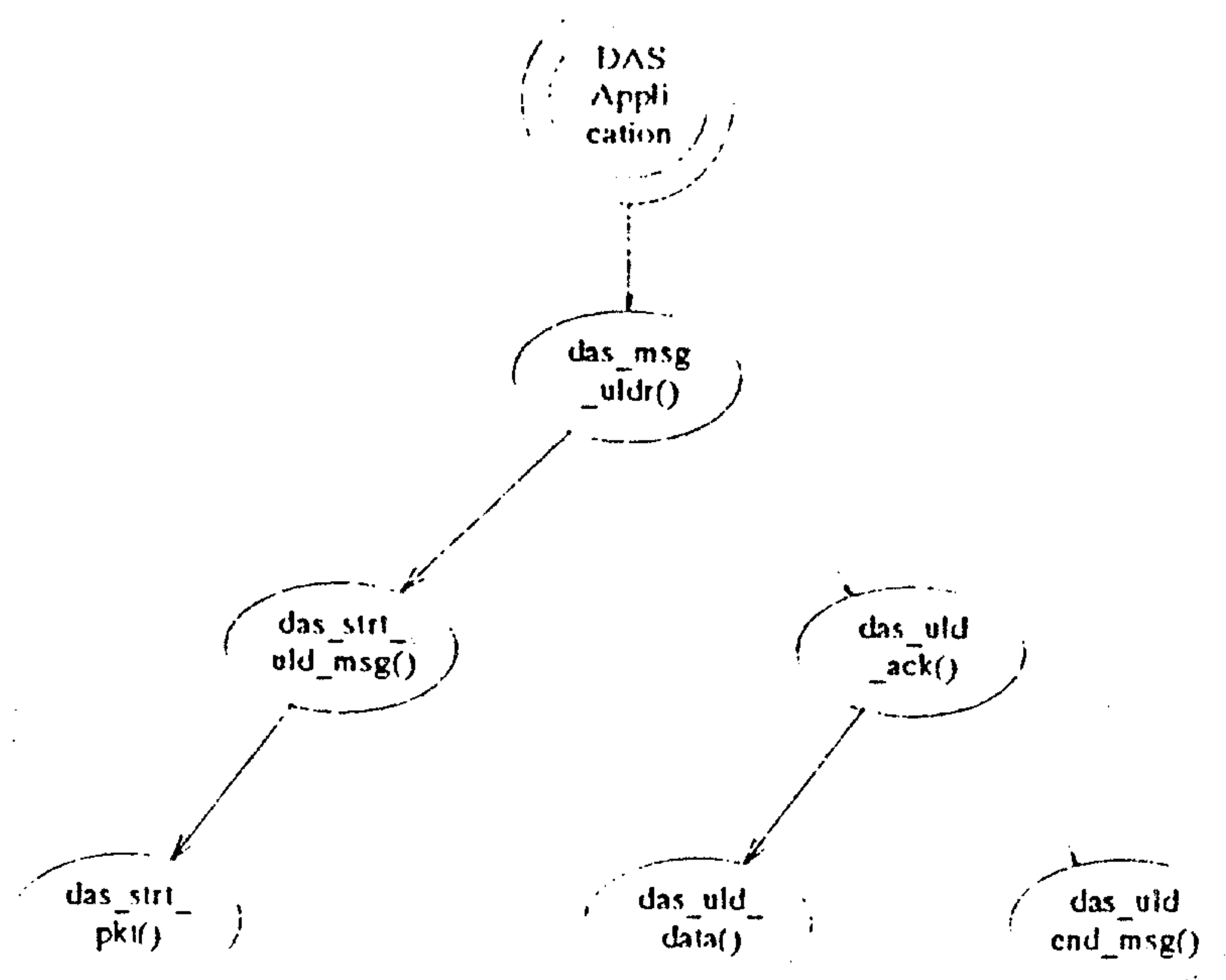


Figure 5. DAS Side Of Uploading Message

240 2.2.4 Uploader Within DAS Application

241 The DAS portion of the uploader runs as a task under the low-level scheduler. The main routine for the DAS
242 portion of the uploader is entered when an event is placed in the uploader's mailbox, DAS_ULD_MBOX. (The
243 correspondence between mailbox and entry point is defined when the system is compiled).

244 The function `das_msg_uldr()` receives events concerning upload from the BMU, and processes them by type of
245 event.

247 2.2.4.1 `das_msg_uldr()`

248 This routine is entered whenever an event is placed in the uploader's mailbox. The connection between this routine
249 and the mailbox is established at compile time. This routine will decide what to do with the received event based on
250 the event type.

252 2.2.4.2 `das_strt_uld_msg()`

253 This routine is called by `das_msg_uldr` when a START_ULD_MSG event is received. It initializes the upload save
254 table, verifies that the requested message number is valid, and then calls `ipc_sethdr` to establish the header for all
255 upload messages sent from the DAS to the BMU. It sends an ULD_INFO event to the BMU, and calls the function
256 `das_strt_pkt` to send the first series of data packets.

258 2.2.4.3 `das_strt_pkt()`

259 This routine is called by `das_strt_uld_msg` to send a small number of data packets to the BMU. This small number
260 (the *window size* referred to earlier) of packets is sent before an acknowledgement is expected.

262 2.2.4.4 `das_uld_ack()`

263 This routine is called by `das_msg_uldr` when an ACK_MSG event is received. It verifies the message number and
264 sequence number. It checks whether all data has been sent for the message being uploaded. If not, function
265 `das_uld_data` is called to send the next sequence. If all data was sent, the DAS uploader will wait for

5 acknowledgements for those packets still in flight. When the acknowledgement for the final piece of the message
267 has been received, the function `das_uld_end_msg` is called to send an `ULD_END` event to the BMU.

269 2.2.4.5 `das_uld_data()`

270 This routine is called by function `das_uld_ack` to format and send the `ULD_DATA` event.

271 It is in this function that part of the performance improvements for the uploader will be implemented. Blocks of
272 PCM data will be moved directly from DPRAM into inter-process communication buffers. This will be done in the
273 following steps:

- 274 • Reference the phrase table to find the starting address of the message
- 275 • Add an offset to get to the next block to send
- 276 • Check whether the next block is still within the message
- 277 • Unlock DPRAM
- 278 • Call `ipc_send` with the DPRAM address of the next block to send
- 279 • Lock DPRAM

281 2.2.4.6 `das_uld_end_msg()`

282 This routine is called by `das_uld_ack` after the final acknowledgement was received. This function formats and
sends the `ULD_END` to the BMU, and sets the state of the upload to IDLE in the upload save table.

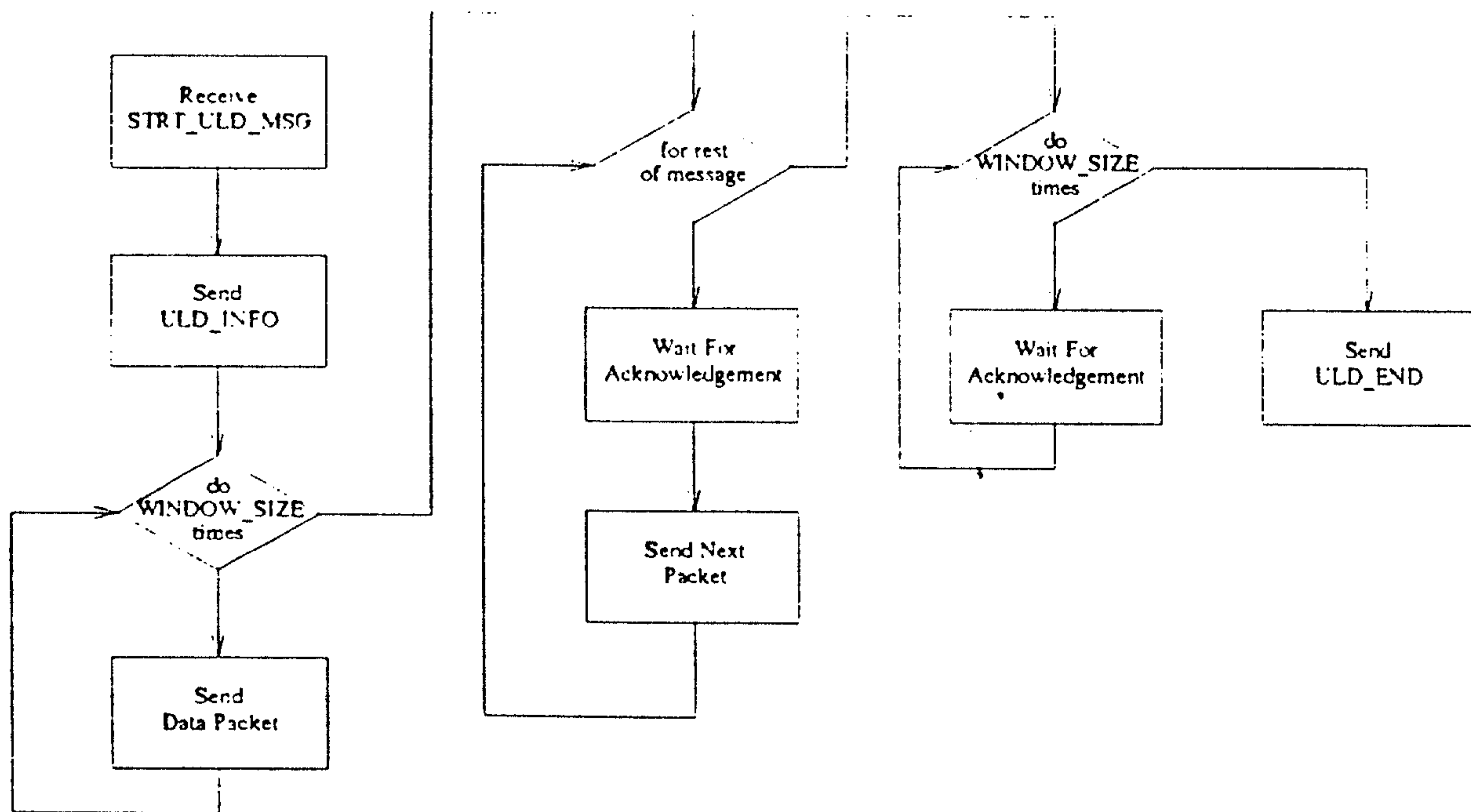


Figure 6. DAS Message Upload Algorithm

284 2.2.5 DAS Uploader Performance

285 The performance goals can be met only if cooperating changes are made to both the BMU and the DAS portions of
286 the uploader. The changes to be made on the DAS to achieve the performance goals are:

- 287 1. Change the data packet size to 2048 bytes.

- 288 2. Implement a 'windowing' approach that will keep a small number of data packets 'in flight' at all times. The
290 algorithm for this windowing approach is shown in the diagram labeled DAS Message Upload Algorithm.
291 An explanation of that algorithm follows:
- 292 • The DAS uploader receives a STRT_ULD_MSG event.
 - 293 • The DAS uploader will send a ULD_INFO event back to the BMU uploader.
 - 294 • The DAS uploader will send the first few data packets to the BMU uploader. This small number is called
295 the *window size*. Only after these packets have been sent will the DAS uploader expect an
296 acknowledgement.
 - 297 • Wait for an acknowledgement of the first packet.
 - 298 • For the rest of the message, the DAS uploader will send a packet of data to the BMU uploader and then
299 wait for an acknowledgement. If packet N has just been sent, the acknowledgement is for packet N -
300 *window size* + 1.
 - 301 • When the DAS uploader reaches the end of the message, it will wait for acknowledgements for those
302 packets still in flight.
 - 303 • When the DAS uploader has received all expected acknowledgements, it will send a ULD_END event to
the BMU uploader.

306 2.3 DAS Message Downloader on BMU (G3KE-XX-0002, Parent: G3KE-XX-XXXX)

307 The BMU message downloader is an essential process that moves voice messages to DAS memory from the hard
308 disk. This process provides service to capabilities such as Personal Announcements, ACD Mail, and others.

309 The goals of this Detailed Level Design Unit are:

- 310 1. Achieve a 2:1 ratio of download time to message length. This means that a voice message that takes 1 minute
311 to play would take 2 minutes to move from disk to DAS memory.
- 312 2. Support downloading to multiple DAS cards. Multiple DAS cards may be downloaded in parallel, but only
313 one message at a time can be downloaded to a single DAS card. It is intended that a single DAS card allow
314 both uploading and downloading simultaneously.
- 315 3. Support new events for Release 2.

317 2.3.1 New events

319 2.3.1.1 DAS_LOAD_MESSAGE_HDR

320 This event is sent to the downloader to save the file header information while downloading the specified
321 announcement (if necessary) and return the header information to the requestor with
322 ANY_LOAD_COMPLETE_HDR event upon completion of downloading. Downloading is not necessary if the
323 mail message type is a TN (Telephone Number) only.

325 2.3.1.2 RMV_ALL_MSG

326 This event is sent by Physical LGH to update the message control to indicate that the messages residing on the DAS
327 card are not available. This event is sent when the DAS card is being removed by Recent Change or by recovery.

329 2.3.2 Message Downloader Master Process

330 The following bubble diagram shows the main components of the master downloader process. Each component of
master downloader process will be described in the next sections.

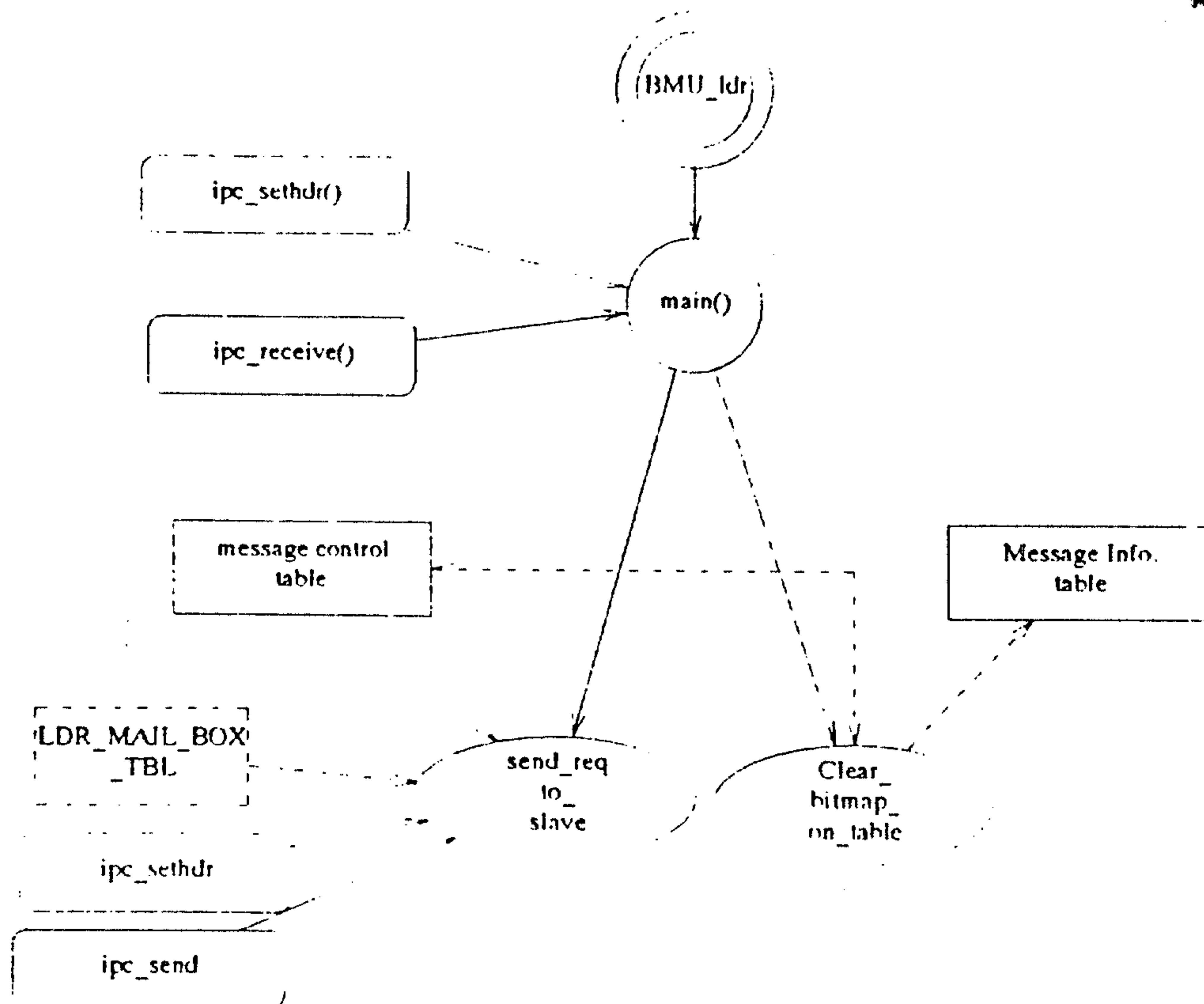


Figure 7. BMU Downloader Master Process

332 2.3.2.1 Main()

333 The master downloader process will be started during Cold initialization by the loader. It's mailbox,
 334 CP_DAS_LDR_MBOX, will be created during subinit. It will make a call to ipc_sethdr to set the header
 335 information so it can receive mail in its mailbox and make a call to ipc_receive to wait for request to arrive in the
 336 mailbox. Following events will be handled by this process.

- 337 1. LOAD_ALL_MSG - Request to download all of the messages that are on disk is received during cold
 338 initialization of the card.
- 339 2. DAS_LOAD_MESSAGE - Request to download a single message.
- 340 3. DAS_LOAD_MESSAGE_HDR - Request to download a single message and save the downloader
 341 information so it can be sent to the requestor.
- 342 4. RMV_ALL_MSG - Request is received to change the bitmap on the message control table to indicate that the
 343 messages on the card are not available. This event is received when the DAS card is removed from service by
 344 Recent Change or by recovery.

346 2.3.2.2 send_req_to_slave()

347 If the request is to load all message, LOAD_ALL_MSG, main() will call this function to forward the message to the
 348 appropriate slave process mailbox that is dedicated to that DAS card. It will call ipc_sethdr and ipc_send to forward
 349 the request to the dedicated downloader (slave) process.

350 If the request is DAS_LOAD_MESSAGE or DAS_LOAD_MESSAGE_HDR, this function will check the message
 351 control table to see if the message to download is a system announcement and needs to be downloaded to one other

2 DAS card. It will then send the request to the appropriate dedicated downloader (slave) processes mailbox(s).

353 This function will calculate the mailbox for the slave process mailbox which is dedicated to the DAS card with the
354 given unit number. It will then call `ipc_sethdr` to set the header information and call `ipc_send` to send the
355 information to the dedicated downloader (slave) process mailbox.

357 2.3.2.3 `Clear_bitmap_on_table()`

358 When the DAS card is removed by Recent Change or by recovery, the message control table needs to show that the
359 messages residing on the card are not available. In this case, `main()` function will receive `RMV_ALL_MSG` request
360 and it will call this function to handle the request. This function will go through the message control table and
361 clears the appropriate bit on the DAS unit bitmap to indicate that the message is not available on this card. It will
362 also clear das message number and unit number on the message information table for the same reason.

364 2.3.3 Message Downloader Slave Process

365 For each DAS card in the system, there will be an dedicated essential process that will handle downloading requests
366 for that DAS card. These process will be slave to Master Process described in previous section and will handle any
367 requests sent by the that process.

368 Following bubble diagrams shows all the components of the downloader slave process and each of these
369 components will be described in the next sections. Two bubble diagrams are created for clarity with box A
370 representing a call to function `bmu_id_all()`.

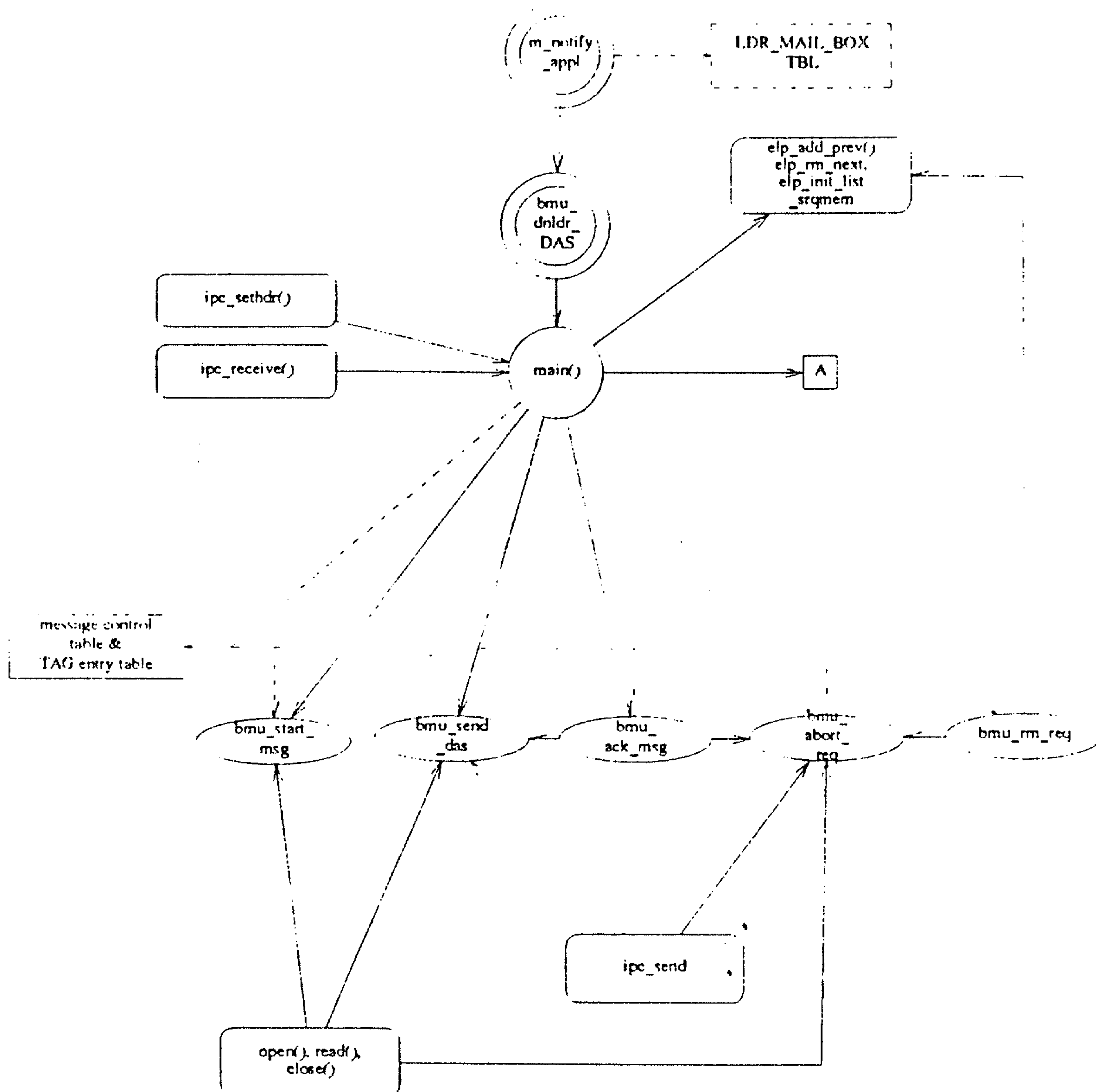


Figure 8. DAS Downloader Slave Process

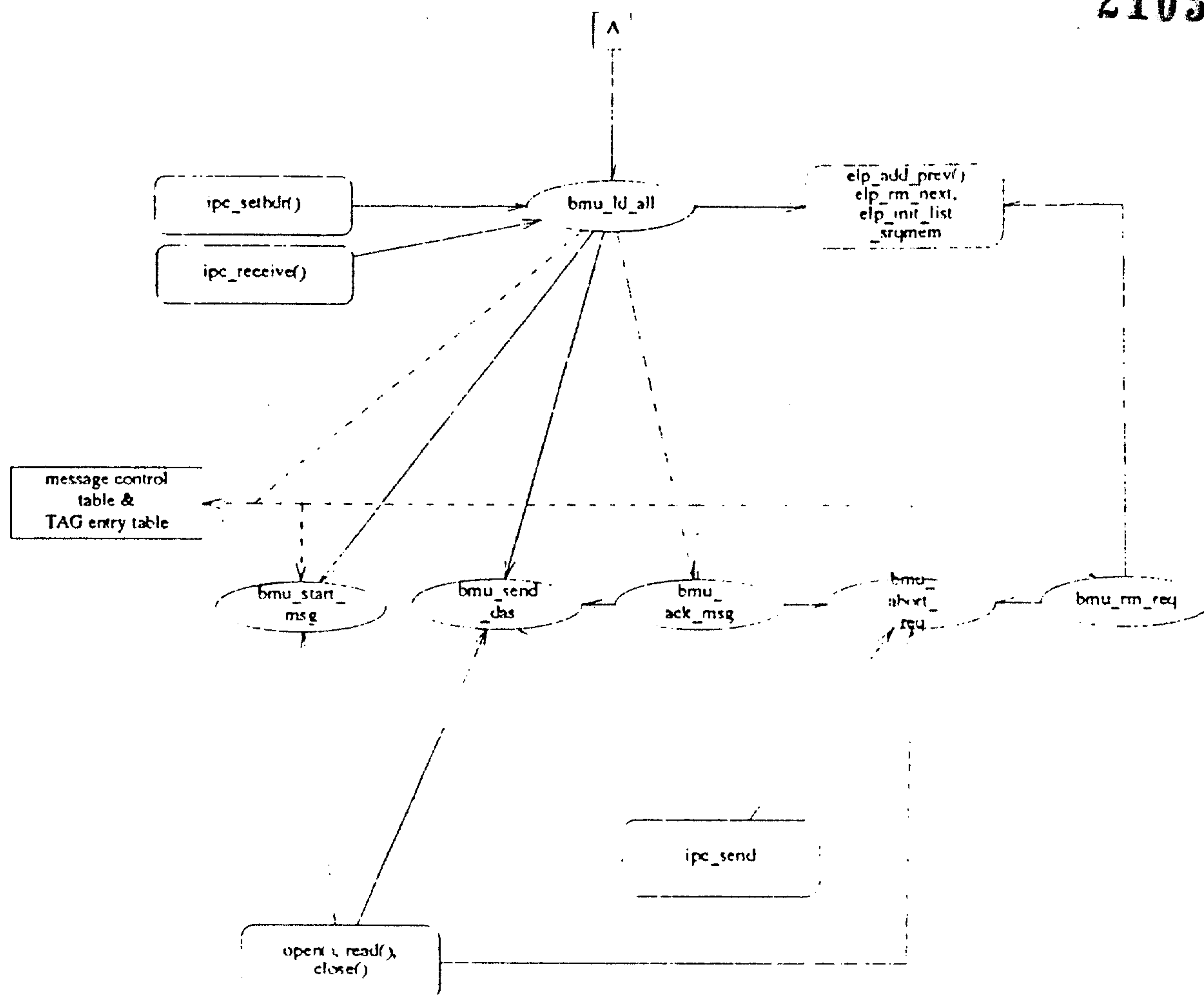


Figure 9. Load all function

372 2.3.3.1 Main()

373 This process is started by the maintenance when the system first comes up. It will receive unit number and its
 374 mailbox id as the arguments. It will call ipc_sethdr and ipc_receive to get ready to receive requests in its mailbox. It
 375 will also allocate queuing space where it will save the in coming requests when it is not idle. This process can
 376 receive any of the following events and it will take action as described.

- 377 1. DAS_LOAD_MESSAGE_HDR, DAS_LOAD_MESSAGE or START_LOAD_EMG_MSG - Request to
 378 download a single message to the DAS card and will call bmu_start_msg() function, if necessary, to handle
 379 the request if the slave process is idle.
- 380 2. LOAD_ALL_MESSAGE - Request to download all of the messages that resides on the DAS card. This is
 381 only done during Cold Initialization of the card. Function bmu_start_msg() is called if the slave process is
 382 idle for each message that needs to be downloaded.

425 **2.3.3.6 bmu_ack_msg()**

426 This function is called by main() after it receives ACK_MSG from the DAS message loader. It will compare the
 427 message sequence number sent with ACK_MSG with what was expected. If it is the correct message sequence
 428 number it will call bmu_send_das() to continue with downloading. Otherwise, it will call bmu_abort_req() function
 429 to stop the downloading process because it received the message sequence number it was not expecting. This could
 430 mean loss of LOAD_DATA message due to IPC error. It will also send MSG_LD_ABORT to DAS message loader
 431 to stop the downloading process on the card.

433 **2.3.3.7 bmu_abort_req()**

434 This function is called by main() after receiving MSG_LD_ABORT request from the DAS message loader or when
 435 bmu_ack_msg() function detects an error due to receiving invalid message sequence number. This function will
 436 inform the requestor of the error, if necessary, and prints out the mnt_inform message. It will update message
 437 control table and message info table to indicate that message is not available on the card.

439 **2.3.3.8 bmu_rm_req()**

440 This function is called to remove or abort the previously made request. It will check if the request is currently being
 441 processed. If it is, it will call bmu_abort_req() else it will search and remove the request from its request queue.

443 **2.3.4 Downloader Performance**

444 The performance goal for the Downloader is to achieve a 2:1 ratio of Downloading time to play time. In other
 445 words, a message that takes 1 minute to play would take 2 minutes to download. Following changes will be made to
 446 slave downloader to make this possible.

- 447 1. LOAD_DATA event will carry 2k bytes long message instead of 512 bytes. Since size of data (DATA_SIZ)
 448 is a #define in the <cp/_hdr/mail_msg_tbl.h> file, DATA_SIZ will only need to be changed from 512 to 2048.
- 449 2. Also, LOAD_DATA header information will be put into ipc_header area and data part of message will only
 450 contain the PCM data. This will enable DAS message loader to move the PCM data directly from the IPC
 451 buffer to the DPRAM.
- 452 3. The file header will be 2048 bytes long, which will force the data to start on a disk sector boundary which will
 453 make a disk access quicker. This will be implemented by making the size of MSG_FILE_HEADER structure
 454 to be 2048 bytes long. This structure is in file <cp/cp/msg_ld_unld/msg_ld_unld/_hdr/msg_file.h>.
- 455 4. Lower the IPC priority level for ipc_send downloading events so call processing events can be processed as
 456 soon as possible. This will improve the overall performance on call processing.

458 **2.3.5 Message Downloader on DAS**

459 Multiple DAS cards does not change the way the DAS message loader works on the DAS card. Only changes
 460 needed are the following changes that would make the downloading process faster.

- 461 1. When the DAS message loader receives the LOAD_DATA message, it will receive the header information in
 462 the IPC header area and the data area of the message will contain just the PCM data. The loader will copy the
 463 PCM data from IPC data area to the DPRAM directly. This will avoid the DAS message loader to copy the
 464 message from IPC data area to the memory allocated by the loader and then copy it again to the DPRAM.
- 465 2. The LOAD_DATA data area will carry 2K bytes long message instead of 512 bytes. This can be
 466 accomplished by changing the #define as described in Downloader Performance section.

468 2.4 DSP Message Downloader on BMU

469 The DSP message downloader on BMU is an non_essential process that moves TMS code and answer detector code
 470 from BMU disk to DSP card. This is an non_essential process which will be forked during Cold initialization and
 471 once it is done downloading of TMS code and answer detector code it will kill itself. The goals of this Detailed
 472 Level Design Unit is to support downloading to as many DSP cards that are possible. Two or more cards may be
 473 downloaded in parallel, but only one code at a time can be downloaded to a single DSP card.

474 This DLDU is concerned only with the portion of message downloading that resides on the BMU.

476 2.4.1 DSP Downloader Process on BMU

477 The following bubble diagram shows the main components of the DSP downloader process on BMU. Each
 component of master downloader process will be described in next sections.

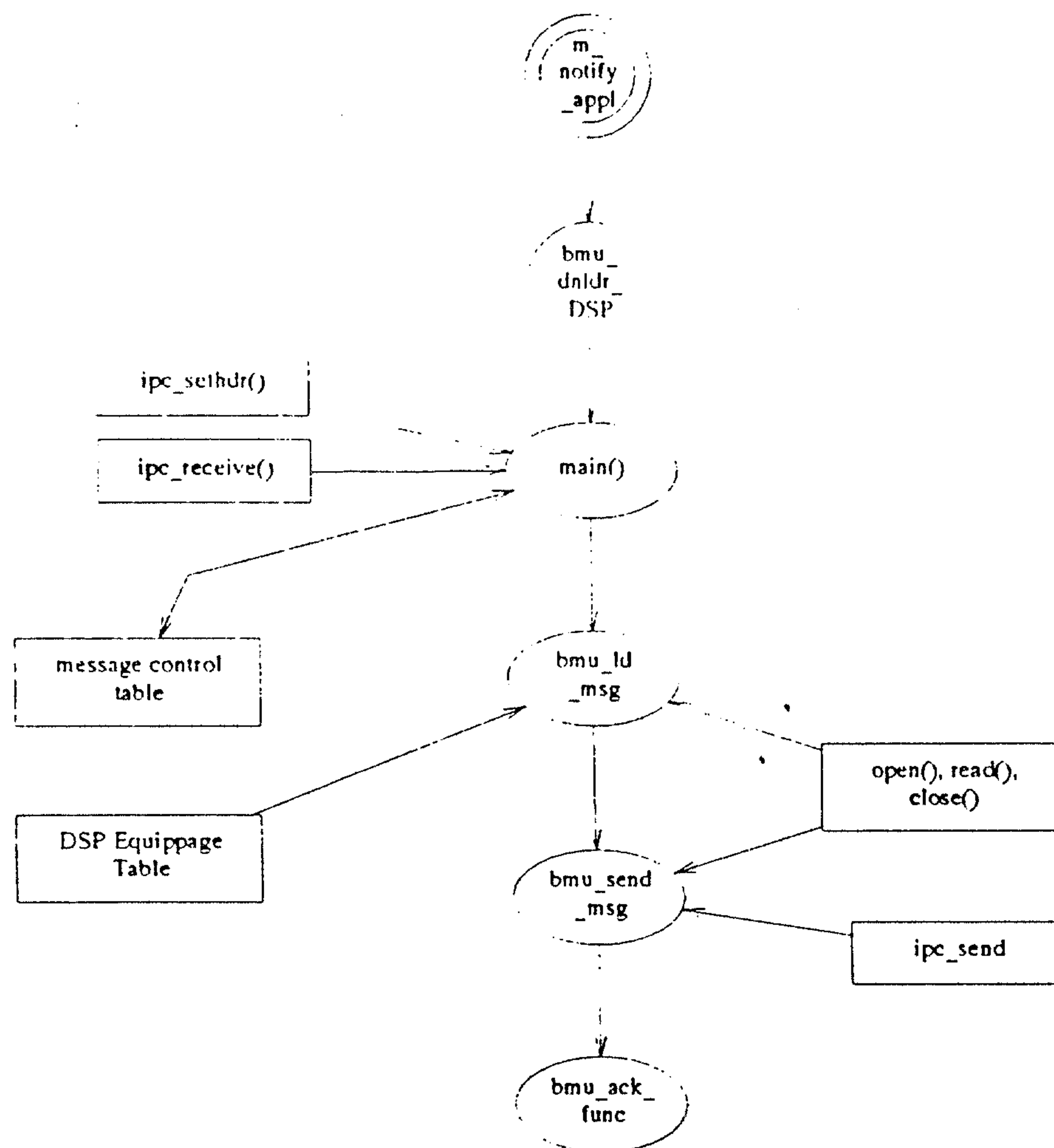


Figure 10. DSP Downloader Process on BMU

479 2.4.1.1 Main()

480 The downloader process will be forked during Cold initialization by the maintenance software configuration
 481 functions. This function will create its dynamic mailbox and starts the downloading process by calling
 482 bmu_ld_msg() function.

484 **2.4.1.2 bmu_ld_msg()**

485 This function is a release 1 function. In release 1, this function handled downloading of both DAS and DSP cards.
 486 All the code dealing with downloading of DAS card will be stripped out.

487 This function will open the file containing DTMF code or answer detector and will call bmu_send_msg() function to
 488 start downloading to the DSP card. Since the order of which code to download is not important, it will call
 489 bmu_send_msg() function to download DTMF code first followed by answer detector code.

490 This function will also create a bitmap to inform DSP downloader of what code to download on the three TMSs.
 491 This information will be available in the DSP equippage table.

492 After the downloading of DTMF code and Answer detector code is completed, this function will close the files and
 493 inform receiver LGH with ANY_GROUP_IN_SERVICE event so all of the ports on the DSP card can be put into
 494 service.

496 **2.4.1.3 bmu_send_msg()**

497 This function will remain as it is besides for the changes described in the **DAS Downloader Performance** section.
 498 Same performance changes will need to be made in DSP downloader on BMU as it is done for DAS.

500 **2.4.1.4 bmu_ack_func()**

501 Same as release 1 code. No changes necessary.

503 **2.4.2 DSP performance changes on BMU**

504 There is no real performance issue with the DSP downloading, but the following changes will be made to make it
 505 consistent with the downloading of DAS card.

506 1. LOAD_DATA event will carry 2k bytes long message instead of 512 bytes. Since size of data (DATA_SIZ)
 507 is a #define in the <cp/_hdr/mail_msg_tbl.h> file, only the #define will need to be changed from 512 to 2048.

508 2. The file header will be 2048 bytes long, which will force the data to start on a disk sector boundary which will
 509 make a disk access quicker. This will be implemented by making the size of MSG_FILE_HEADER structure
 510 to be 2048 bytes long. This structure is in file <cp/cp/msg_ld_unld/msg_ld_unld/_hdr/msg_file.h>.

511 3. Lower the IPC priority level for ipc_send downloading events so call processing events can be processed as
 512 soon as possible. This will improve the performance on call processing.

514 **2.4.3 DSP Message Downloader on DSP**

515 In release 2, answer detector will be downloaded to DSP card along with the DTMF code. Since DSP message
 516 loader already checks the bitmap to determine which TMS the downloaded code should be loaded on, nothing needs
 517 to be changed on the downloader to handle downloading of answer detector code.

519 **2.4.4 DSP performance changes on DSP**

520 Changes described in **DSP performance changes on BMU** will effect the way DSP downloader works because it
 521 uses the same #define, events and structures used on BMU downloader.

383 3. READY_TO_LOAD - This event is received after START_LOAD_MSG is sent to the loader on the DAS
384 card. This indicates that DAS loader is ready to receive data from BMU. Function bmu_send_das() is called
385 to start the downloading of data messages.

386 4. ACK_MSG - This event is received for each LOAD_DATA event sent to the DAS card. Function
387 bmu_ack_msg() is called to verify this message.

388 5. MSG_LD_ABORT - This event is received when the DAS loader finds an error in downloading. Function
389 bmu_abort_req() will be called to handle this event.

391 2.3.3.2 bmu_ld_all()

392 This function is called by main() after receiving load all message request. Until the completion of downloading of
393 all messages, this function will handle the loading messages, ACK_MSG, MSG_LD_ABORT, and
394 READY_TO_LOAD, immediately and will queue any other requests for downloading.

395 First it will send ANY_GROUP_IN_SERVICE event to TAG LGH to put the ports in the TAG group in service. It
396 will put in service the TAG groups that does not require any downloading (Tones, emergency, common pool,
397 Personal announcement and ACD Mail).

398 Secondly, it will go thru the message control table and find all the messages with the status ON_DISK. For each
399 message that is on disk, it will start the downloading process by calling bmu_send_msg() function followed by a call
400 to bmu_ld_all_sub() function. After downloading group of messages, it will send ANY_GROUP_IN_SERVICE to
401 the TAG LGH to put the ports in service.

403 2.3.3.3 bmu_start_msg()

404 This function is called by main() after receiving request to download message(s). This function will send
405 START_LOAD_MSG event with the DAS message number, message size and mailbox id to DAS message loader
406 to start the loading process on the card.

408 2.3.3.4 bmu_ld_all_sub()

409 This function is called when bmu_ld_all() function finds any message to download. This function will take control
410 of the downloader process and will requests relating to downloading as they arrive. It will queue any other
411 downloading requests.

413 2.3.3.5 bmu_send_das()

414 This function is called by main() after receiving READY_TO_LOAD event from DAS message loader to start the
415 downloading of data. It will be also called by bmu_ack_msg() function to continue the downloading process after
416 receiving an acknowledgement of the LOAD_DATA message.

417 This function will be kept from Release 1 code. In release 1, this function handled sending of
418 START_LOAD_MSG, LOAD_DATA messages and also received all the acknowledges for LOAD_DATA.
419 Because main() function will be receiving all the mail, seperate function needs to be created to handle each of them.
420 Code that handles START_LOAD_MSG will be moved to bmu_start_msg() function and handling of acknowledges
421 will be done by bmu_ack_msg() function. So only task this function will be handling is the sending of
422 LOAD_DATA to the DAS card.

423 After it has sent the last data message, it will close the file on disk.

2.5 Queueing

The slave uploader and downloader processes will queue requests received while an upload or download is in progress. The queue will be created and managed by using the existing list processing package, `elp`. The queue will be implemented as a doubly-linked list. When a request is to be put into the queue, memory will be allocated, the structure filled in, and the function `elp_add_prev()` called. When an element is removed from the queue, it is processed as any event, and the memory freed.

The memory from which elements of the queue are allocated is reserved when the slave process is created. This memory is reserved by using the function `srq_mem()`. If there is no space available when an attempt is made to allocate an individual queue element, the original requestor will be sent a failure indication.

A queue element contains links used by the queueing package, and space for the event. The structure for a queue element is shown in the section on Data Definitions.

2.6 Data Organization

2.6.1 Message File Header

The message file header must be 2048 bytes long.

```
typedef struct {
    unsigned int file_type;    /* PCM_DATA or TMS_DATA */
    unsigned int length;      /* length of the data */
    unsigned int checksum;    /* checksum over the data */
    char file_desc[L_FILE_DESC]; /* ASCII description (opt) */
    unsigned int msg_type;    /* voice, tn, both */
    unsigned int time_stamp; /* date & time of recording */
    struct {
        unsigned char digit_count; /* how many digits in TN */
        char digits[6]; /* digits (as BCD!) */
    } phone_number;
    unsigned int flash_list[MAX_FLASH]; /* list of flash ann. */
    char filler[HDR_FILLER_SIZE]; /* fill it out to 2K */
} MSG_FILE_HEADER;
```

The time stamp will be carried throughout the system as seconds. It will be converted to a display format (i.e. MM:DD HH:MM:SS) by whichever entity is going to control the display.

2.6.2 Request Queue Structure

Both the uploader slave processes and the downloader slave processes will queue requests that can't be handled at the moment they're received. The structures for both queues must handle all the events expected by that process.

2.6.2.1 Upload Requests

The uploader expects `DAS_UPLOAD`, `DAS_UPLOAD_HDR`, and `DAS_DEALLOCATE_MSG` events.

```
typedef struct {
    DBL_LIST link;
    union {
        DAS_DEALLOCATE_MSG de_alloc;
        UPLOAD_MSG upload;
        UPLOAD_MSG_HDR upload_hdr;
    } request;
} UPLOAD_Q_ENTRY;
```

2.6.2.2 Download Requests

572 The downloader expects LOAD_ALL_MSG, DAS_LOAD_MESSAGE and DAS_LOAD_MESSAGE_HDR
573 events.

```
574 typedef struct {
575     DBL_LIST link;
576     union {
577         DAS_LOAD_MESSAGE das_load_msg;
578         DAS_LOAD_MESSAGE_HDR das_load_msg_hdr;
579     } request;
580 } DOWNLOAD_Q_ENTRY;
```

582 2.6.3 Constants/Literals

583 The size of the packet of data passed between the BMU and the DAS must be equal to the disk sector size. This
584 makes for efficient reads from and writes to the disk. This value will be a manifest constant (i.e. a #define value) so
585 if the value has to change, it has to be changed in only one place.

```
586 #define DATA_SIZ      2048
```

587 The file header must be 2048 bytes long to force reads and writes of the following data to start on disk sector
588 boundaries. The information in the file header does not take up that much space, so we have to pad it out. This
589 value is the size of that pad area.

```
590 #define HDR_FILLER_SIZE      1976
```

592 2.6.4 Mail Boxes

593 The mailbox table will be created during initialization by cp_subinit_bmu function. In order for cp_subinit_bmu
594 function to create this table, following needs to be done:

- 595 1. Create sched entry, CP_LDR_MBOX_TBL_SE, in escad_entries.h
- 596 2. Add an entry in the cp/cp/sub_init/sub_init.bmu/_src/cp_smm_bmu.i file. This entry should contain the
597 following:
598 Block Name - "ldr_mbox_tbl"
599 MODE - Read/Write (ESMM_URWS)
600 SCAD ID - CP_LDR_MBOX_TBL_SE
601 ENTRIES (Size of table) - Since software limit for number of DAS card is 32, size will be 32 times the size of
602 loader mailbox table (LDR_MAIL_BOX).

603 The master uploader process will use the CP_DAS_ULDR_MBOX and DAS_ULDR_MBOX, both already
604 established for the uploader. The master downloader process will use the CP_DAS_LDR_MBOX and
605 DAS_LDR_MBOX, both already established for the downloader.

606 Each slave downloader and slave uploader process will have a mailbox of its own for receiving events from
607 the master process and from the DAS. These mailboxes will be created dynamically by the m_notify_appl
608 process when the slave process is created. The mail box id will be stored in a table accessible to both the
609 master and slave processes.

610 When the master process receives a request that is to be handled by a slave process, it (the master process)
611 looks in this table to see if the appropriate DAS is active. If it is, it uses that mail box id to pass the request to
612 the slave process for that DAS. That mail box will also be used by the DAS application to send responses
613 back to the slave process.

2.0 GENERAL DESCRIPTION

2.1 General

The Digital Audio Source (DAS) Card provides the G3000E system with call progress, DTMF, and MF tones, as well as with origin, delay, and intercept announcements. Any such PCM-encoded "message" stored in the card's memory can be delivered to the G3000E timeslot interchange network for output on any channel of the system. The DAS also provides the G3000E system with the ability to "record" PCM from any switched channel in the system. This recording capability can be employed by the switch to formulate the card's set of announcements or to serve in an emergency recorder capacity. DAS Cards are part of the G3000E Network Termination Subsystem as illustrated in Figure 2.1-1.

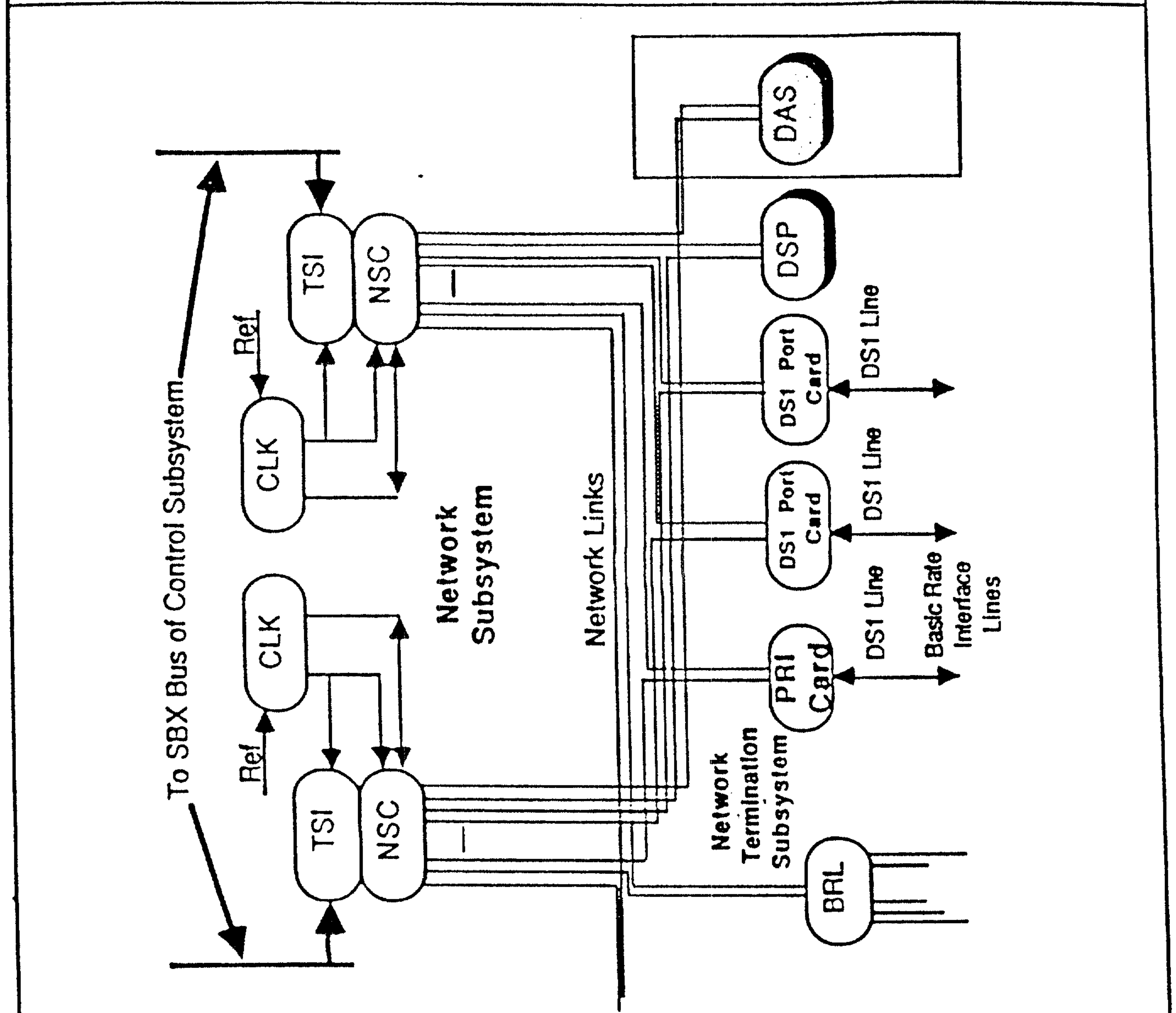
The standard DAS Card is provided with sufficient memory to store four minutes of PCM-encoded message data. With the addition of optional expansion memory, twelve more minutes of PCM storage can be gained. This message memory can be segmented into 256 elements of random length, each segment corresponding to one message. Two models of the DAS Card exist - one with 48 channels of access to the system's timeslot interchange network provided and a second with 96 channels of access provided. The DAS can be configured to output any message on any of these channels and a given message can be output on multiple channels simultaneously. A given message can either be specified to run to completion once as output to a channel or it can be made to repeat indefinitely. Commands must be sent via the Network Links to a DAS Card indicating the number of a channel on which output is desired along with the number of the specific message to be output and its characteristics.

2.2 Typical Applications

In its typical applications, the DAS Card will be downloaded at Initial Program Load with the desired message repertoire. Additional (or replacement) messages can be added from tape using the system's tape reading facilities with subsequent download of this information to the DAS Card. Additionally, the switch can utilize the DAS Card's recording capability to place new messages directly into the DAS Card's message memory. A protocol must be developed and administered by the switch's control complex to facilitate the direct recording of PCM into the card's message memory.

The DAS Card will play a central role in G3000E ACD and PBX applications. A typical ACD application requires up to thirty city-of-origin announcements and up to five delay announcements. A typical PBX application requires up to thirty intercept announcements. Both ACD and PBX applications need the capability to source a variety of tones. The DAS can be configured in either of two fashions to source these "messages": they can be provided on a "demand" basis with the DAS receiving a command to initiate a specific message on a specific channel or each message can be repeated continuously on its own dedicated channel from the DAS. In the case of continuous tones, the dedicated channel approach is clearly indicated. The dedicated channel approach might even be appropriate for certain types of announcements where, by

Figure 2.1-1 G3000E Switch Architecture



holding a given announcement to an N second duration, the receiving party can be (asynchronously) connected for 2N seconds to insure its complete receipt.

2.3 Functional Feature Set

Figure 2.3-1 Functional Feature Set

- Storage for 4 minutes of PCM standard with optional expansion to 16 minutes.
- Generates system tone plant.
- Sources origin, delay, and intercept announcements.
- Tone and announcement "message" repertoire downloadable.
- 256 distinct messages supported.
- PCM recording capability.
- Connection to either 48 or 96 channels of a G3000E switch.

3.0 FUNCTIONAL CIRCUIT DESCRIPTION

3.1 General

The DAS Card is composed of two printed circuit assemblies which are rigidly joined and interconnected by ribbon cable. The main board of the DAS is assembly 91-E01400-A and a block diagram of its circuitry is given in Figure 3.1-1. The DEX board is assembly 81-001472-A and a block diagram of its circuitry is given in Figure 3.1-2. The main board contains four mounting holes into which plastic spacers are pressed. The DEX board contains an aligned set of four mounting holes which can be pressed onto the spacers extending from the main board, rigidly connecting the two boards. The circuitry on the main board is electrically connected to that on the DEX board through ribbon cable secured on each end by a pair of 50-pin connectors.

The main board of the DAS contains a 96-pin female DIN connector mounted such that the card can be secured into a mating connector on the backplane of a slot in a Network Termination Shelf. The DAS Card will occupy two card slots of a Network Termination Shelf due to the stacking of its two printed circuit boards and thus will only fit into slots 1 through 15 of a Network Termination Shelf. While each of those fifteen slots will accept a DAS Card, care should be taken in positioning any 96 channel models of the DAS Card in a Network Termination Shelf. With a 96 channel DAS, the card can only be placed in slots 1 through 13 AND the two card slots to its right MUST be left unoccupied. This restriction on the placement of 96 channel DAS Cards is due to the distribution of Network Link connections on the Network Termination Shelf backplane and the requirement that four such Network Links must be reserved for a 96 channel DAS. There are no restrictions on the placement of 48 channel DAS Cards in the Network Termination Shelf. Note that any unoccupied slot in a Network Termination Shelf can be filled with a "blank" card which contains only a faceplate matching the G3000E color scheme in order to provide the shelf with a more aesthetic appearance.

The main board of the DAS receives -48V input from pins on the 96-pin backplane connector. A DC-DC converter is present on the main board to derive from this input the 5V required for circuitry on both the main and DEX boards. The main board also houses the Network Link interface circuitry which provides access to the G3000E switch through other pins of the 96-pin backplane connector, the DRAM providing the card's message storage, and the DRAM Controller circuitry. The DEX board contains a 68000 microprocessor, SRAM for the card's application program, EPROM to facilitate downloading of the card's application program and message complement from the switch's control complex, a sequencer circuit which controls the message output of the DAS, and a Dual-Port RAM for communication between the microprocessor and the sequencer.

Figure 3.1-1 Main Board Circuitry

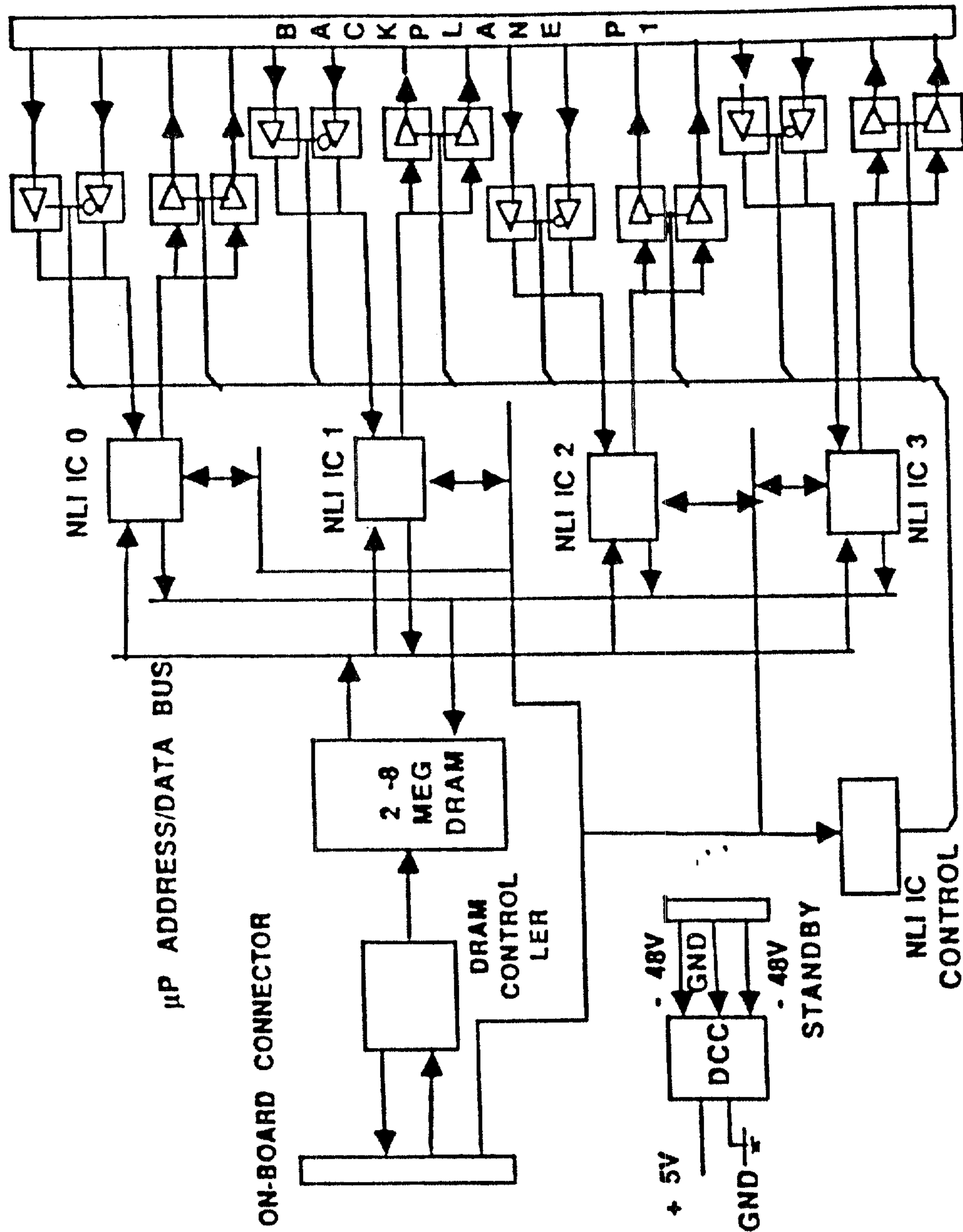
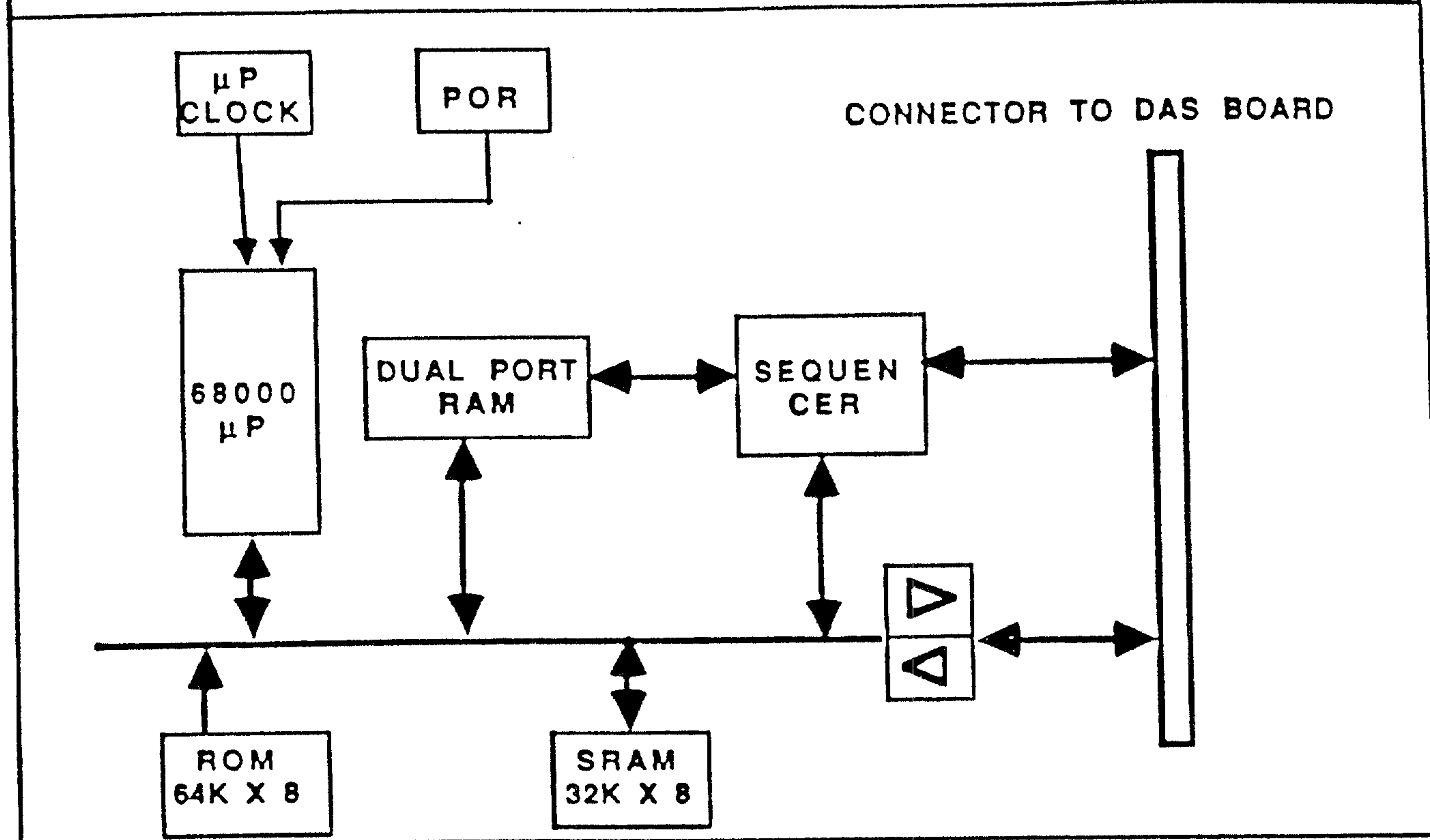


Figure 3.1-2 DEX Board Circuitry



Upon power-up, the DAS Card's 68000 will begin executing its ROM program. This program must initialize the card and prepare for the downloading of an applications program and the card's message complement from the switch's control complex. This downloading operation will occur over the Network Links to the card and will be facilitated by an NLI IC. The card's 68000 will be interrupted by an NLI whenever a block of downloaded data has been received from the control complex. The 68000 must retrieve this data from the NLI's internal Receive FIFO and place it in the appropriate part of memory - into SRAM if the received data represents part of the card's application program or into DRAM if the data represents part of the card's message complement. Note that a table which indicates the address range in DRAM corresponding to a given message number must be downloaded as a supporting part of the application program. Upon conclusion of the download process, the 68000 will begin to execute the applications program which was placed in the SRAM. The DAS Card must subsequently receive instructions from the control complex via the Network Link as to how each channel on the card is to be configured - whether a given channel is to be used for message output along with a specification of the desired output message number or whether that channel is to be used as a recording input along with a specification of an associated DRAM address range for placing the received PCM data.

5.0 SOFTWARE CONSIDERATIONS

5.1 Programmer's Model

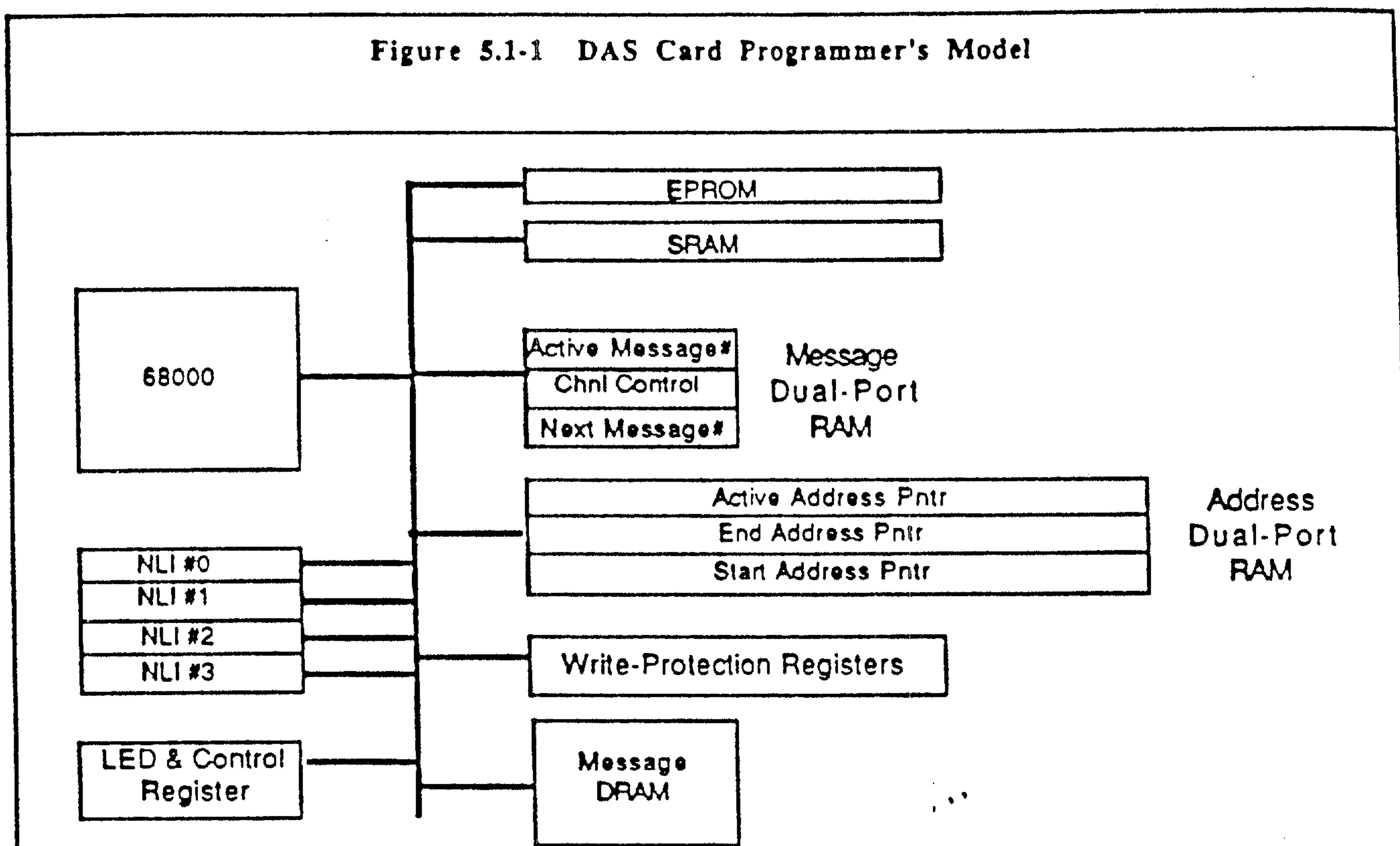


Figure 5.1-1 indicates the various elements of the DAS Card addressable by the card's microprocessor.

Upon reset, virtually all registers within each of the card's NLIs will be cleared and the 68000 will begin executing from its reset vector address. The 68000 must then program the desired channel numbers into each NLI's Position Register and report its presence to the NSC via a Network Link. The NSC will regularly check each of its NLIs to determine if a message has been received from its associated Network Termination Card and the message from the DAS Card will ultimately be acknowledged.

A quiet tone (constant value) function code should be loaded as both the Next Message number and the Active Message number for each of the 96 channels. The "sequential" and "play" status should be set up as

Channel Control data for all 96 channels. A loop should be begun which assures that all 96 channels constantly have a Next Message number queued, insuring a continuous flow of PCM for each channel.

5.2 General Requirements

The software on the card performs overall control of the DAS and performs communications and maintenance functions. It communicates with the control complex through the Network Link and sets registers inside the NLI ICs to select which PCM timeslots are activated.

The software performs the queueing function of PCM messages for each channel. It monitors the status reports from the card's sequencer and either pops from the queue and/or reports channel status to the control complex.

The software interprets commands from the control complex and sets the correct codes to the card's sequencer. It is used during message recording to keep track of available space in the card's DRAM and set appropriate pointers in the card's Dual-Port RAMs.

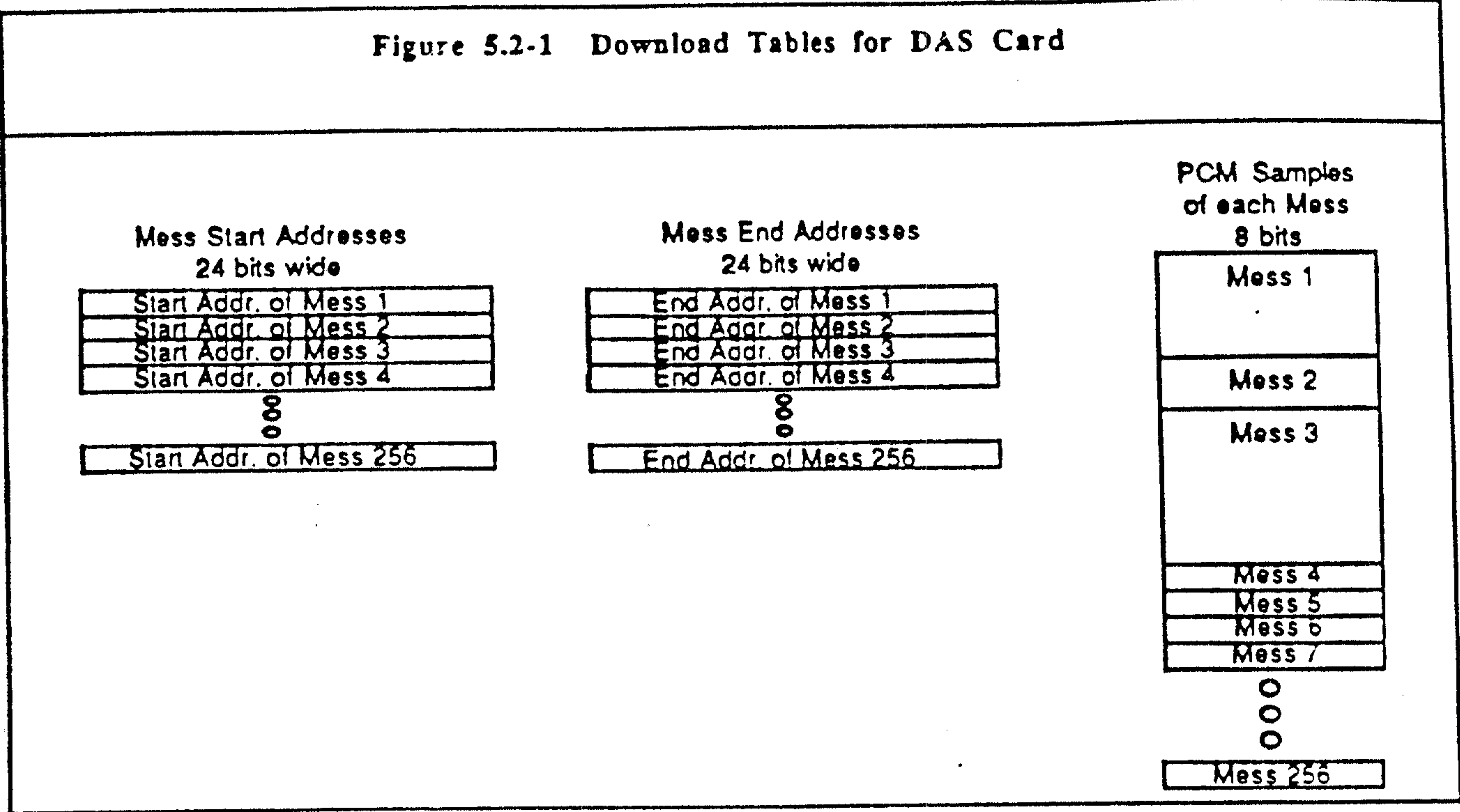
The software controls all upload and download operations to/from the control complex.

A routine must be provided in the card's EPROM to initialize the card and to handle communication between the 68000 processor and the NLI ICs. The information which must be downloaded to the DAS includes:

- 1) The DAS application software will be loaded into the card's SRAM. This software involves executing a loop to assign messages to all 96 channels and monitoring the status reported on each channel.
- 2) The PCM samples will be downloaded to the card's DRAM. Up to 8 Megabytes of memory or 16 minutes of messages can be stored on this card. This PCM data includes samples of the followings:
 - The PCM samples for each of the call progression tones.
 - The PCM samples for each of the DTMF/MF tone digits.
 - The PCM samples for each of the voice messages.
- 3) A file containing a concatenated form of all the messages destined for a particular DAS Card will contain a header that contains each message's starting and ending address. This is the file that will actually be installed in the card's Message and Address Dual-Port RAMs.

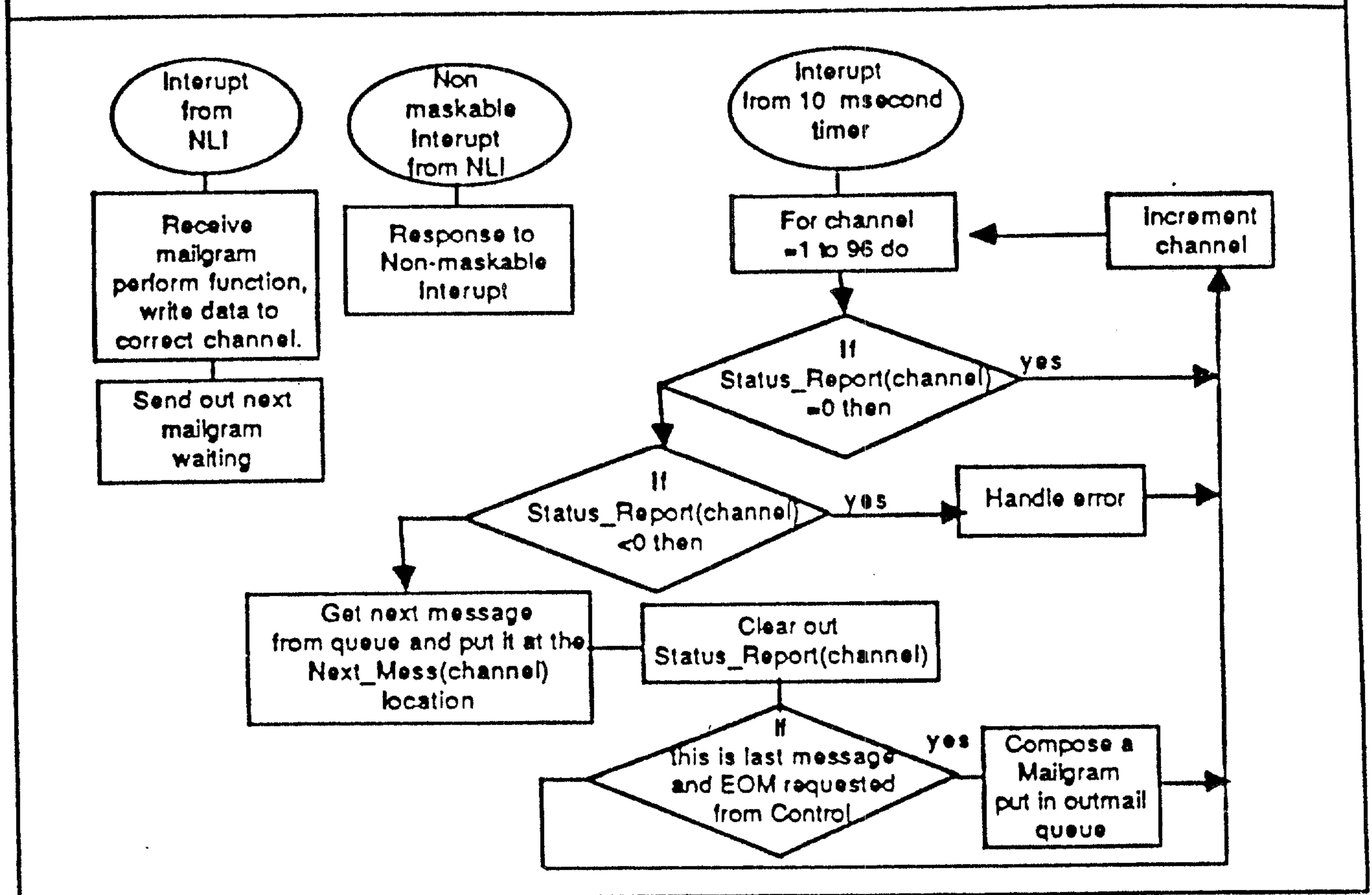
Figure 5.2-1 indicates the starting and ending address data along with the actual PCM data which must be downloaded.

Figure 5.2-1 Download Tables for DAS Card



The 68000's main functions are in keeping messages queued for each channel in the Dual-Port RAM interface to the card's sequencer and in controlling communications with the control complex via the Network Links. The diagram below illustrates one possible means of structuring the card's application software.

Figure 5.2-2 Applications Software Functional Flow



5.3 Control Codes

The following single byte codes are examples of functions required to control the Digital Audio Source Card. They are sent to the Digital Audio Source Card from the Network Shelf Controller over the Network Link and are received and interpreted by the 68000 on the DAS. The DAS Card's 68000 can translate these functions into the proper subroutines necessary to control the hardware in the desired fashion.

Figure 5.3-1 68000 Command Example

Bytes of transmission for one command from the Controller.

Function Code 0-255	Channel 0-96	# Bytes following 0-255	A number of bytes between 0 and 255.
---------------------------	-----------------	-------------------------------	---

Channel 0 indicates a function
that is independent
of any particular channel.

— A Control Instruction like one of the ones illustrated below.

Each command from the Network Shelf Controller sent via the Network Link could be of the form shown in Figure 5.3-1. A function code is followed by a channel number. These are the types of commands necessary to fully control the card: self test, upload or download a message, configure a certain channel for a given output, start or stop a message, etc. These are illustrated below in Figure 5.3-2.

Figure 5.3-2 Types of Instructions for DAS Control

Card Control☐ Self Test Card☐ Initialize Card**Message Control**☐ Make this message number have the begin and end address following.☐ What is the begin and ending address of the following message number?☐ Download the following message and call it this message number.☐ Upload the following message.**Channel Control**☐ Turn on signalling bit insertion to this channel.☐ Turn off signalling bit insertion to this channel.☐ Abort current message on this channel, replace it with this message, and Report End of Message String.☐ Abort current message on this channel and replace it with this message, Do Not Report End of Message String.☐ Concatenate this message to the queue for this channel number and Report End of Message String.☐ Concatenate this message to the queue for this channel number, but Do Not Report End of Message String.

In addition, there are controls that help the card do recording of PCM. Figure 5.3-3 shows a minimum set of recording commands. It is assumed that the pointer controls are administered by the control complex. Alternatively, more intelligent commands could be given with the interpretation being performed on the DAS card itself. These higher level commands could be "Play a Recording", "Rewind a Recording", "Stop", "Fast Forward", or "Record".

Figure 5.3-3 Further Instructions for Recording

Recording Control

☐

Change this channel to a read from PCM bus.

☐

Change this channel to a write to PCM bus.

5.4 Reply Codes

The following are codes that the DAS could return to the NSC to report status.

Figure 5.4-1 Examples of DAS Status Reports

Bytes of transmission for one report to the Controller.

Function Code 0-255	Channel 0-96	# Bytes following 0-255	A number of bytes between 0 and 255.
---------------------------	-----------------	-------------------------------	---

Channel 0 indicates a function
that is independent
of any particular channel.

A Single byte return code of 1
will mean function successful.

A single byte return code of 0
means function unsuccessful.

Figure 5.4-2 Status Report Coding

Transmission☐

Return Function Code and channel number with either a 1 for successful or 0 for failed.

☐

Gives a function to initiate action like reporting ready for service or failure detected.

☐

Gives a function Code and channel number indicating this transmission is part of an upload of data.

 * GENERAL COMMENTS *

There will only be 48 channels on the Iss. 2 DAS. All references to a 96 channel DAS in this document are no longer valid and should be ignored.

The Iss. 2 DAS will have 8Mbytes (16 minutes) of PCM storage. There are no longer any options pertaining to the amount of PCM storage on the DAS card and any references in this document to "optional" amounts of PCM storage should be ignored.

The Iss. 2 DAS will contain 128kX16 of SRAM. References to the amount of SRAM in this document (especially references to "optional" amounts of SRAM) should be replaced with the 128kX16 figure.

The Iss. 2 DAS will contain only a single red LED. References to other than a single LED in this document should be ignored.

 * OVERVIEW OF CHANGES *

It should be noted that section 5.0 of the circuit description is no longer accurate as a result of the numerous improvements being made to the DAS card in terms of performance and functionality. The following paragraphs serve to provide an overview of how the new issue DAS card is to be controlled by software.

First, the error detection logic has been expanded with the addition of a Bus Error Latch (BER_LTH) for capturing the causes of BERR's when certain error conditions occur. The BER_LTH is to be read by the 68000 as part of the Bus Error routine. The latch is cleared by writing bit 0 to a zero. The functions monitored by the BER_LTH are described in subsequent paragraphs.

Error detection has been enhanced on the sequencer with the addition of some new on-line checks:

- 1) The Active Address of a channel is now compared to its Start Address for the currently active message. This means that the Active Address will now be checked for BOTH boundary conditions.
- 2) A parity bit has been added to the Message Dual-Port RAM. This will help insure the integrity of the message data and the state of the play/record bit. This parity bit must be generated by software (such as is done with APM memory control on many G3000E circuit cards) and should reflect even parity across the entries to the Message Dual-Port RAM.

If an address or parity error is detected by the sequencer, the error type and the number of the channel being served when the error occurred are loaded into an ERROR FIFO and an interrupt to the 68000 is generated. The sequencer error can be disabled if desired by a control register bit.

The End-of-Message (EOM) detection capability of the card has been enhanced by the addition of an EOM FIFO. Once the sequencer detects that a channel has reached the EOM, the number of the channel will be loaded into the EOM FIFO and an interrupt to the 68000 will be generated. This means has been employed to insure that the EOM indication for a channel persists longer than 125uS,

which had been a problem with the prior design.

The Sequence/Repeat mode selection for each channel no longer has any meaning. All channels operate in what previously referred to as Sequence mode. If a message is to "Repeated" on a given channel, its Next Message value should be specified to be the same as during the previous message. (Note: once written, a Next Message value will not change until altered by the 68000!) Interrupts will occur at each EOM of the channel in such a "repeat" message scenario, however.

Care must be taken when altering the Next Message data in the Message Dual-Port RAM. Next Message should only get changed in response to an EOM interrupt on the associated channel.

The means for causing a channel Abort has been changed due principally to worst case design problems with the current issue card's architecture. The Abort bit has been removed from each channel's entry in the Message Dual-Port RAM and is now controlled from a separate register. This register can only accomodate the Abort of a single channel at a time. To cause a channel to Abort, its channel number is written into the Abort Register latch at the same time that bit 7 of the latch is set to a logic one. When the Abort function is actually performed by the sequencer, an Abort interrupt will be given to the 68000.

With the aforementioned scheme of making use of interrupts to the 68000, it is no longer necessary for the processor to poll the Status Bytes of each channel to determine what a course of action to take. For this reason, the Status Byte associated with each channel has been eliminated - Aborts are handled via the Abort Register (and an associated interrupt), EOM is handled by entries into the EOM FIFO (with an associated interrupt), and errors are handled by entries into the EPROR FIFO (with an associated interrupt).

The record function has been enhanced with the ability to determine the last Active Address for a message which has been Aborted. (Note that this function did not actually work in the prior design.) When a channel is in the record mode, the Active Address is written into two different sections of the Address Dual-Port RAM. When it is desired to end a record session, the channel should have an Abort performed on it and after the interrupt is performed the last Active Address for the recording channel will be in that channel's Abort Address location in the Address Dual-Port RAM.

 * SPECIFIC CHANGES *

*** 1

Page 12, Figure 3.1-2 shows an incorrect size on the ROM and the SRAM blocks. The ROM should be shown as a 32K x 16. The SRAM should be shown as a 128K x 16.

Page 18, Figure 4.2-1 is the DAS card memory map. The memory map will change as follows:

Address	Function
0x000000 - 0x00FFFF	Read Only Memory (ROM)
0x020000 - 0x05FFFF	Static RAM (SRAM)
0x060000 - 0x060FFF	Address Dual Port RAM
0x061000 - 0x061FFF	Message Dual Port RAM
0x0E0000 - 0x0E003F	NLI 0 (odd bytes only)
0x0E0040 - 0x0E007F	NLI 1 (odd bytes only)
0x0E0100 - 0x0E0107	82C55 I/O port (odd bytes only)
0x0E0108 - 0x0E0109	End Of Message FIFO (odd bytes only)
0x0E010A - 0x0E010B	Abort channel (odd bytes only)
0x0E010C - 0x0E010D	Error FIFO
0x0E010E - 0x0E010F	Bus Error Latch (odd bytes only)
0x800000 - 0xFFFFFFFF	PCM DRAM

The memory locations for the DRAM, the 82C55, and the first two NLI will remain the same. The memory location for NLIs number 2 and 3 will become unused.

*** 3

Page 20, Figure 4.2.3-1.

The three 1kX8 bit Address Dual-Port RAMS (ADPRAM) have been replaced with a single 2kX16 bit device.

The Star Address, End Address, and the new Abort Address data contained in the Address Dual-Port RAM will be positioned as two contiguous addresses. The lower 16 bits of the PCM address will be in the even word of the Address Dual-Port RAM and the upper 8 bits in the odd word of the Address Dual-Port RAM.

The sequencer address accesses of the Address Dual-Port RAM will be as follows:

A9 = 0 A10 = 0	Active Address. Address bits 0 - 8 represent the 48 channels that PCM data is being transferred on.
A9 = 1 A10 = 0	Start Address. Address bits 0 - 8 represent the message number being referenced.
A9 = 0 A10 = 1	End Address. Address bits 0 - 8 are used the same as Start Address.
A9 = 1 A10 = 1	Abort Address for channel in record mode. Address bits 0 - 8 are used the same as on the Active Address.

In the case of the Active Address and the Abort Address, the lower 6 bits will contain the channel number that the message is being used with.

The 68000 address accesses of the Address Dual-Port RAM will be as follows:

0x060000 - 0x06033F	Unused channels
0x060340 - 0x0603FF	Active Address for 48 channels
0x060400 - 0x0607FF	Start Address for 256 messages
0x060800 - 0x060BFF	End Address for 256 messages
0x060C00 - 0x060C3F	Unused channels
0x060C40 - 0x060CFF	Abort Address for 48 channels

*** 4

Page 22, Figure 4.2.4-1.

The Message Dual-Port RAM is also being changed from a 1kX8 bit device to a 2K x 16 bit part. The control and status locations are no longer in separate locations within the Message Dual-Port RAM. The message data now occupies a 16 bit word, with the lower 8 bits being the message number and the upper 8 bits being the equivalent of the "control" byte. It is now possible to write the Next Message number and the "control" information in a single write cycle. The "status" function has been removed entirely from the Message Dual-Port RAM.

The "control" byte now has only the play/record setting for the channel along with and a parity bit. This parity bit will be over the message number and the play/record bit and must be generated by software. The bit assignments for Message Dual-Port RAM data entries is described below.

Address bit 10 will be used to determine if the access is for the Current Message or the Next Message. (Note: the 68000 should not attempt to either read or write the Current Message data once the sequencer has been initialized.) With A10 = 0, the access will be for the Current Message. If A10 = 1, the access will be for the Next Message.

The addressing for 68000 access of the Message Dual-Port RAM will be as follows:

Address	Function
0x061000 - 0x06119F	Unused
0x0611A0 - 0x0611FF	Current Message for 48 channels
0x061200 - 0x06199F	Unused
0x0619A0 - 0x0619FF	Next Message for 48 channels
0x061A00 - 0x061FFF	Unused

Channel numbers on the DAS are from 0 to 47 as defined by the NLIs on the card. Access of a given channel's data in the Message Dual-Port RAM must use an offset by 0x1A0 - this value being added to the channel number and then logically OR'ed with the address of 0x61800 in order to write that channel's Next Message data. The data written is the Next Message number in bits 0 - 7 and the

"control" data in bits 8 - 15. In order to insure proper operation, any access of the Message Dual-Port RAM should be a word (16 bit) access. The bit pattern should be as described below:

Data bits	Function
0 - 7	Message number
8	Play/Record: 1 = Play, 0 = Record
9	Parity bit over the other bits
10 - 15	Unused, should be zero

*** 5

Page 23, Figure 4.2.4.1-1.

The channel control byte will change as described above.

*** 6

Page 24. The End Of Message (EOM) indication is going to be handled with an EOM FIFO, with an interrupt generated to the 68000 when a channel has finished its Current Message.

The EOM interrupt can be cleared by emptying the EOM FIFO. This device will be at memory location 0x0E0109.

The format of the data will be as follows:

Data	Function
0 - 6	Channel number having completed its Current Message
7	Not used

*** 7

Page 24, section 4.2.5.

There is a reference to SRAM write protect being overridden when the bus error mask is set. This will be changed on Iss. 2 so that when the bus error mask is set and a write to write protected memory occurs, no bus error will occur BUT the memory location will NOT be written.

*** 8

Page 25, Figure 4.2.5-2.

The 82C55 LED and control register bits are going to change.

The main changes are going to be the deletion of two LED controls (green and yellow) and the TL23EN bit. The TL23EN bit is no longer required as the DAS is only going to be available in a 48 channel model.

The bit assignments for 82C55 port A are:

2103204

Bit	Function
6, 7	Unused
5	DRAM lock (prevents 68000 DRAM access); 0 = disable
4	Sequencer on/off; 1 = on, 0 = off
3	Sequencer error disable; 1 = enabled, 0 = disabled
2	Bus error disable; 1 = enabled, 0 = disabled
1	Sequencer error interrupt disable
0	Red LED; 1 = off, 0 = on

The assignments for port B are:

Bit	Function
7	Protect 0x03C000 - 0x03FFFF
6	Protect 0x038000 - 0x03BFFF
5	Protect 0x034000 - 0x037FFF
4	Protect 0x030000 - 0x033FFF
3	Protect 0x02C000 - 0x02FFFF
2	Protect 0x028000 - 0x02BFFF
1	Protect 0x024000 - 0x027FFF
0	Protect 0x020000 - 0x023FFF

Assignments for port C:

Bit	Function
7	Protect 0x05C000 - 0x05FFFF
6	Protect 0x058000 - 0x05BFFF
5	Protect 0x054000 - 0x057FFF
4	Protect 0x050000 - 0x053FFF
3	Protect 0x04C000 - 0x04FFFF
2	Protect 0x048000 - 0x04BFFF
1	Protect 0x044000 - 0x047FFF
0	Protect 0x040000 - 0x043FFF

Note that the write protection afforded by each bit of the 82C55 ports B and C is on 8k word blocks of the card's SRAM.

Page 30, Section 4.2.8, Interrupt priority.

Because the sequencer is going to communicate certain data to the 68000 via FIFOs, the interrupt structure will have some additions.

The new structure will be as follows:

Level	Function
7	NMI
6	10mS interrupt
5	Error FIFO entries present
4	Abort Complete
3	EOM FIFO entries present
2	NLI interrupt
1	Unused

*** 10

Page 34, Figures 4.3-3 and 4.3-4.

The diagram for the message pointer will change in that there is no longer any status going back to the processor from the Message Dual-Port RAM. All status and error conditions are going to be handled by a FIFO.

*** 11

New registers, their attributes, and contents are described below:

Sequencer error port: Address 0x0E010C - 0x0E010D. Read only register.

Bit	Function
8 - 6	These bits are to be encoded for the error data as follows:
8 7 6	
x 0 1	Active Address less than Start Address
x 1 0	Active Address greater than End address
0 x x	Current Message parity error
1 x x	Next Message parity error
5 - 0	Channel number with error

BER_H (Bus error port):

Address 0x0E010F. Read Only register.
All bits: 1 = error

Bit Function

7 - 3	Unused
2	Write to write protected RAM
1	Write to EPROM
0	Access to locked DRAM

Abort channel port: Address 0x0E010B. Read/Write register.

Bit	Function
7	Unused
6	Enable abort; 1 = enable
5 - 0	Channel number to Abort

Rcvd 2103204
.615100 14

Errata; Iss 2 DAS:

Changes from previous errata:

*** 1

The bus error register is no longer cleared by writing bit 0 on Port A of the 82C55 to a zero. The bus error register is now cleared by performing a write function to the bus error register. The data pattern during this operation is a "Don't care".

*** 2

The 68000 will no longer have access to the "active address" portion of the address DPRAM. Accessing this range of memory will result in a "time out" bus error. The address range is from 0x060000 to 0x0603FF.

*** 3

The 68000 will no longer have access to the "current message" section of the message DPRAM. The address range is 0x061000 to 0x0617FF. Accessing memory in this range will result in a "time out" bus error.

*** 4

The 82C55 port assignments have been modified. Some of the functions have been moved to different bit positions and there have been some features added. The write protect resolution have been changed to 16K (word) boundaries from the 8K (Word) boundaries.

Two of the unimplemented bits on port A of the 82C55 are going to be used to reset the two FIFO's in to sequencer circuit. These bits will cause the associated FIFO to clear its buffer and remove any interrupts it may be causing. The bit must be toggled, ie. set to a zero and then set to a one. The bit must stay a zero for at least 25nS in order to perform the reset. During normal sequencer operation the bit must be a one.

The function of the sequencer on/off bit (Port A, bit 4) has been changed. When it is desired to disable the sequencer, the bit should be set to a one. When the bit is set to zero, the sequencer is enabled. There is a time constraint on the bit, it must be left in one of its states for at least 125uS. This must be done because the sequencer is only turned on/off during the frame reset time which occurs every 125uS.

A side effect of disabling the sequencer is to generate a reset for the FIFO buffers. This was done to prevent sequencer interrupts when the sequencer is turned off.

The bus error disable bit has been moved from port A, bit 2 to port A, bit 3. The function of this bit has not been changed.

Port A, bit 2 is now used to disable the sequencer error FIFO interrupts. setting this bit to a zero, the error interrupts are enabled. When to a one, the error interrupts are disabled.

Port A, bit 1 is used to invert the parity bit when the 68000 is accessing PCM DRAM. When this bit is set to a zero, a parity error will be generated when performing a read function on DRAM. Also, while this bit is a zero, incorrect parity will be written to the DRAM.

Port A, bit 0 is now used to control the red LED. Its function has been changed so a one will turn the LED on and a zero will turn the LED off.

Summary for 82C55, port A:

Bit	Function	
7	RST_ERR_FIFO*	Resets error FIFO 0 = Reset FIFO 1 = Normal
6	RST_EOM_FIFO*	Resets EOM_FIFO 0 = Reset FIFO 1 = Normal
5	DRAM_LCK*	Inhibit 68000 access to DRAM 0 = No access 1 = Access allowed
4	I/O_SEQ_OFF/ON*	Disable/Enable sequencer 0 = Sequencer on 1 = Sequencer off
3	BUS_ERR_DIS*	Disable bus error 0 = Error disabled 1 = Error enabled
2	SEQ_ERR_INT_DIS	Disable sequencer error interrupt 0 = Interrupts enabled 1 = Interrupts disabled
1	PCM_PAR_INV	Invert PCM parity bit 0 = Invert parity 1 = Normal parity
0	RED_LED	Red LED control 0 = LED off 1 = LED on

The resolution of the write protect has been changed from 8K (Word) blocks to 16K (Word) blocks. In order to protect a block, the bit should be written to a one.

Summary for 82C55, Port B:

Bit	Function
7	Protect 0x058000 to 0x05FFFF
6	Protect 0x050000 to 0x057FFF
5	Protect 0x048000 to 0x04FFFF
4	Protect 0x040000 to 0x047FFF
3	Protect 0x038000 to 0x03FFFF

- 2 Protect 0x030000 to 0x037FFF
- 1 Protect 0x028000 to 0x02FFFF
- 0 Protect 0x020000 to 0x027FFF

Port C of the 82C55 is no longer used.

Upon power up, all of the bits on both of the 82C55 ports will be a one.

*** 6

The sequencer error FIFO has the error bits encoded differently. The encoding scheme is as follows:

Bit			Error	
8	7	6		
1	1	1	PCM_PAR_ERR	Parity error in PCM path
1	1	0	CUR_MSG_ERR	Parity error in current message
1	0	1	NXT_MSG_ERR	Parity error in net message
1	0	0	START_ADD_ERR	Current active address is less than the message start address
0	1	1	END_ADD_ERR	Current active address is greater than the message end address

New Features:

*** 1

A feature that was requested by the software group has been implemented in the Iss. 2 design. The requirement was for a special message to be reserved that would not generate an End Of Message (EOM) interrupt. This would then relieve software of servicing EOM interrupts for channels that are not being used.

This feature is referred to as Message Idle (MI). The MI causes several things to happen.

1. There is no EOM interrupt generated for a channel set to MI.
2. There is no error checking performed on a channel that is configured for MI.
3. A channel set to MI will not send any PCM sample from DPAM to the NLI. Because of this the NLI will transmit all 1's (Quiet Tone) for that channel.
4. When the sequencer is disabled, ALL channels will be initialized

to MI. This allows some of the hardware to power up to a known state. This also means that if the sequencer is disabled during normal card operation, all of the channels will be in the MI mode when the sequencer is enabled.

5. The only way to terminate the MI mode is to perform an ABORT function on the channel. It is also required that a valid message number be written into the Next Message portion of the Message Dual Port RAM PRIOR to performing an ABORT on the channel desired.
6. If an error is detected by the sequencer logic, the error will be reported via the error FIFO and the channel will be forced to MI. This is done because it would be easy to fill up the error FIFO in the case of a malfunctioning channel.
7. When a message is only required to play once and then go to quiet tone, all that is necessary to do is write the Next Message portion of the Message Dual Port RAM to all 0's. The MI mode is detected by hardware as being a message number of all zeros.
8. Although it may appear that all zeros would imply a record function, in the case of MI there is no record function. When a channel is set to MI, nothing will happen for that channel.

*** 2

Another register has been added to the Iss 2 DAS. This register will allow the CPU to read the empty bit and the full bit for the error and EOM FIFO's. This register is called the FIFO Status Register (FIFO_SR). The address for this register is 0x0E0110, it uses only the low order 4 bits of the odd byte and it is a read only register.

The reason this register has been added is to allow the CPU to determine if the FIFO's have been properly serviced before returning from an interrupt routine that was called as a result of one of the FIFO's generating an interrupt. When the CPU has finished servicing an EOM (or Error) indication, it can read this register and determine if another channel needs servicing before it returns from the interrupt. This is done by reading the appropriate Empty bit (either EOM or Error) and making sure the bit is a zero (no entries in the FIFO) before the return is performed.

If either one of the Full bits read a zero, this is an indication that one (or both) of the FIFO's is not being serviced when it is supposed to. The FIFO's that are used are 1K deep and, in normal operation, should never fill up. A full FIFO is an indication of a serious problem with the subsystem.

Following is breakdown of the FIFO_SR:

Bit	Function
4 - 7	Not used
3	Error FIFO Full - If this bit is a zero, the Error FIFO

is full. A one is normal for this bit.

- 2 Error FIFO Empty - When this bit is a zero, the Error FIFO is empty.
- 1 EOM FIFO Full - When this bit is a zero, the EOM FIFO is full. A one in this bit position is normal.
- 0 EOM FIFO Empty - The EOM FIFO is empty when this bit is a zero.

We claim:

1. In an automatic call distributor having a plurality of agent sets each with a telephone to communicate with incoming telephone calls from customers at customer sets of an external telephonic switching system and a visual display for displaying information concerning customers, the improvement being an automated preannouncement system, comprising:

means for storing preannouncement voice messages in different voices of a plurality of different agents respectively;

means for associating a sign-in code with a particular agent of the plurality of different agents;

means at the plurality of agent sets for entering in the sign-in code; and

means responsive to the sign-in code entered at any one of the plurality of agent sets to enable at least one of the preannouncement voice messages in the voice of the particular agent associated with the sign-in code to be automatically played to incoming telephone calls from customers which have been connected by the automatic call distributor to the one of the agent sets at which the sign-in code of the particular agent has been entered and in which the at least one of the preannouncement voice messages is played in the voice of the particular agent connected to the incoming telephone calls upon the initial connection of the incoming telephone calls by the automatic call distributor to the one of the agent sets.

2. The automatic call distributor of claim 1 in which the storing means includes means for digitally

storing the preannouncement voice messages in a hard
disk associated with a central processing unit in the
5 automated preannouncement system.

3. The automatic call distributor of claim 1
in which the storing means includes means for storing
different preannouncement voice messages respectively
associated with different types of customers.

4. The automatic call distributor of claim 3
including means for correlating different
preannouncement voice messages to different types of
customers upon receipt of an incoming call from a
5 different type of customer.

5. The automatic call distributor of claim 1
including means for correlating the preannouncement
voice messages with the information concerning the
customers that is displayable on the visual display of
5 the agent sets.

6. The automatic call distributor of claim 1
including means for automatically playing a
preannouncement voice message while the information
concerning customers is visually displayed at the visual
5 display of the one of the agent sets.

7. The automatic call distributor of claim 1
including means for storing multiple preannouncement
voice messages in a plurality of different agent voices.

8. The automatic call distributor of claim 7
in which the multiple voice messages storing means

includes means for storing a plurality of identically
worded message announcements in the different agent
5 voices.

9. The automatic call distributor of claim 8
including
means for selecting one of the preannouncement
voice messages in the voice of the particular agent
5 with the sign-in code of the particular agent, and
means for playing to one of the customer sets
connected through the automatic call distributor the one
of the preannouncement voice messages in the voice of
the particular agent associated with the sign-in code
10 entered at the one of the agent sets.

10. In an automatic call distributor having a
plurality of agent sets each with a telephone to
communicate with customers at customer telephone sets of
an external telephonic network, the improvement being an
5 automated preannouncement system, comprising:

a voice message memory accessible to the
plurality of agent sets;

means for internally storing in the voice
message memory a plurality of different preannouncement
10 voice messages respectively associated with different
types of customers each of the voice messages being
generated in a voice of each of a plurality of different
agents wherein the preannouncement voice messages are
played to the customer sets upon initial connection of
15 incoming calls to the agent sets;

means for determining the type of a customer
making an incoming call from one of the customer
telephone sets;

means for connecting an incoming call to one
20 selected agent of a plurality of different agents; and
means responsive to the determining means and
the connecting means for automatically selectively
playing to the customer and to the one selected agent
one of the plurality of different preannouncement voice
25 messages associated with the type of the customer making
the incoming call and in voice of the one selected agent
connected to the incoming call upon the initial
connection of the incoming call to the one agent.

11. The automatic call distributor of claim 10
in which the storing means includes means for digitally
storing the different preannouncement voice messages in
a hard disk associated with a central processing unit in
5 the automated preannouncement system.

12. The automatic call distributor system of
10 in which the automated preannouncement system
includes means for visually displaying on one of the
agent sets information concerning the customer calling
from one of the customer telephone sets while the one of
5 the plurality of different preannouncement voice
messages is played to the customer and to the one agent
upon initial connection of a incoming call to the agent.

13. The automatic call distributor of claim 10
in which the selectively playing means includes means
for associating a sign-in code with the particular
agent,

5 means at the agent sets for entering in the
sign-in code, and

means responsive to the sign-in code entered at the agent set to enable the preannouncement voice message in the voice of the particular agent associated with the sign-in code to be played to the customer connected through the automatic call distributor to the particular agent.

14. In an automatic call distributor having a plurality of agent sets each with a telephone to communicate with customers and a display screen to obtain visual information relating to customer types of said customers, the improvement being an automated preannouncement system, comprising:

means for automatically identifying one of a plurality of different customer types for each incoming telephone call, at least one of the plurality of different customer types associated with each of the customers;

means for storing a plurality of preannouncement voice messages and a plurality of associated display screen messages respectively associated with the plurality of different customer types, each of the preannouncement voice messages being in a voice of one of a plurality of different agents and are to be played prior to the one of the agents talking to the customer;

means for connecting an incoming telephone call from one of the customers to one of the agents; and

means responsive to the customer type identifying means and the incoming telephone call connecting means for automatically providing one of the plurality of display screen messages to the display

screen of one of the agent sets associated with the one
of the customer types connected thereto and for
automatically playing one of the plurality of
30 preannouncement voice messages to a customer associated
with the one of the customer types in the voice of the
one of the agents connected to the incoming telephone
call upon the incoming telephone call being initially
connected to the one agent set and prior to the one
35 agent talking to the customer.

15. The automatic call distributor of claim 14
in which the automated preannouncement system includes
means for digitally storing the voice messages in an
announcement memory associated with a central processing
5 unit in the automated preannouncement system.

16. The automatic call distributor of claim 14
in which the automated preannouncement system includes
means for automatically playing one of the plurality of
voice messages in response to the one message associated
5 with the type of customer visually appearing on the
display screen of an agent set.

17. The automatic call distributor of claim 14
in which the selectively playing means includes
means for associating a sign-in code with a
particular agent,
5 means at the agent sets for entering in the
sign-in code, and
means responsive to the sign-in code entered
at one of the plurality of agent sets to enable one of
the plurality of voice messages which is in the voice of
10 the particular agent associated with the sign-in code to

be played to the customer connected by the automatic call distributor with the particular agent at the one of the agent sets.

18. In an automatic call distributor having a plurality of agent sets each with a telephone to communicate with incoming telephone calls from customers at customer sets of an external telephonic switching system and a visual display for displaying information concerning customers, the improvement being a method of automatically playing preannouncement messages, comprising the steps of:

storing preannouncement voice messages in different voices of a plurality of different agents respectively;

associating a sign-in code with a particular agent of the plurality of different agents;

entering the sign-in code at one of the plurality of agent sets; and

automatically playing one of the preannouncement voice messages in the voice of the particular agent associated with a sign-in code entered at any one of the plurality of agent sets to incoming telephone calls from the customers which have been connected by the automatic call distributor to one of the agent sets at which the sign-in code of the particular agent has been entered and in which the one of the preannouncement voice messages played in the voice of the particular agent is played upon an initial connection of an incoming telephone call by the automatic call distributor to the one agent set at which the sign-in code is entered.

19. The method of claim 18 in which the step of storing preannouncement voice messages includes the step of digitally storing the preannouncement voice messages in a hard disk associated with a central processing unit of the automatic call distributor.

20. The method of claim 18 in which the step of storing preannouncement voice messages includes the step of storing different preannouncement voice messages respectively associated with different types of customers.

21. The method of claim 20 including the step of correlating different preannouncement voice messages to different types of customers upon receipt of an incoming call from a different type of customer.

22. The method of claim 18 including the step of correlating the preannouncement voice messages with the information concerning each of the customers that is displayable on the visual displays of the plurality of agent sets.

23. The method of claim 18 including the step of automatically playing the one of the preannouncement voice messages while the information concerning customers is visually displayed at the visual display of the one of the agent sets in which the sign-in code is entered.

24. The method of claim 18 including the step of storing multiple preannouncement voice messages in a

plurality of different voices corresponding to a plurality of different agents.

25. The method of claim 24 in which the step of storing multiple voice messages includes the step of storing a plurality of identically worded message announcements in the plurality of different voices
5 corresponding to the different agents.

26. The method of claim 25 including the steps of

selecting one of the preannouncement voice messages in the voice of the particular agent, and
5 playing to one of the customer sets connected to an agent set through the automatic call distributor the one of the preannouncement voice messages in the voice of the particular agent associated with the sign-in code at the agent set.

27. In an automatic call distributor having a memory accessible to a plurality of agent sets each with a telephone to communicate with incoming telephone calls from customers at customer telephone sets of an external
5 telephone network, the improvement being a method of automatically playing preannouncement messages, comprising the steps of:

internally storing in the memory accessible to the agent sets a plurality of different preannouncement
10 voice messages respectively associated with different types of customers, each of the preannouncement voice messages being in a voice of an agent is played to customer telephone sets upon initial connection of incoming calls to an agent set;

15 determining a type of a customer placing an
incoming call from one of the plurality of customer
telephone sets;
 determining which one of a plurality of
different agents will answer the incoming call; and
20 selectively playing to the customer calling
from one of the plurality of customer telephone sets,
one of the plurality of different preannouncement voice
messages associated with the type of customer in the
voice of the agent who is determined to answer the
25 incoming call upon the initial connection of the
incoming call to the agent prior to the agent talking to
the customer.

 28. The method of claim 27 in which the step
of internally storing different preannouncement voice
messages includes the step of digitally storing the
different preannouncement voice messages at a hard disk
5 associated with a central processing unit in the
automatic call distributor.

 29. The method of claim 27 including the step
of visually displaying on the agent set information
concerning the customer making the incoming call while
the preannouncement message is played to the customer.

 30. The method of claim 27 in which the step
of selectively playing includes the steps of
 associating a sign-in code with a particular
agent,
5 entering in the sign-in code at one of the
agent sets, and

playing one of the preannouncement voice
messages in the voice of the particular agent associated
with the sign-in code entered at the one of the agent
10 sets to the customer connected through the automatic
call distributor to the particular agent.

31. In an automatic call distributor having a
plurality of agent sets each with a telephone to
communicate with incoming telephone calls from customers
and a display screen to obtain visual information
5 relating to customer types, a plurality of agents being
associated with the plurality of agent sets to answer
the incoming telephone calls, the improvement being a
method of automatically playing preannouncement
messages, comprising the steps of:

10 identifying one of a plurality of different
customer types of an incoming call;

storing a plurality of preannouncement voice
messages and corresponding display screen messages
respectively associated with the plurality of different
15 customer types, each of the preannouncement voice
messages being in a voice of one of the plurality of
agents respectively;

connecting an incoming call from one of the
customers to one of the agents;

20 automatically providing one of the plurality
of display screen messages to the display screen of one
of the agent sets associated with the type of customer
connected thereto, the one of the agent sets having a
particular one of the agents associated therewith; and

25 playing to the customer connected to the one
of the agent sets one of the plurality of
preannouncement voice messages which corresponds to the

30 one of the plurality of display screen messages being provided to the agent set which is associated with the identified customer type of customer connected thereto and which is in the voice of the particular one of the agents upon the incoming call being initially connected to the one agent set and prior to the one agent talking to the customer.

32. The method of claim 31 including the step of correlating and playing a different voice message to a different type of customer upon receipt of an input from a different type of customer.

33. The method of claim 31 including the step of digitally storing the voice messages in a hard disk associated with a central processing unit in the automated preannouncement system.

5 34. The method of claim 31 including the step of automatically playing one of the plurality of voice messages in response to the one voice message associated with the type of customer visually appearing on the display screen of an agent set.

35. The method of claim 31 further includes the steps of
associating a sign-in code with the particular agent;
5 entering in the sign-in code at one of the agent sets, and
playing the voice message in the voice of the particular agent associated with the sign-in code entered at the one of the agent sets to the customer

10 connected by the automatic call distributor with the
particular agent at the one of the agent sets.

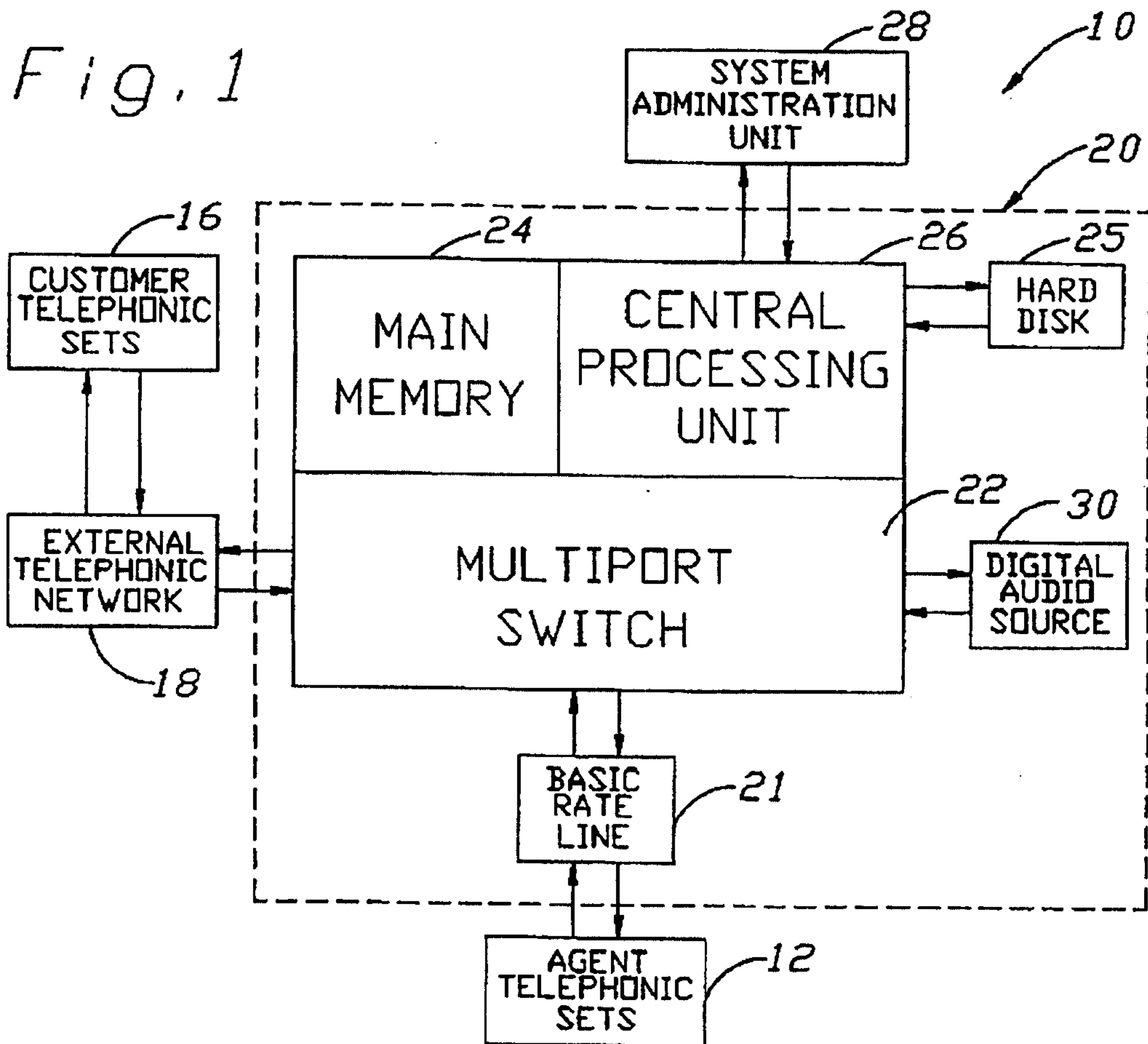


Fig. 2

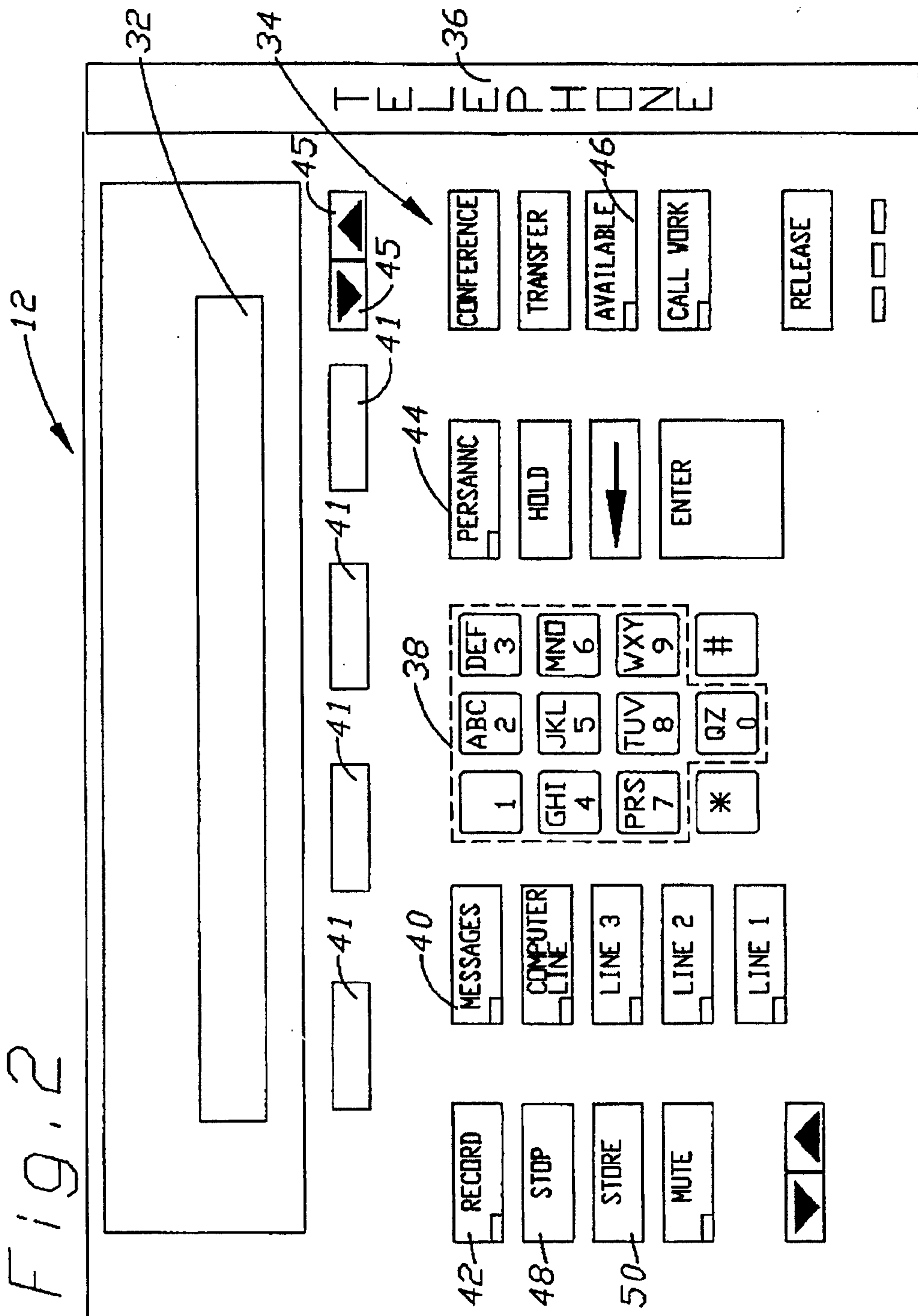


Fig. 3A

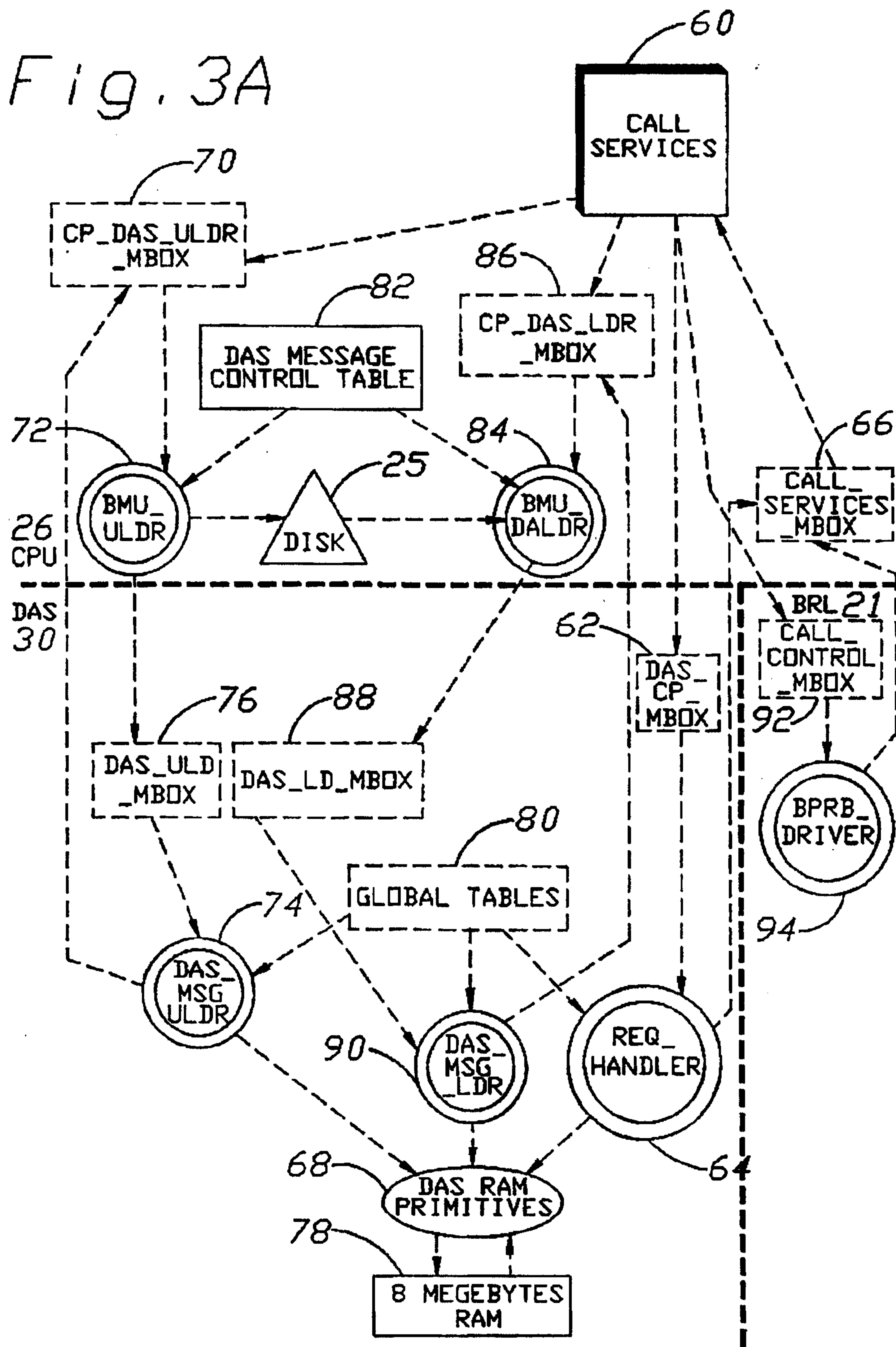


Fig. 3B

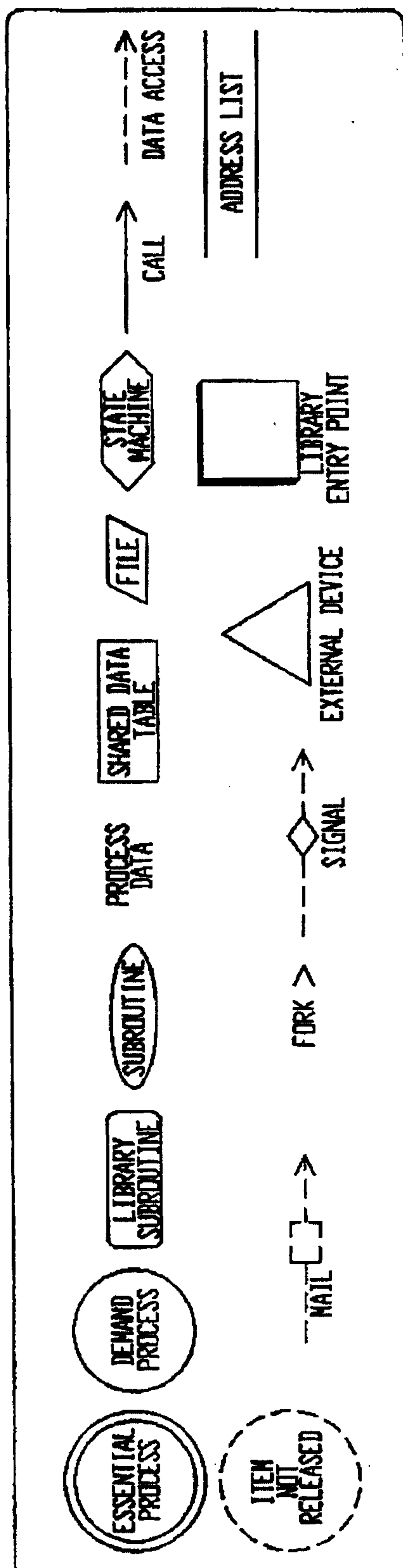


Fig. 4A

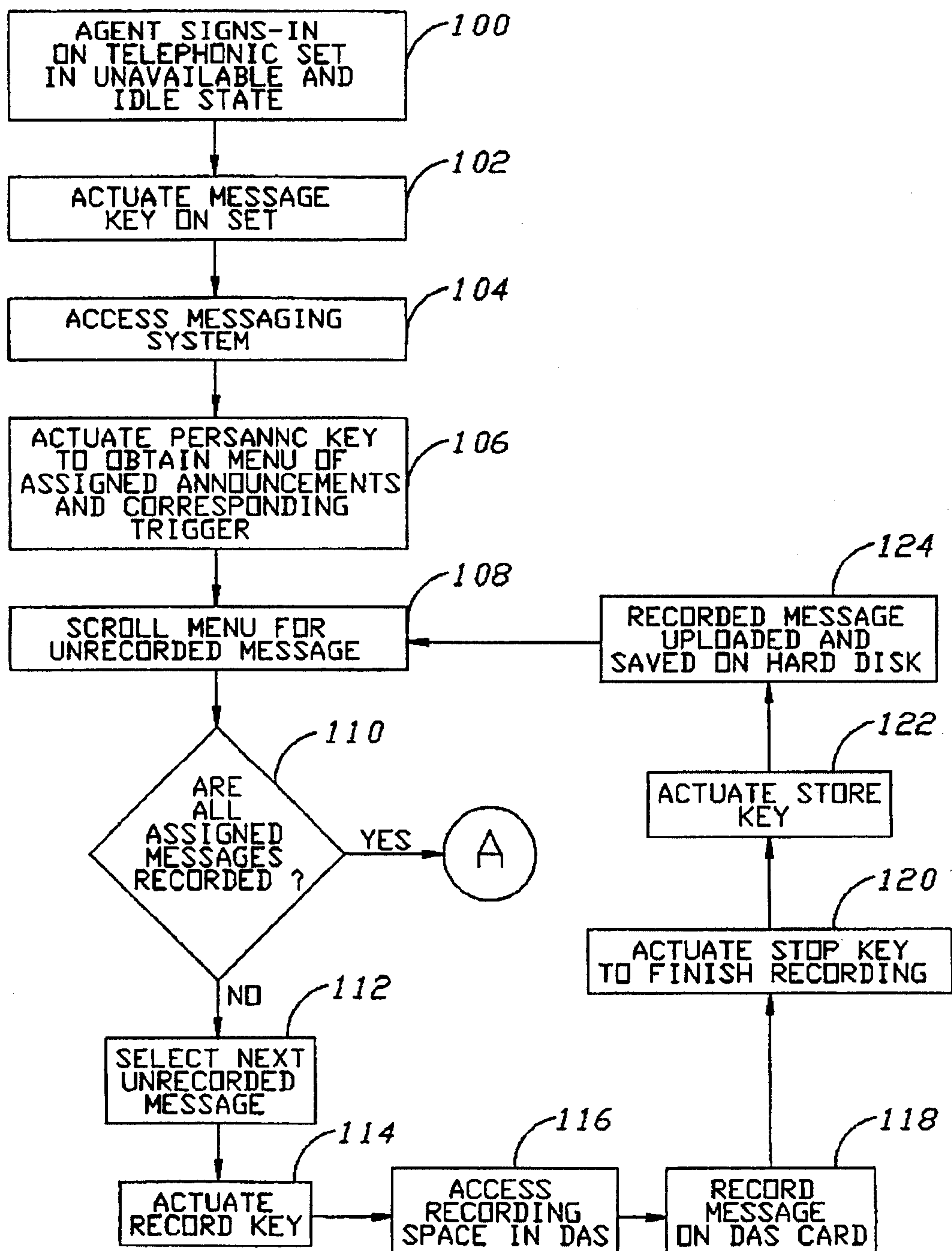


Fig. 4B

