SEMINAR NO. JA3080

TRANSACTION FACILITY ANALYSIS
(TAF)

STUDENT HANDOUT

PROPRIETARY NOTICE

REVISION RECORD	
REVISION	**DESCRIPTION**
A	Manual Release.
(07-01-83)	
Publication No. JA3080-1	

REVISION LETTERS I, O, Q AND X ARE NOT USED

TRANSACTION FACILITY ANALYSIS

Address comments concerning this manual to:

CONTROL DATA CORPORATION
PS CYBER Training
ETCSEM
1450 Energy Park Dr.
St. Paul, MN 55108

or use Comment Sheet in the back of this manual.

COURSE DESCRIPTION

COURSE TITLE: TAF Analysis

COURSE LENGTH: 5 Days (30 Hours)

DESCRIPTION: This class is a lecture course in which the student will receive classroom instruction on TAF internals. The primary emphasis will be on learning the flow of TAF and how all of its tables relate to the processing of transactions. All three data managers that TAF supports will be discussed. Installation parameters will be discussed and how their values affect the operation and throughput of TAF.

PREREQUISITES: The student should have a basic understanding of what transaction processing is, as well as the internals of the NOS operating system and TAF from a user's perspective. Some data management and networks background would be helpful.

OBJECTIVES: Upon successful completion of this course, the student will be able to appropriately set up all the installation parameters to suit a given application and create the TAF files necessary for TAF to run. By understanding the general flow of the transaction through TAF and the interrelationship between all of TAF's tables, the student will be able to analyze any TAF problems that occur. The student will also become aware of factors that affect performance and possibly how to change them to improve performance.

I. TAF/EMS-NAM PROCESSING OVERVIEW

II. TAF MEMORY MAP

III. FLOW OF TRANSACTION THROUGH TAF

IV. TASK CONTROL/SUBCONTROL POINTS

V. KEY TABLES

VI. INTERNAL TRACING

VII. STATISTICS & INSTALL PARAMS

Ems site TAF ANALysis

COURSE OUTLINE

I. TAF Installation and Terminology

- A. Installation of TAF
- B. Glossary

II. Overview of TAF

- A. Parts of TAF
- B. Input from Terminal to TAF

III. Overview of TAF Processing

- A. TAF Terminal Input Flow
- B. Subcontrol Point Theory
- C. Subcontrol Point Structure

IV. TAF Coding Conventions

V. Structure of TAF

VI. Initialization

- A. TAF Procedure File
- B. TAFREC
  1. TAF Initilization File (COMKTIF)
  2. Communication Recovery File (COMKTRF)
- C. TAF1
- D. Preset - PRE

VII. Tables

- A. Communication Block (COMKCBD)
  1. System Header
  2. TAF Queue Words
  3. User Header
- B. Task System Area (COMKTSA)

C. Subcontrol Point Table (COMKSCD)

D. Requested Task List (RTL)

E. Active Transaction List (ATL)

F. Task Load Stack (TLS)

G. Task Library Directory (COMKTLD)

H. Transaction Directory (COMKTLD)

I. Terminal Status Table (COMKTST)

J. Communication Recovery File Table (COMKTRF)

K. NAM Communications Table (COMKNWC)

L. Task Rollout Table (ROLT)

M. Internal Task Trace Packet (ITTP)

N. Element Descriptor Table (EDT)

O. Low Core Pointers (V-Symbols)

VIII. Flow of a Transaction through TAF

A. Transaction Flow Thru Tables

1. Inital Transaction Input

2. Recovery Situation

IX. Flow of Processing within TAF

A. TMDC - Time Dependent Control

B. TSSC - Time Slice Subcontrol Points

C. TAF Work Queuing

1. Initial Transaction Input

2. Communication Recovery Requests

3. TAF/CRM Batch Concurrency Requests

COURSE OUTLINE (Continued)

X. RA+1 Request Processing

A. SRTN - Subcontrol Point Return

B. RA+1 Requests

XI. Task Scheduling

A. TAF Scheduling of Tasks

1. Initial Transaction Input
2. Task Requests
3. Task Cease
4. System Origin Transactions

B. SCHD - Scheduler routine

XII. Task Rollout/Rollin Processing

A. TAF Rollout of Tasks

1. WAITINP Request
2. Terminal Output Limit
3. CALLRTN Request
4. WAIT Request
5. Task Memory Request

B. ROLL - Rollout routine

XIII. TAF Memory Management

A. Processes and Routines

1. Request Subcontrol Point Memory
2. CRM/CMM Memory Requests
3. Task Memory Requests
4. Reduction of Field Length

B. Detection of Potential Blocked Tasks

C. Examples

XIV. TAF Termination

- A. Process and Routine
- B. TAF2

XV. TAF/CRM

- A. Initialization
- B. Structure of Tables and Recovery Files (COMKCRM)
  1. Data Base/Recovery File Table (TDRF)
  2. After Image Recovery File Table (TARF)
  3. Before Image Recovery File Table (TBRF)
  4. Logical Name Table (TLNT)
  5. File Control Table (TFCB)
  6. Lock Table (TKOK)
  7. Transaction Sequence Table (TSEQ)
  8. After Image Recovery File
  9. Before Image Recovery File
- C. TAF/CRM Memory Map
- D. TAF/CRM Processing
  1. Input/Output Queue Entries
  2. CRM RA+1 Request Processing
  3. Flow of Processing
  4. CRM Deferred Logging Exit
- E. Batch Concurrency
  1. Tables
  2. Flow of Processing
- F. Data Base Recovery
  1. Automatic Component Processes

COURSE OUTLINE (Continued)

2. Batch Component Processes
3. Data Base Status
4. Recovery Considerations

XVI. TAF/CDCS Interface

A. Processing within TAF
B. SSC RA+1 Request
C. Recovery Aspects

XVII. TAF/TOTAL Interface

A. Initialization
B. Processing within TAF

XVIII. TAF/NAM Interface

A. TAF/NAM Communication
B. V - Symbols and Block Headers
C. AIP Calls
D. Major Processes

XIX. Recovery Revisited

A. Communication and Data Base Recovery
B. Recovery Situations

XX. TAF Installation Parameters and Tuning

A. Installation Parameters
B. Tuning Guidelines

XXI. Analysis of TAF Problems

XXII. Exercises

TAF INSTALLATION AND TERMINOLOGY

TAF INSTALLATION

- Install TAF and Related Binaries in System
- TAF Subsystem Related Files
  - TCF
  - NCTFid
  - ZZCRFid
  - JOUR0
  - TASKLIB
- Application Related Files
  - xxJ
  - ZZxxA0i/ZZxxB0i
  - Data Base Files
  - xxJORn
  - xxTASKL

SYSTEM FILES

TCF	TAF Configuration File. It contains the names of the data managers and data bases to be initialized.
TASKLIB	System Task Library. This file contains the tasks that are common to all applications under TAF and is defined under the Transaction Subsystem User Number.
xxTASKL	User-specified task library for application xx.
xxJ	This file identifies the journal files for a specific application [xx] and provides the user index and user number of the data base. It also describes the data base files for this application.
JOUR0	System Journal File. The transaction subsystem writes journal entries for all transactions to this file.
xxJORn	Optional journal files for each application [xx] for task journalizing of transactions.
TAFPRC	Procedure file containing control statements for TAF.
NCTFid	Network Communication file(s). They contain a list of users and applications in which they should be validated for.
ZZCRFid ZZxxA0i ZZxxB0i	Communication and data base recovery files.

REQUIRED SYSTEM TASKS

- BTASK
  - Recovers runnable BTRAN transactions
  - Reads CRF
  - Calls CTASK to rerun transaction
- CTASK
  - Data base recovery for
    - . Interactive users
    - . BTRAN users
    - . Recovery mode preset
    - . Terminal failures
- ITASK
  - Initial terminal input
- KDIS
  - K. SWITCH
- MSABT
  - Error messages to originating terminal

REQUIRED SYSTEM TASKS (Continued)

- OFFTASK
  - Request for an inactive task
- RCTASK
  - Recovers rerunnable CDCS transactions
- RTASK
  - Recovers rerunnable transactions after
    - . Terminal disconnects
    - . Network failures
    - . TAF failures
    - . System failures
- SYSMSG
  - K. MESSAGE
  - Task originated K-display command diagnostic

GLOSSARY

GENERAL NAMES

AAMI--The Cyber Record Manager interface routine that handles the data base concurrency.

AIP--Network 'Application Interface Package'. It contains those object time routines that allow communication with the network.

APPLICATION--Under TAF, the logical entity made up of the following files; xxJ, xxTASKL, data base files and recovery files. An application is designated by a two character identifier.

CDCS--Cyber Data Base Control System. A Control Data common product data management system.

CMM--Common Memory Manager. A memory manager which the Cyber Record Manager uses to manage a part of TAF's field length. Tasks may also use a copy of CMM to manage their own field lengths.

CRM--Cyber Record Manager. A Control Data common product. It is used by TAF as part of the TAF/CRM Data Manager to perform the physical I/O to mass storage.

DMREC--The TAF/CRM Batch Recovery utility program. It will perform recovery of data base files in an off-line situation and maintain a directory of data base and after image dump tapes.

FDL--Fast Dynamic Loader. It consists of routines that CRM calls to load its capsules in the dynamic area of the CMM buffer. It is part of the Cyber Loader, a common product.

KTSDMP--The TAF utility program that will process and format for output the field length dump of a TAF task.

KTSROLL--The local file that TAF ' rolls' tasks to. The task's entire field length as well as the Task System Area is written to this file during rollout.

LIBTASK--The TAF utility used for creating or maintaining a particular task library.

TAF--Control Data's 'Transaction Facility'. It is a subsytem under NOS that allows efficient on-line transaction processing. It consists of an absolute compass program and additional overlays (called the transacting executive). During initialization it will load a particular data manager (TAF/CRM, TOTAL) within its field length or initiate communication with one (CDCS).

TAF/CRM--A data manager that is loaded in TAF's field length during initialization. It consists of the following pieces of code: AAMI, CRM, CMM, and FDL.

TASK--Usually a Cobol or FTN program that uses the TAF object time routines for file and terminal I/O and executes under control of the transaction executive.

TOTAL--A data manager developed by CINCOM. It is supported as one of the data managers that the transaction executive will interface to.

TRANSACTION--A task, or a chain of tasks initiated via CALLTRN, CALLRTN, or CALLTSK with CEASE.

TROB--The local file that TAF will write a portion of its field length to when there is no activity. TAF is said to be 'idle' during this time when its field length is reduced.

TRANSACTION EXECUTIVE TABLES NAMES/ABBREVIATIONS

ATL--ACTIVE TRANSACTION LIST (TAF)
The active transaction list (ATL) correlates the RTL entries to the communication blocks that are waiting for those tasks. When a task from the RTL gets loaded, TAF chains through the ATL's to find all the CB's queued on that task. There can be a maximum of QL (queue limit defined for the task) ATL entries queued on one RTL.

CB--COMMUNICATION BLOCK (COMKCBD)
The communication block's (CB) main purpose is to function as an input file for TAF. As soon as TA F receives input from a terminal, it puts the input in a CB where it will remain until transaction completion or the task does a wait for terminal input. A transaction can consist of multiple tasks with the communication block being the means to pass input and data from one task to the next. There are three parts to the CB--the system header, user header, and input from the terminal. The system header contains control information strictly for TAF's usage. The user header is accessable by the task, but cannot be changed except by TAF. The rest (66 words) is available for usage by the tasks. The last word contains the packed date and time. Initially, the input from the terminal gets stored there and passed to the first task. That task then has the option of passing the input on to the next task (if there is one) as is or changing it. TAF always copies that part of the CB from the task back into TAF's field length and back out to the new task.

EDT--ELEMENT DESCRIPTOR TABLE (TAF)
The EDT contains information about journal files and task libraries associated with a data base. It is used by the executive when searching for a task to load from a particular library.

ITTP--INTERNAL TASK TRACE PACKET (TAF)
The internal task trace packet (ITTP) is contained in a circular buffer (PBUF). Each entry is four words long and contains information about the task that has just issued an RA+1 request. This trace information is a valuable tool when analyzing any TAF problems that may occur.

NCT--NAM COMMUNICATION TABLE (COMKNWC)
The NCT contains detailed information about the status of each terminal while it is logged in. The Application Connection Number (ACN) is used as an offset into this table.

OEP--OVERLAY ENTRY POINT NAME LIST
It contains the information necessary for loading any of the transaction executive's overlays. The information is generated by the ENTRY macro.

OREL--OVERLAY RELOCATION LIST
This list contains the pointer to the location of instructions and the number of instructions to relocate after loading this overlay. This information is kept at the end of each overlay and is generated by the ENDOVL macro.

RLAT--ROLLOUT FILE ALLOCATION MAP
A map of the unused space on the task rollout file, KTSROLL. Each bit in this map designates one block on the rollout file. Each block is ROLBL words in length.

ROLT--ROLLOUT TABLE (TAF)
The rollout table is only used when a request a task has made cannot be satisfied immediately. Ideally a task would never roll out since transactions are typically very short in duration. If TAF decides to roll a task out, it creates a rollout table entry, but in most cases will not roll the task out immediately. The delay is added to increase the chance of the request that caused the rollout to complete. Even once the rollout is initiated, if the request completes, TAF stops the rollout in progress. It just zeros out the ROLT entry and forgets it ever started the rollout. As long as the SUBCP table is still intact, the task can be restarted.

RTL--REQUESTED TASK LIST (TAF)
The requested task list (RTL) contains all tasks waiting to be scheduled, like the NOS input and rollout queue. Everytime the scheduler gets called, it evaluates all tasks in the RTL and finds the task with the highest queue priority and smallest field length. If sufficient field length can be obtained to load the task, the requested task gets moved from the RTL to the task load stack.

SUBCONTROL POINT (SUBCP) TABLE
The subcontrol point is to TAF what control points are to NOS. A task executes at a Subcp is guaranteed memory protection and system security. The executive (TAF) processes all RA+1 requests which gives TAF complete control over all tasks. TAF can load whatever task it wants to and do any storage moves within its field length necessary to make room for that task. TAF can also request additional field length from the system in order to load a task. TAF does not have the restriction that the field length associated with a Subcp has to be in the same order as the Subcp's themselves, i.e., Subcp 3's FL can be after Subcp 5's FL.

By examining the Subcp, the analyst can determine which communication block (CB) a task is executing with and which ones are waiting to execute.

TIF--TAF INITIALIZATION FILE (COMKTIF)
This local file is created by TAFREC to pass recovery information and the terminal status table to TAF1.

TLD--TASK LIBRARY DIRECTORY (COMKTLD)
The task library directory (TLD) contains all control information for each task. TAF attaches all task libraries and reads the directories into its field length, leaving space for TLDL (10 is default) extra tasks to be added during production. Each time a task is requested, the TLD associated with the data base for the terminal is searched to find the called task. If the task is not found, TAF searches the system TLD. If the task is in neither of those TLD's, the transaction is aborted. From the TLD entry, TAF knows whether to load the task from disk or EM and at what position to initiate the read.

TLS--TASK LOAD STACK (TAF)
If the task selected from the RTL needs to be loaded (i.e., not in core and reusable), TAF tries to find the field length required and a free Subcp table entry. If Subcp table entry and field length are available, TAF moves all CB's linked through the ATL's to that task (RTL) into the Subcp table entry. TAF then generates an entry in the TLS to load the task. It is from this table that TAF actually does the I/O to load a task from either disk or EM.

TRD--TRANSACTION DIRECTORY (COMKTLD)
This directory is created and maintained by LIBTASK. It is read into TAF's field length during initialization. It contains information relative to a transaction like the name, the TLD offsets for the tasks that make up the transaction and which data manager this transaction uses.

TSA--TASK SYSTEM AREA (COMKTSA)
The task system area (TSA) is like the control point area in NOS. It contains the task's exchange package and system control information. Since it is located in the task's negative field length, the executive is the only one that can read and write it.

TSB--TAF STORAGE BUFFER (TAF)
A buffer made up of four word entries that any processor in the executive can reserve to store information. It is most often used for queuing work for the executive during processing of a TAF/CRM batch concurrency request.

TST--TERMINAL STATUS TABLE (COMKTST)
The terminal status table (TST) is built from the NCTFid file. Through it TAF keeps track of what users are logged in so only one user can be logged in with the same name. The TST's pri-

mary function is to keep track of what each user is doing and what kind of privileges it has (user argument area).

TTFT--TERMINAL FILE TABLE (COMKTST)
The transaction executive table that associates terminals in the Terminal Status Table with the correct Communication Recovery File (CRF). This table is used to find the correct CRF from which to do I/O for recovery requests.

TTRF--COMMUNICATION RECOVERY FILE TABLE (COMKTRF)
This table defines the attributes of the Communication Recovery File, and contains the FET from which I/O is done on the file. Attributes include maximum size and number of user (RPUT/RGET) messages for this file.

TAF/CRM TABLE NAMES/ABBREVIATIONS

BCT--BATCH COMMUNICATION TABLE (COMKBRD)
This table contains the information necessary for communication with a batch job, and the batch concurrency parameters passed from the batch job for TAF/CRM request processing. The number of entries in this table is based on the installation parameter, TBCON.

TARF--AFTER IMAGE RECOVERY FILE TABLE (COMKCRM)
This table contains controlling information for After Image Recovery file (ARF) processing. The FET as well as an image of the ARF header is part of this table.

TBRF--BEFORE IMAGE RECOVERY FILE TABLE (COMKCRM)
This table contains controlling information for Before Image Recovery file (BRF) processing. The FET as well as an image of the BRF header is part of this table.

TDRF--DATA BASE AND RECOVERY FILE TABLE (COMKCRM)
This table contains controlling information for data bases and links to the recovery files. Statistical information about the data base as well as interface information for the batch recovery utility, DMREC, is also kept in this table.

TFCB--FILE CONTROL TABLE (COMKCRM)
This table holds the current TAF/CRM request and the FIT for the data base file. There is one entry reserved for each active user of the file.

TKOK--LOCK TABLE (COMKCRM)
This table contains the controlling information thru which file and record locks on data base files are granted and released.

TLNT--LOGICAL NAME TABLE
This table contains controlling and statistical information for a data base file. It contains all the alternate key descriptors for a multiple index file.

TSEQ--TRANSACTION SEQUENCE TABLE
This table tracks and controls access to the TAF/CRM data manager. Through this table resource information necessary for a transaction's use of TAF/CRM is kept track of.

NOS CONSOLE

X. DIS.

SUI=16

ATTACH, TAFDOLD.

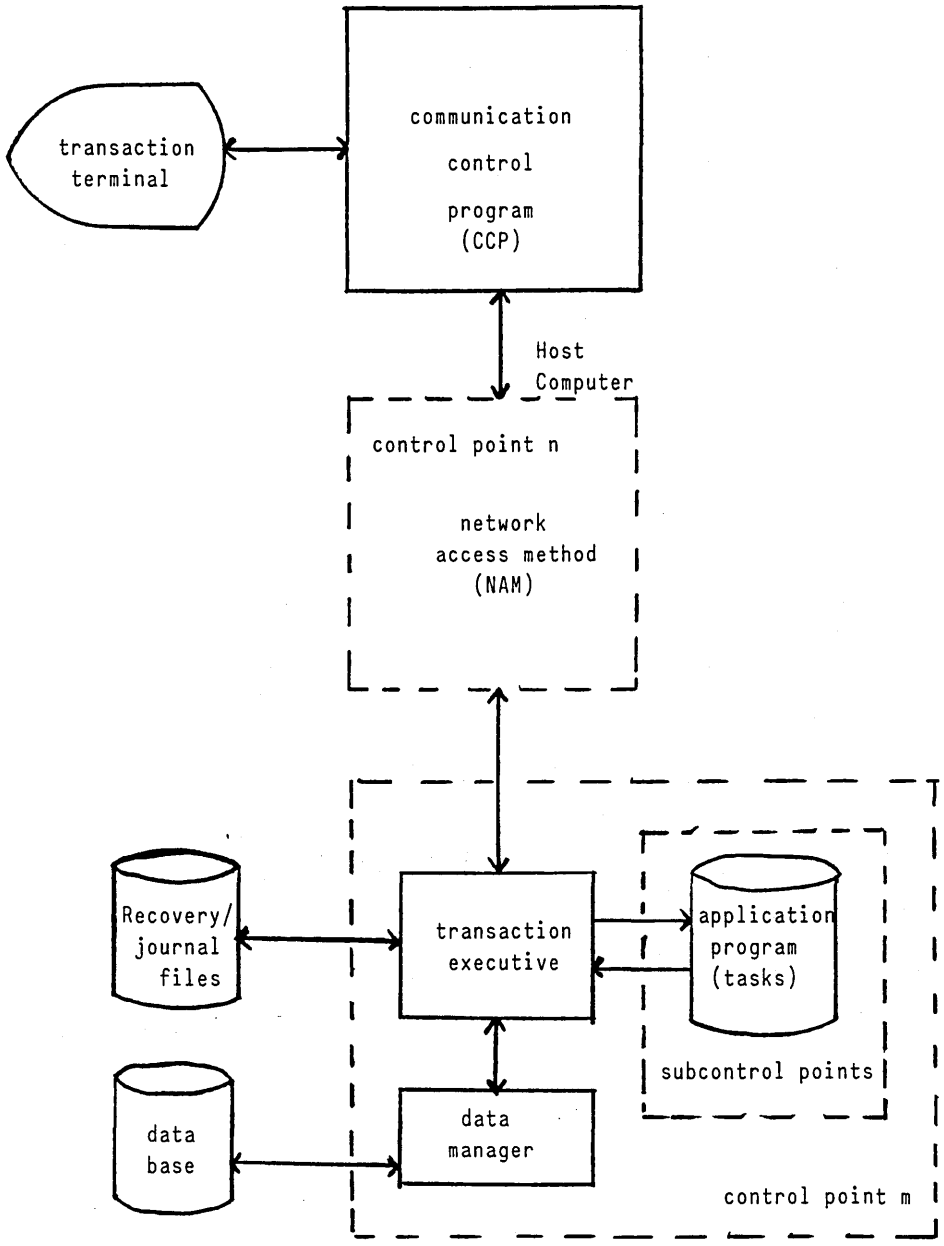
COPYCI, TAFDOLD.

ROUTE, OUTPUT, DC=PR, REP=6

DROP.

OVERVIEW OF TAF

TRANSACTION PROCESSING USING TAF WITHIN NOS

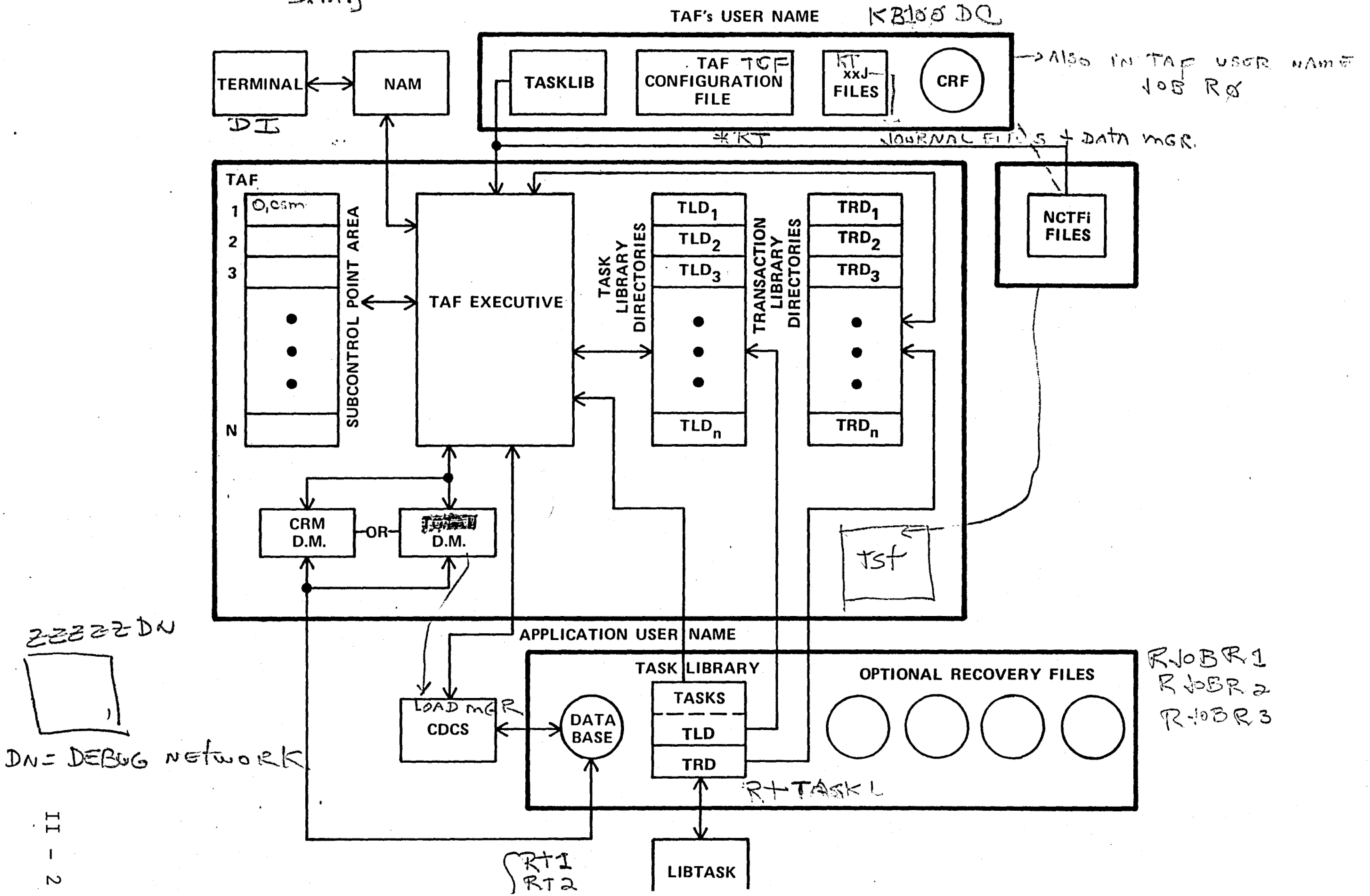


TCF → "RT" IS DATA BASE ID

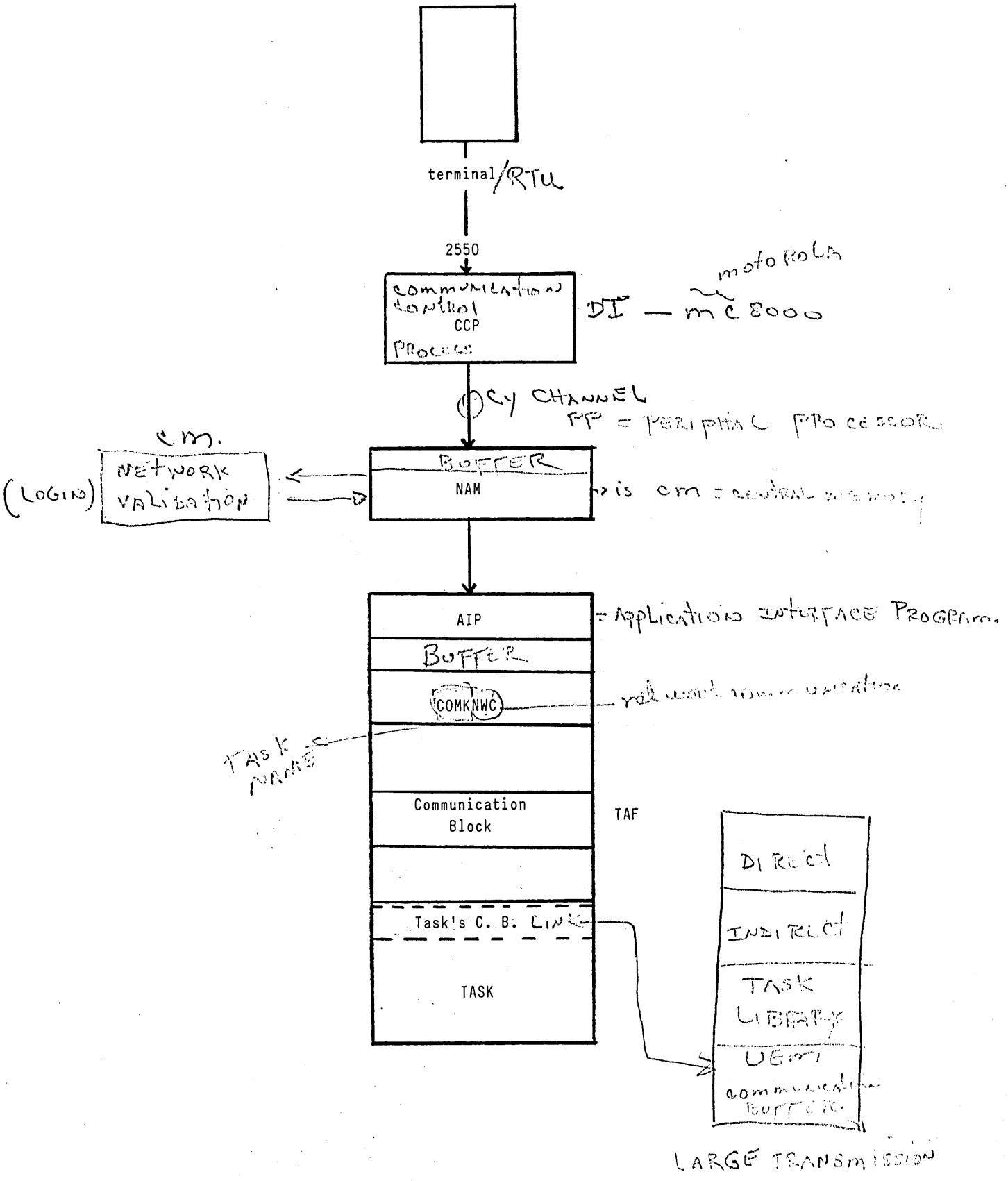
RTJ → CRM IS DATABASE MGR. (CDC D.M.)

CRM = AAMI

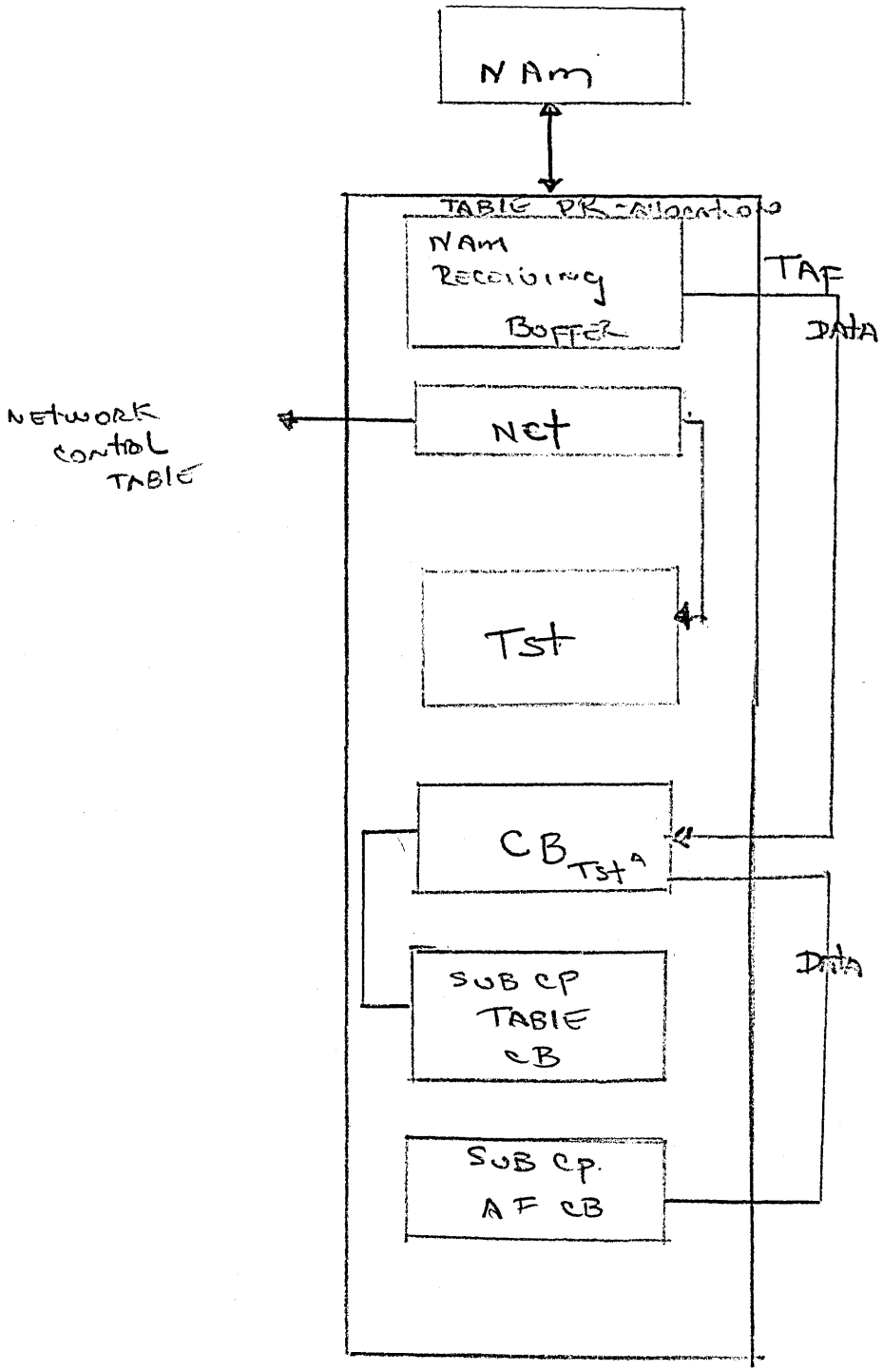
AAMI = ADVANCED ACCESS METHODS INTERFACE



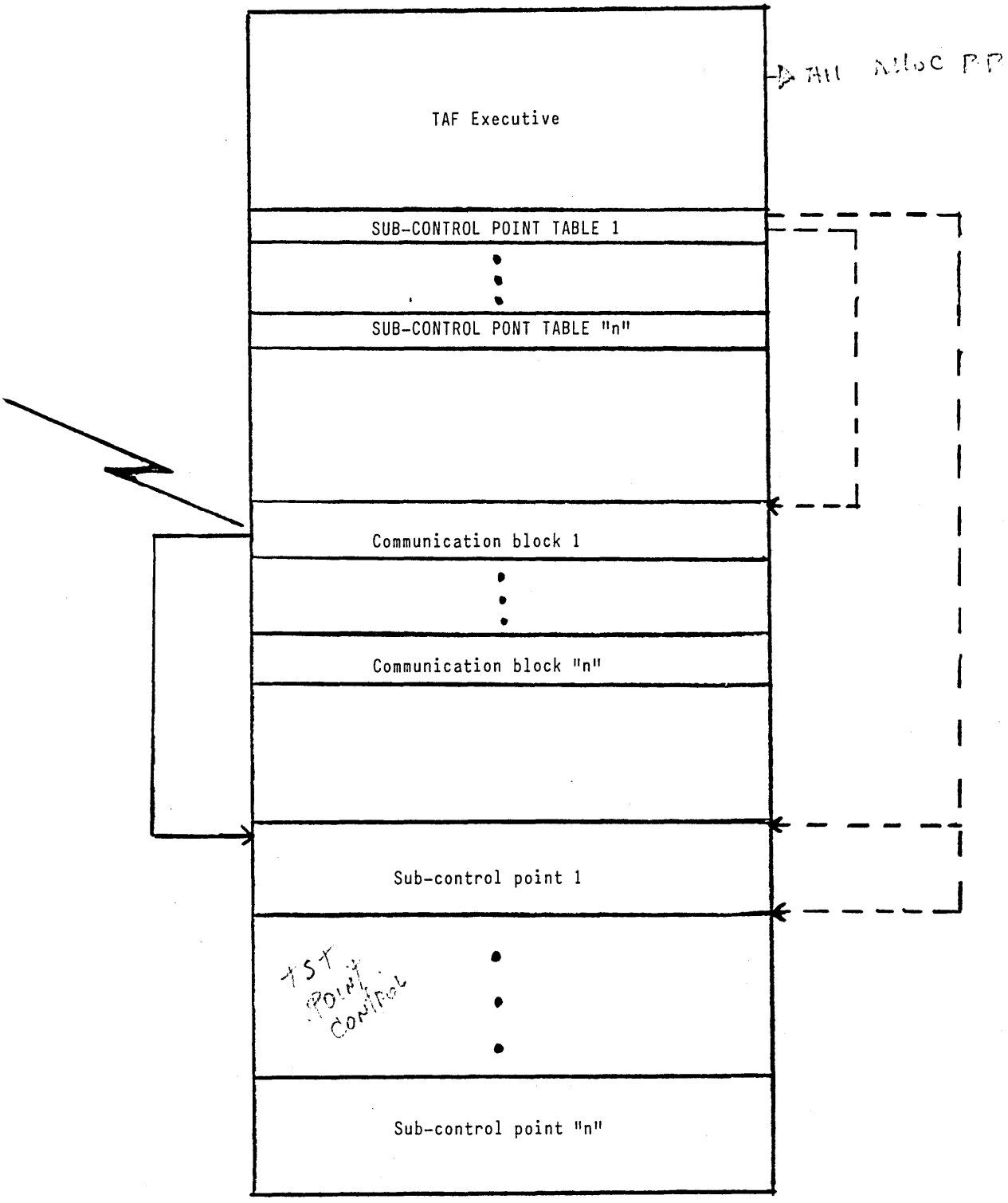
INPUT FROM TERMINAL TO TAF



OVERVIEW OF TAF PROCESSING


NAM
TABLE PRE-ALLOCATION
NAM Receiving Buffer
TAF
DATA
NETWORK CONTROL TABLE
Net
Tst
CB Tst A
SUB CP TABLE CB
SUB CP. AF CB
DATA

TAF TERMINAL INPUT FLOW



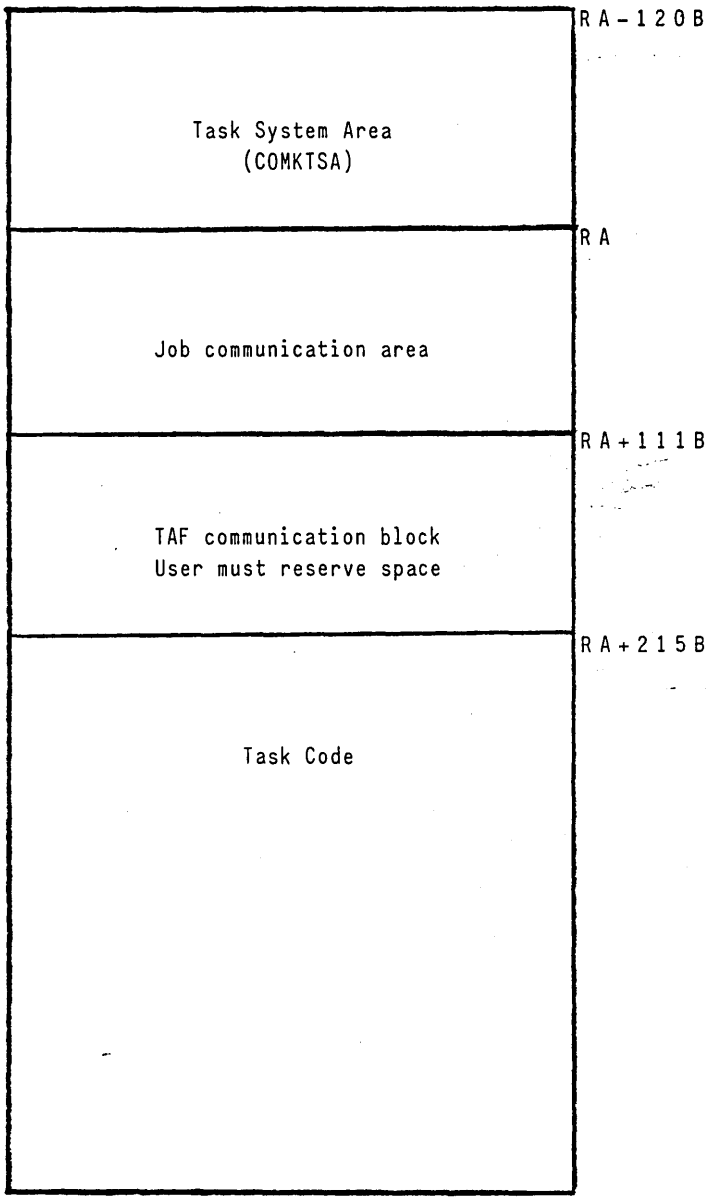
SUBCONTROL POINTS

- Division of a CM control point
- An executive at the control point must control the subcontrol point
- SUBCP. - IPRDECK entry

XJP Cpointr

Request for SLB cp

SUBCONTROL POINT STRUCTURE



TAF CODING CONVENTIONS

CODING CONVENTIONS

- FIELD Macro - Defines fields within words.

  XXXX FIELD word, left bit, right bit

  Where

  XXXXW = word location of field

  XXXXN = number of bits in the field

  XXXXS = left most bit in the field

  Field Manipulation Macros

  PUTFLD Y,RN,TAG

  GETFLD Y,RN,TAG

- Txx Opdef - Saves a 'level of indirection' in referencing TAF tables. The 'K' portion of the instruction is set during TAF's initialization.

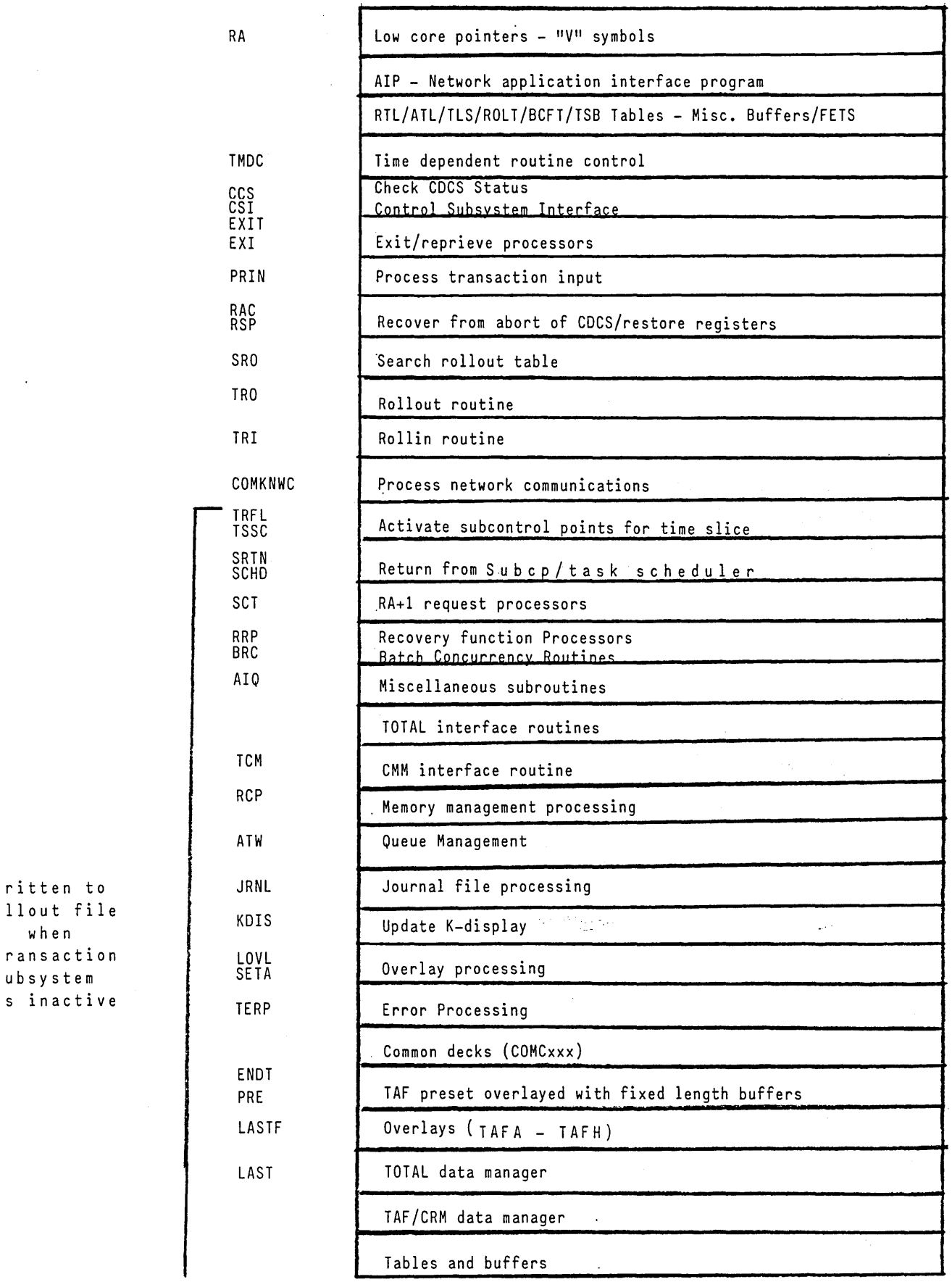
  TA1 B5,VTST will become

  SA1 B5+K where K=FWA of TST

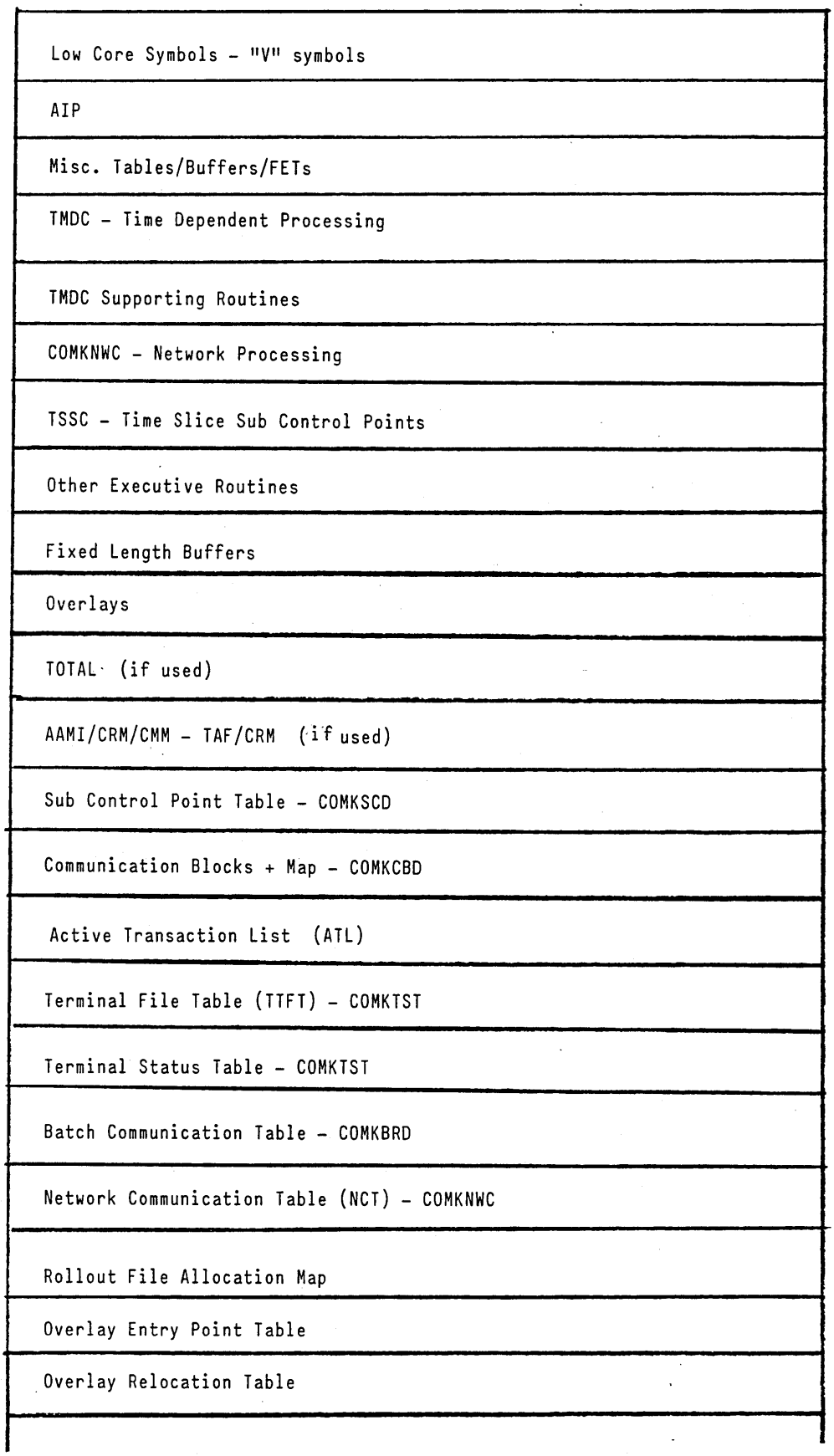
- Registers B2 and B7 are sacred

STRUCTURE OF TAF

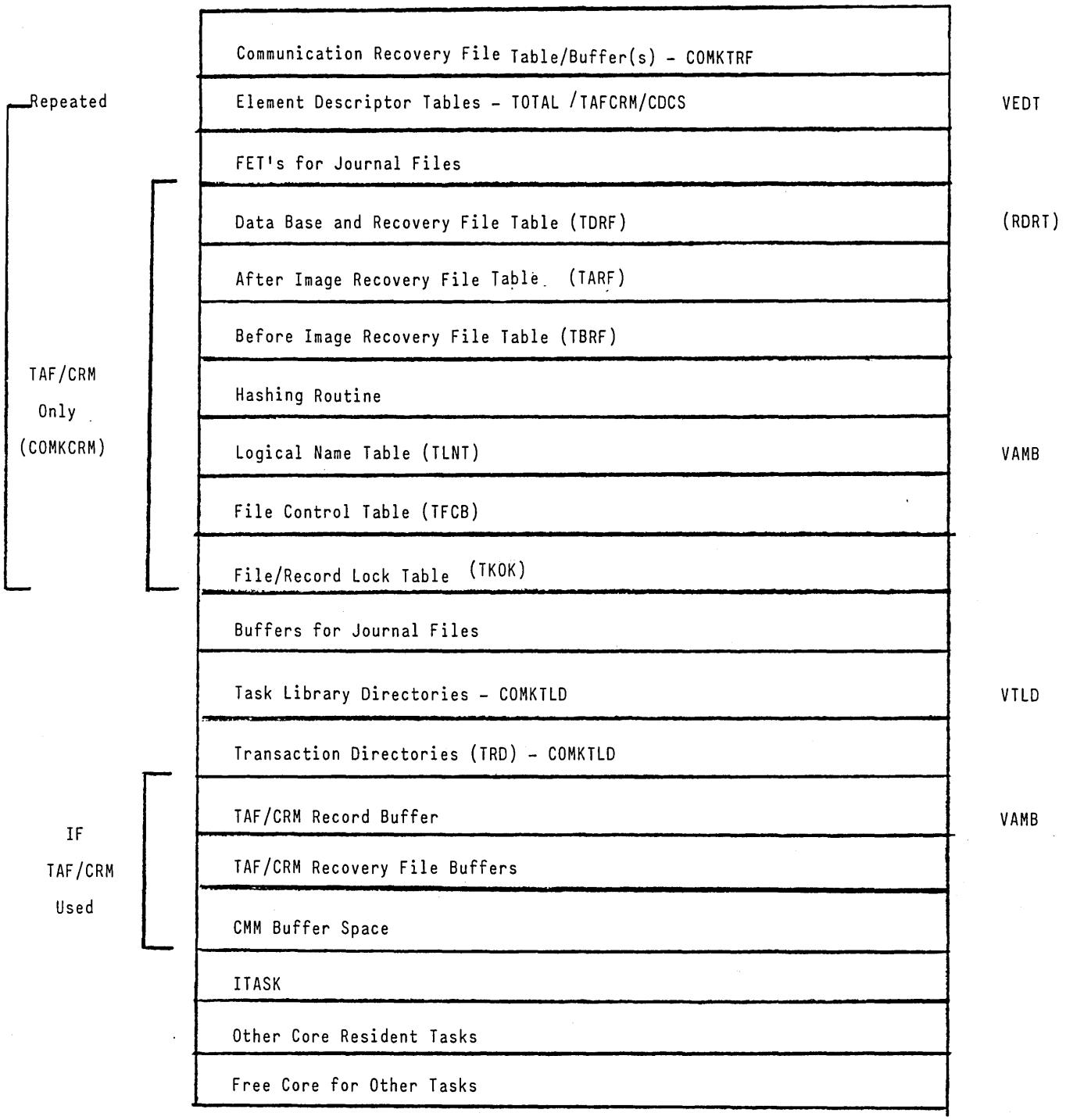
TAF MEMORY MAP



TAF MEMORY MAP



TAF MEMORY MAP (cont)



CATALOG OF TAFPL 1

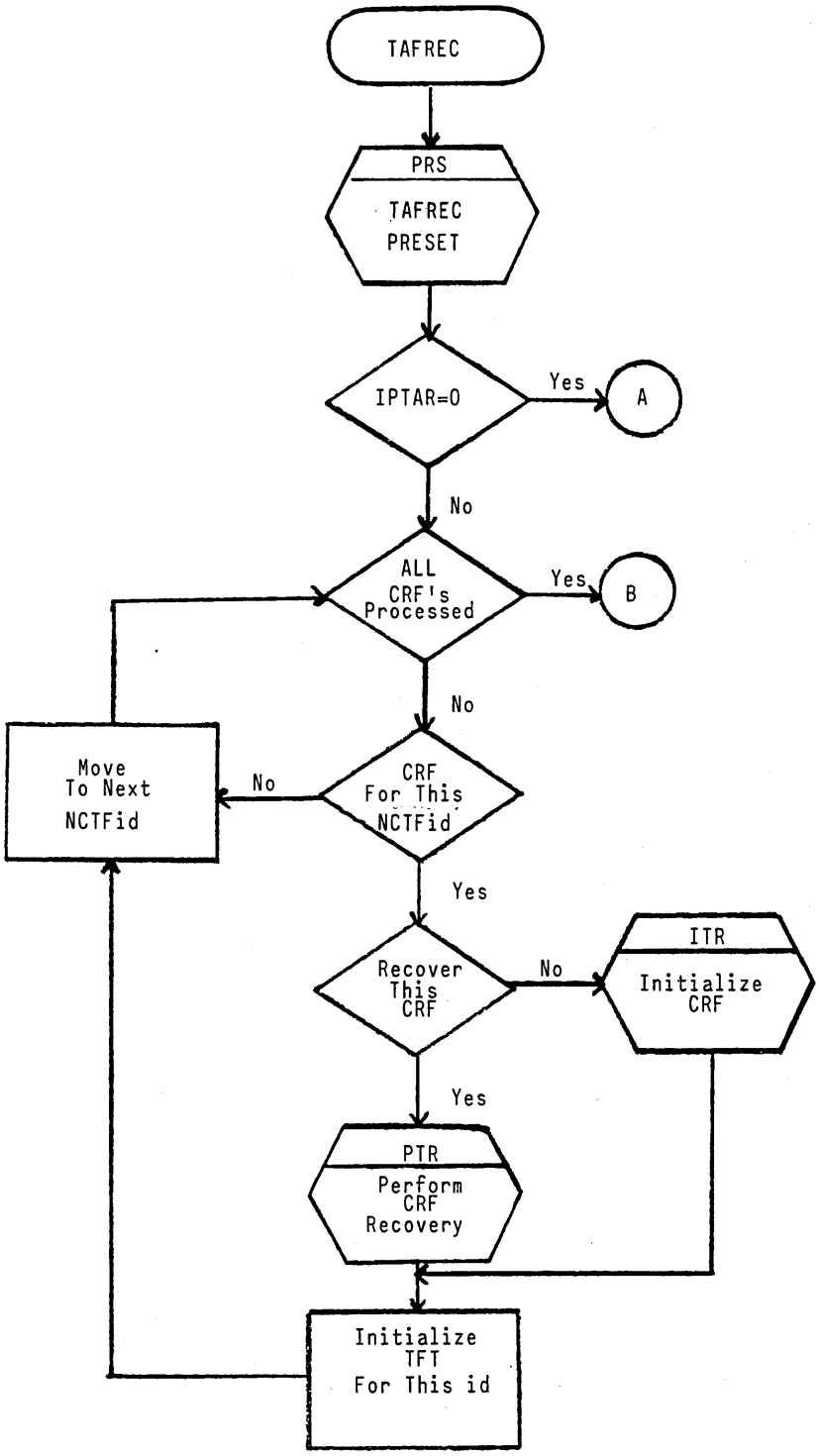
REC	NAME	TYPE	FILE LENGTH	CKSUM	DATE	
1	COPYRT	OPLC (64)	17	2471	82/02/26.	
2	COMKMAC	OPLC (64)	6704	7054	82/02/26.	
	KMAC1					
3	COMKARF	OPLC (64)	3246	1537	82/02/26.	
	NS2023					
4	COMKBRD	OPLC (64)	1707	6671	82/02/26.	
5	COMKBST	OPLC (64)	275	1663	82/02/26.	
6	COMKCBD	OPLC (64)	1712	2035	82/02/26.	
7	COMKCBT	OPLC (64)	327	0376	82/02/26.	
8	COMKCRM	OPLC (64)	13727	3620	82/02/26.	
	NS2001	NS2023				
9	COMKDPB	OPLC (64)	3124	1401	82/02/26.	
10	COMKFIO	OPLC (64)	627	2635	82/02/26.	
11	COMKFLD	OPLC (64)	1060	3500	82/02/26.	
12	COMKIPR	OPLC (64)	3744	0711	82/02/26.	
13	COMKKIM	OPLC (64)	270	1755	82/02/26.	
14	COMKNWC	OPLC (64)	43431	6104	82/02/26.	
	KNWC1	KNWC2	KNWC3	KNWC4	KNWC5	KNWC6
	KNWC7					
15	COMKNWF	OPLC (64)	404	0214	82/02/26.	
16	COMKOPD	OPLC (64)	1516	3521	82/02/26.	
17	COMKRRD	OPLC (64)	574	3414	82/02/26.	
18	COMKSCD	OPLC (64)	677	2213	82/02/26.	
19	COMKSOT	OPLC (64)	212	3666	82/02/26.	
20	COMKSTC	OPLC (64)	250	7003	82/02/26.	
21	COMKTAF	OPLC (64)	63	5463	82/02/26.	
22	COMKTDM	OPLC (64)	154	2025	82/02/26.	
23	COMKTER	OPLC (64)	504	7624	82/02/26.	
24	COMKTIF	OPLC (64)	403	1055	82/02/26.	
25	COMKTIP	OPLC (64)	317	2260	82/02/26.	
26	COMKTFR	OPLC (64)	5	1134	82/02/26.	
27	COMKTLD	OPLC (64)	2121	7775	82/02/26.	
28	COMKTRF	OPLC (64)	2025	5421	82/02/26.	
29	COMKTRN	OPLC (64)	62	1466	82/02/26.	
30	COMKTSA	OPLC (64)	635	6723	82/02/26.	
31	COMKTSC	OPLC (64)	551	0347	82/02/26.	
32	COMKTST	OPLC (64)	1033	1351	82/02/26.	
33	COMKZFN	OPLC (64)	234	0515	82/02/26.	
34	TAFPRC	OPL (64)	272	4644	82/02/26.	
35	CALLKTS	OPL (64)	1125	0424	82/02/26.	
36	1TP	OPL (64)	3056	6544	82/02/26.	
37	KTSDMP	OPL (64)	11150	7500	82/02/26.	
38	LIBTASK	OPL (64)	34374	0102	82/02/26.	
	NS2033	LIBTA1				
39	TAF	OPL (64)	334773	7362	82/02/26.	
	NS2001	NS2023	TAF1	TAF2	TAF4	TAF5
	TAF6	TAF7	TAF8	TAF9	TAF10	TAF12
	NS21000					
40	TAFREC	OPL (64)	40435	2637	82/02/26.	
	NS2023	NS21000				
41	BAAML	OPL (64)	20164	1607	82/02/26.	
42	DMREC	OPL (64)	131072	0004	82/02/26.	
	DMREC1	DMREC2	NS2023	DMREC6	DMREC8	DMREC11
	DMREC12					

```
 43   TARL       OPL  (64)        4124     6765    82/02/26.
 44   TMSG       OPL  (64)        1116     7126    82/02/26.
 45   AAMI       OPL  (64)      136627     3307    82/02/26.
       NS2001     AAMI1      AAMI2      NS2023     AAMI3

 46   AAML       OPL  (64)        5253     7073    82/02/26.
 47   BEGIN      OPL  (64)         456     7457    82/02/26.
 48   BLDABH     OPL  (64)        2011     0377    82/02/26.
 49   CALLRTN    OPL  (64)         641     3331    82/02/26.
 50   CALLTRN    OPL  (64)         345     7565    82/02/26.
 51   CALLTSK    OPL  (64)         701     5012    82/02/26.
 52   CEASE      OPL  (64)         235     2021    82/02/26.
 53   CHKON      OPL  (64)         256     5323    82/02/26.
 54   CMDUMP     OPL  (64)        1553     0063    82/02/26.
 55   DSDUMP     OPL  (64)        1274     1331    82/02/26.
 56   EXTRACT    OPL  (64)        1730     6612    82/02/26.
 57   INTOT      OPL  (64)         733     4261    82/02/26.
 58   JOURNL     OPL  (64)         330     1320    82/02/26.
 59   LIMITS     OPL  (64)         235     6071    82/02/26.
 60   LOGIN      OPL  (64)         723     7072    82/02/26.
 61   MULTCB     OPL  (64)         455     4435    82/02/26.
 62   SEND       OPL  (64)        3536     2426    82/02/26.
 63   SETCHT     OPL  (64)        1031     7332    82/02/26.
 64   SUBMT      OPL  (64)         611     3366    82/02/26.
 65   TARO       OPL  (64)         714     2307    82/02/26.
 66   TERMDEF    OPL  (64)        4433     4761    82/02/26.
 67   TSIM       OPL  (64)         660     5377    82/02/26.
 68   TSTAT      OPL  (64)         467     2345    82/02/26.
 69   WAIT       OPL  (64)         401     7152    82/02/26.
 70   WAITINP    OPL  (64)         334     2431    82/02/26.
 71   WSTAT      OPL  (64)         466     5443    82/02/26.
 72   BTASK      OPL  (64)        2107     0762    82/02/26.
       BTASK1

 73   CRMTASK    OPL  (64)       17523     6647    82/02/26.
 74   CTASK      OPL  (64)       11551     7571    82/02/26.
       NS2001

 75   ITASK      OPL  (64)        5043     4630    82/02/26.
 76   KDIS       OPL  (64)         730     2445    82/02/26.
 77   LOGT       OPL  (64)         356     4672    82/02/26.
 78   MSABT      OPL  (64)        2106     7533    82/02/26.
 79   OFFTASK    OPL  (64)         477     5613    82/02/26.
 80   RCTASK     OPL  (64)        2013     1204    82/02/26.
 81   RTASK      OPL  (64)        3165     4736    82/02/26.
 82   SYSMSG     OPL  (64)        1321     6113    82/02/26.
 83   XTASK      OPL  (64)        1161     7423    82/02/26.
 84   TAFPL.     OPLD              251     5706   82/10/25.

 85   * EOF *          SUM =    1230063
      CATALOG OF TAFPL            FILE     2
REC   NAME       TYPE            LENGTH   CKSUM      DATE


      * EOI *          SUM =          0
CATALOG COMPLETE.
```

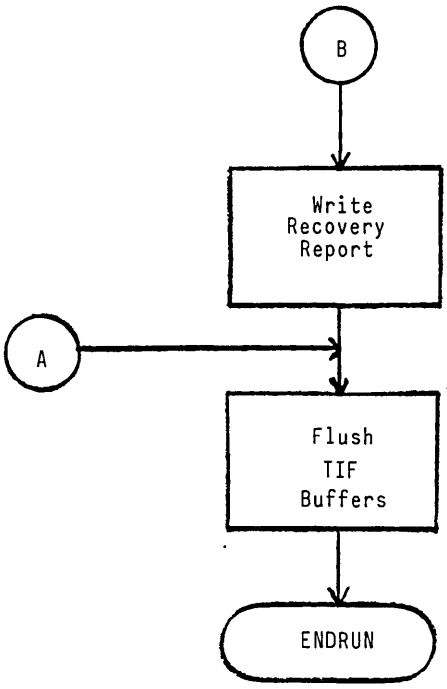
INITIALIZATION

```
.PROC,TAFPRC.
* ENTER USER CONTROL STATEMENT HERE.
OFFSW(3)
ONSW(5)
SET,EF=0.
WHILE,EF.NE.ODE.AND.EF.NE.ORE,ENDTAF.
RFL(160000)
TAFREC.
OFFSW(3)
IFE,EF.NE.0,TAFRECABT.
EXIT.
DMD.
DMD,0,377777.
*
* TAFREC ABORT.
*
REWIND,ZZIRRF,TCF,ZZTIF.
* LIST TAF RECOVERY REPORT.
COPYEI,ZZIRRF,OUTPUT.
* LIST TAF CONFIGURATION FILE.
COPYSBF,TCF,OUTPUT.
* LIST TAF INITIALIZATION FILE.
TDUMP,I=ZZTIF,L=OUTPUT.
EXIT.
ENDIF,TAFRECABT.
TAF1.
TAF2.
EXIT.
DMD.
DMD,0,377777.
*
* TAF ABORT.
*
TAF2.
DLFP.
REWIND,ZZIRRF,TCF.
* LIST TAF INITIALIZATION OR RECOVERY REPORT.
COPYEI,ZZIRRF,OUTPUT.
* LIST TCF.
COPYSBF,TCF,OUTPUT.
DAYFILE.
ROUTE,OUTPUT,DC=PR.
ONSW(3)
SET,EF=0.
IFE,EF.NE.0,TAF2ABT.
EXIT.
DMD.
DMD,0,377777.
DLFP.
REWIND,ZZIRRF,TCF.
* LIST TAF INITIALIZATION OR RECOVERY REPORT.
COPYEI,ZZIRRF,OUTPUT.
* LIST TCF.
COPYSBF,TCF,OUTPUT.
DAYFILE.
ROUTE,OUTPUT,DC=PR.
ONSW(3)
SET,EF=0.
EXIT.
ENDIF,TAF2ABT.
ENDW,ENDTAF.
```

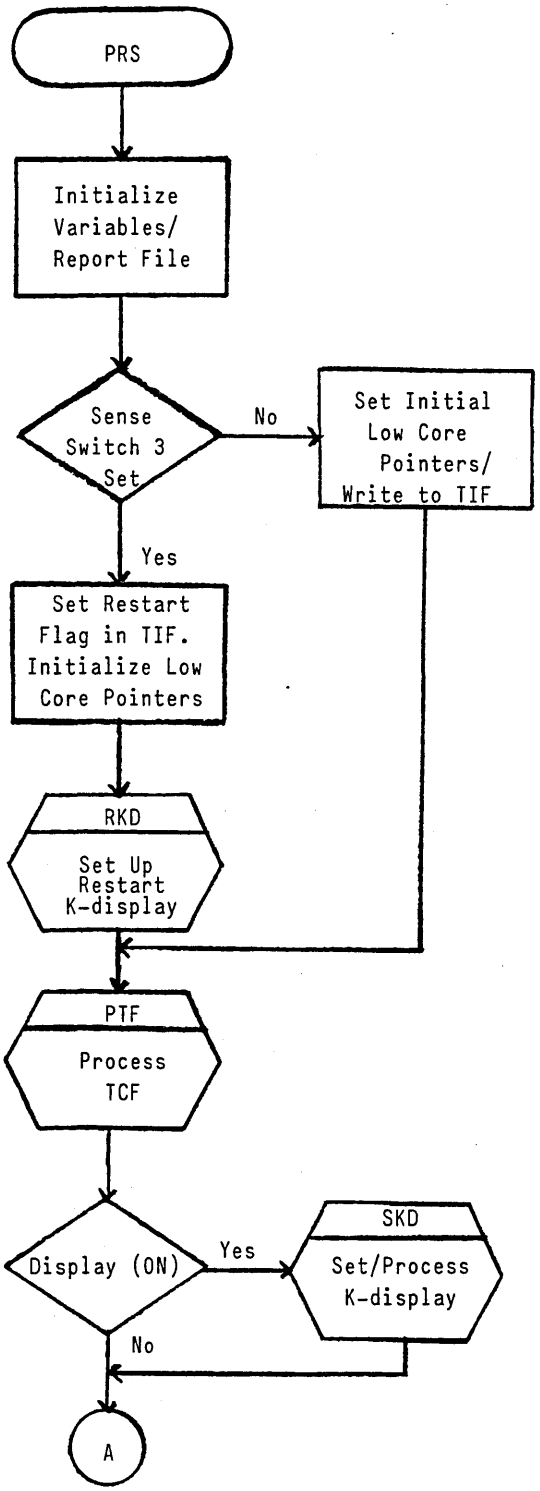
TAFREC - Main Loop



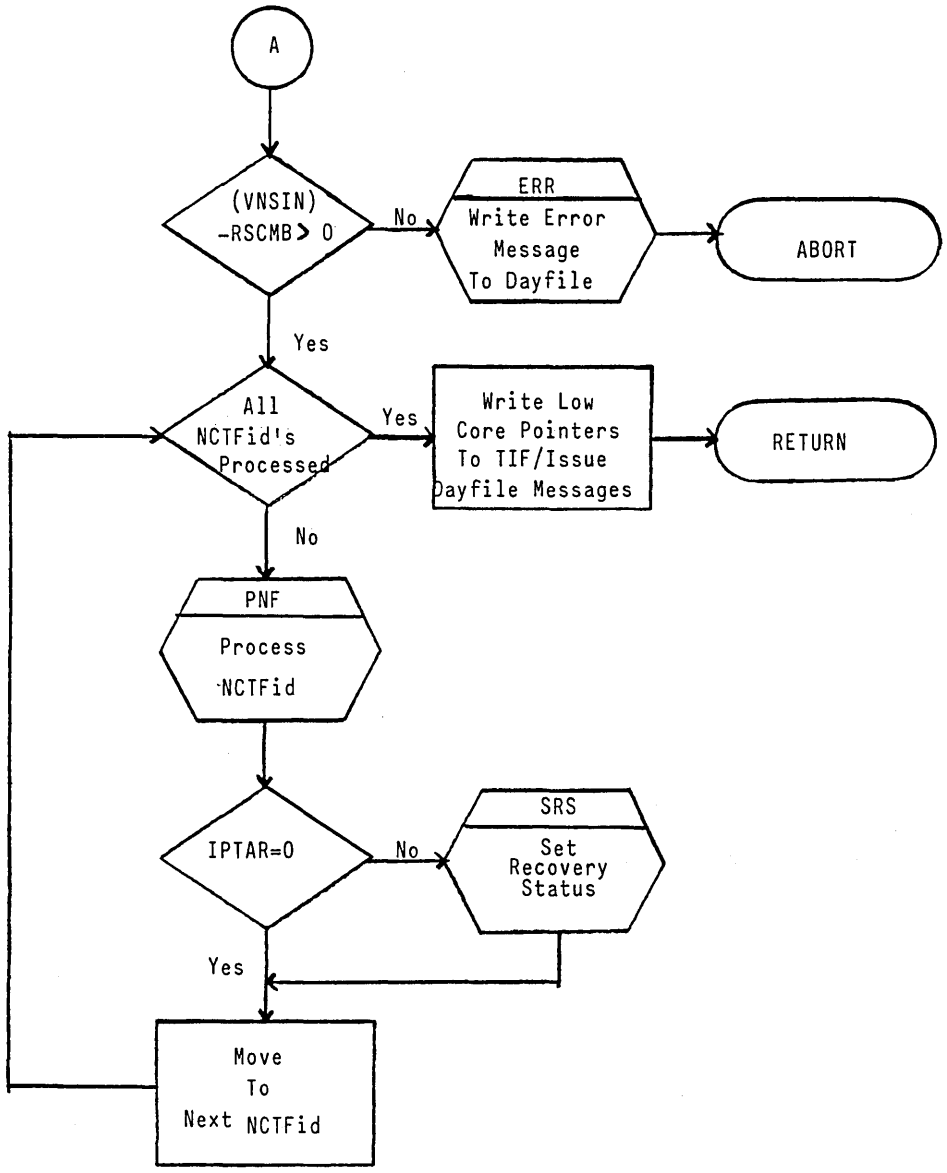
TAFREC - Main Loop (cont)



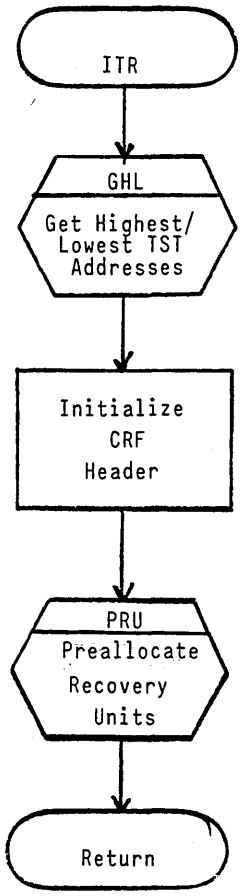
TAFREC - Preset



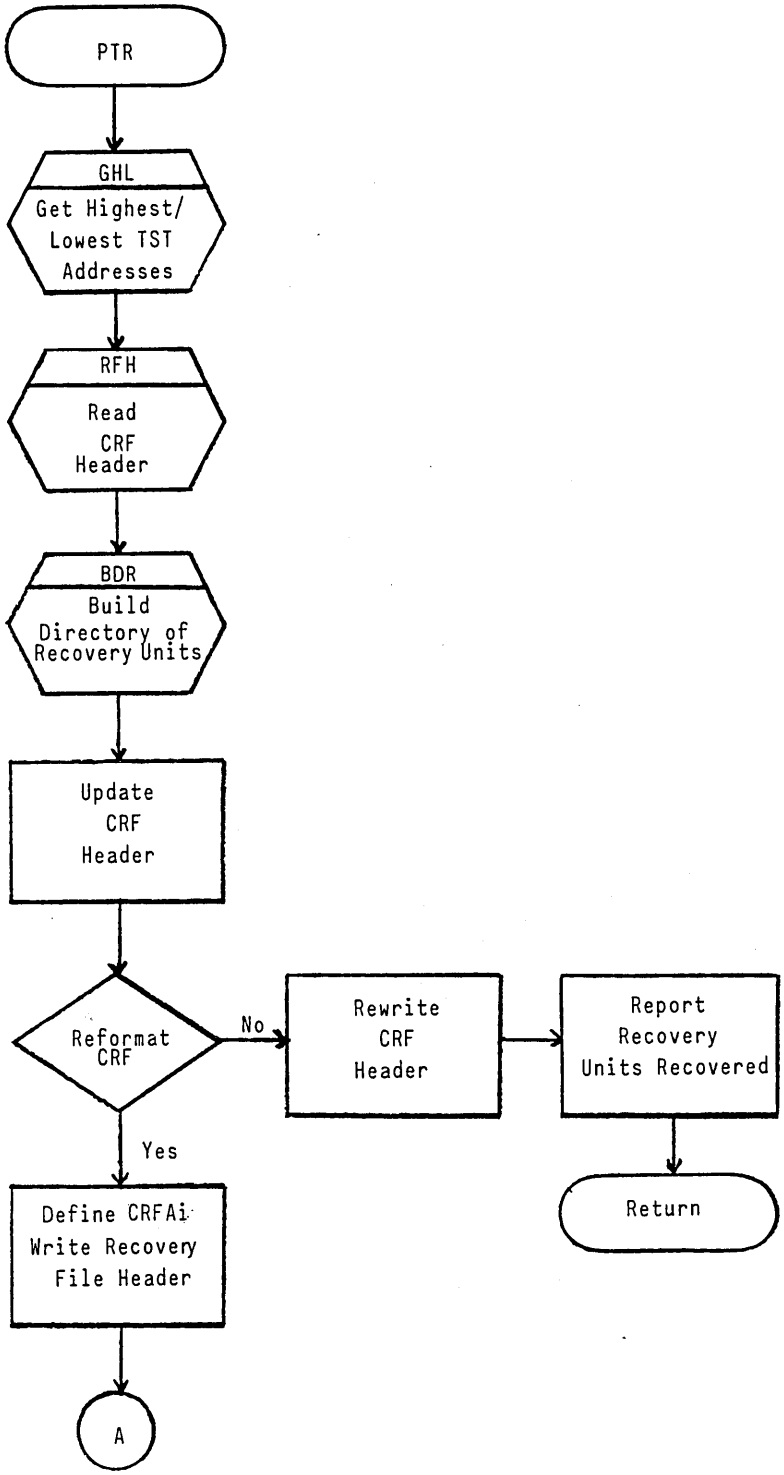
TAFREC - Preset (cont)



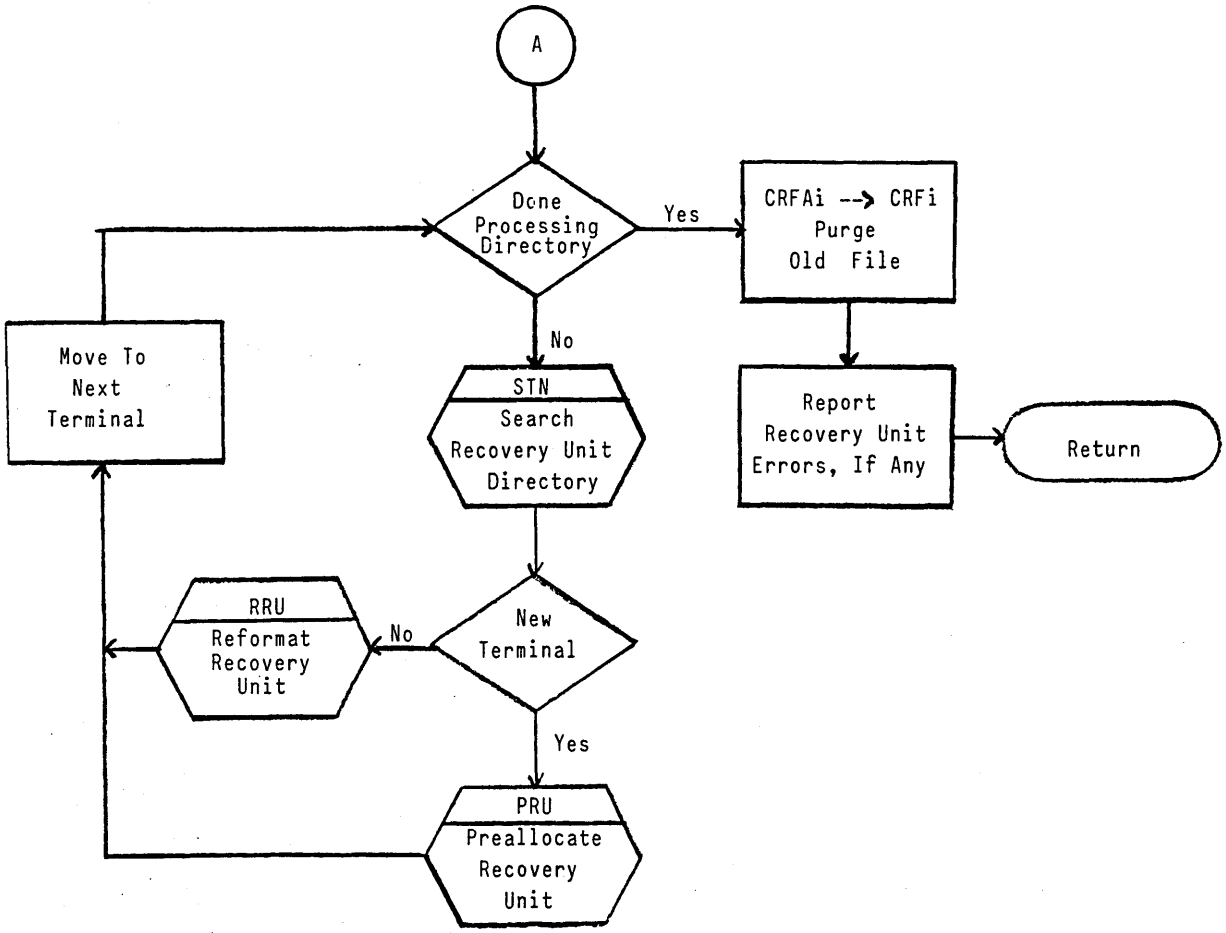
TAFREC - Initialize CRF



TAFREC - Recover CRF



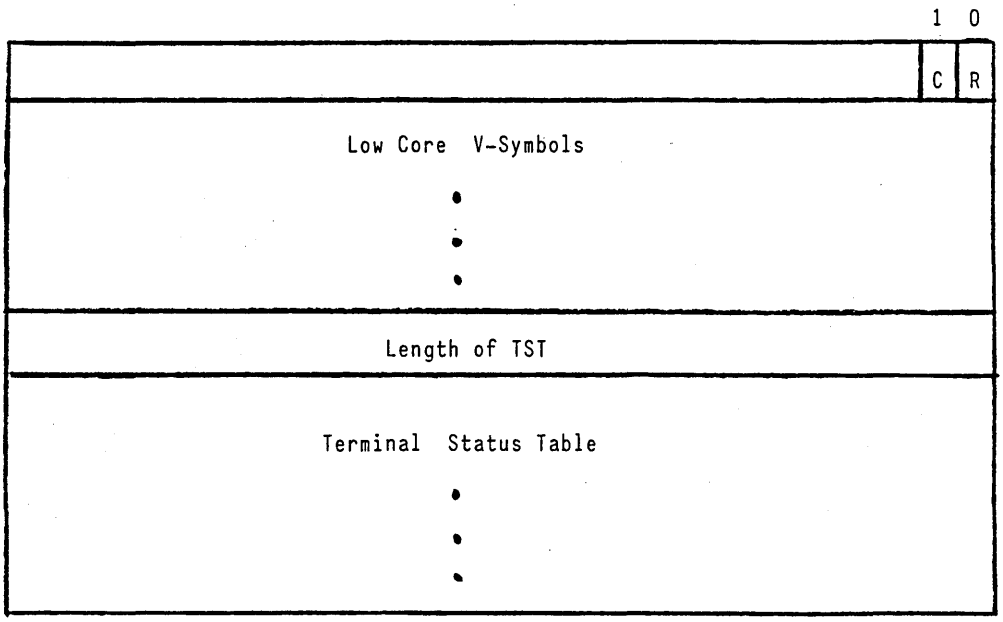
TAFREC - Recover CRF (cont)



TAF INITIALIZATION FILE - TIF
(COMKTIF)

```
***         *COMKTIF* PROVIDES A DESCRIPTION AND COMMON
*           DEFINITIONS OF THE *TAF* INITIALIZATION FILE FOR *TAFREC*,
*           *TAF1* AND *TAF2*.
*
*           THE *TIF* IS A SEQUENTIAL FILE CREATED BY *TAFREC* DURING
*           INITIALIZATION.  IF THE *TIF* IS LOCAL WHEN *TAFREC* BEGINS
*           EXECUTION, IT INDICATES *TAF* RESTART.  THAT IS, *TAF* HAS
*           ABORTED, BUT *TAF2* GAINED CONTROL DUE TO THE *EXIT*
*           STATEMENT IN THE PROCEDURE FILE, TO CLEAN UP.  FOLLOWING
*           CONTROL STATEMENTS TRANSFERRED CONTROL BACK TO *TAFREC*.
*
*           THE FILE LAYOUT IS -
*
*T  W1     58/ ,1/ C,1/ R
*T, W2     60/           LOW CM V-SYMBOLS
*T,        60/             DEFINED BY
*T,        60/              *COMSTRX*
*T, W(I)   60/
*T, W(J)   60/ TSTL
*T,        60/              TERMINAL
*T,        60/               STATUS
*T, W(N)   60/               TABLE.
*
*           WHERE
*                  C    = 1, IF *CRM* RECOVERY FILE IS TO BE INITIALIZED.
*                  R    = 1, IF *TAF* IS TO BE RESTARTED.
*                  I    = 2+VLOCL  (*VLOCL* DEFINED IN *COMSTRX*).
*                  J    = I+1.
*                  TSTL = LENGTH OF TERMINAL STATUS TABLE.
*                  N    = J+TSTL.
*
*           *TAF1* WILL READ THE *TIF* FILE AND LOAD THE V-SYMBOLS AND
*           *TST* INTO MEMORY.  IF NOT TAF RESTART, *TAF1* WILL
*           INITIALIZE V-SYMBOLS WHICH COULD NOT BE INITIALIZED IN
*           *TAFREC*.  *TAF1* REWINDS THE *TIF* FILE AND WRITES ALL
*           INITIALIZED V-SYMBOLS BACK OVER THE *TIF*.


 TICR      FIELD  0,1,1        1, IF INITIALIZE *CRM* RECOVERY FILE
 TIRF      FIELD  0,0,0        1, IF *TAF* RESTART
```

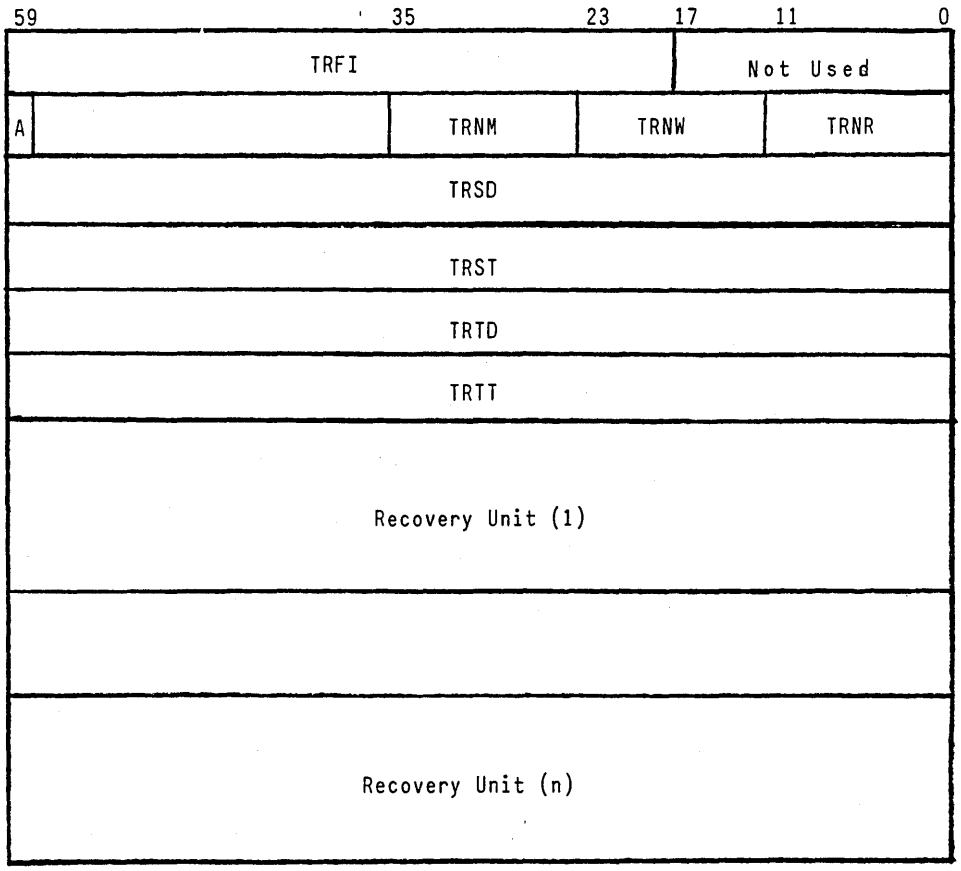
TAF INITIALIZATION FILE - TIF
(COMKTIF)



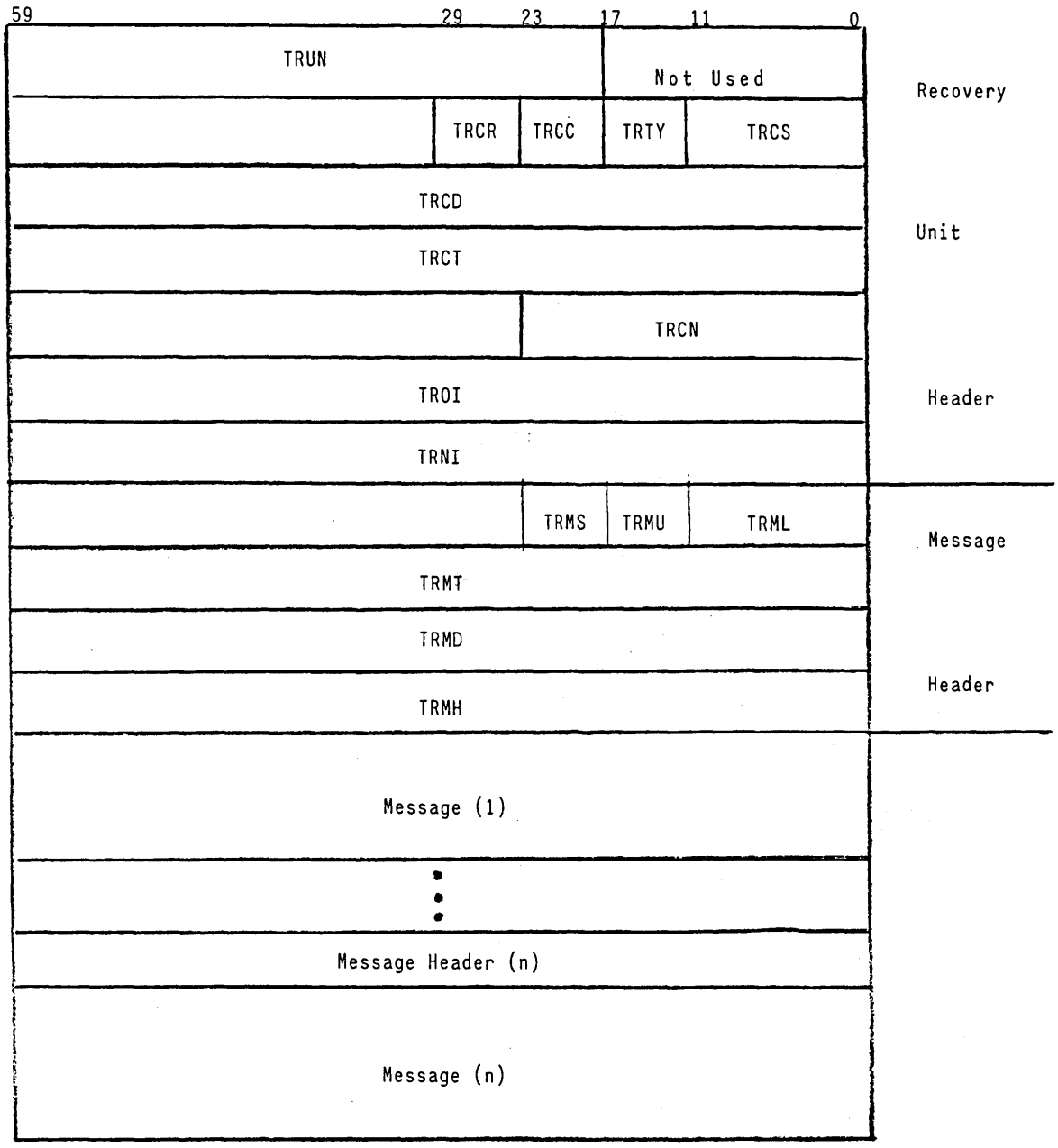
COMMUNICATION RECOVERY FILE - CRF
(COMKTRF)

```
***     *COMKTRF* PROVIDES DEFINITIONS FOR THE *TAF*
*       RECOVERY FILE.  THIS DECK REQUIRES *COMKFLD*.
*
*       THE *TAF* RECOVERY FILE, TRF IS A FIXED LENGTH,
*       RANDOM ACCESS FILE USED BY *TAF* TO RECORD
*       RECOVERY INFORMATION ABOUT RECOVERABLE TRANSACTIONS.
*       THERE IS A TRF FOR EACH NETWORK FILE IN USE IF *TAF*
*       AUTOMATIC RECOVERY IS ENABLED.
*       THE TRF CONSISTS OF A HEADER RECORD, AND RECOVERY UNIT
*       RECORDS.  THERE IS ONE RECOVERY UNIT FOR EACH TRANSACTION
*       TERMINAL DECLARED IN THE CORRESPONDING NETWORK FILE.
*
*       EACH RECOVERY UNIT CONSISTS OF A HEADER, AND TWO OR MORE
*       MESSAGE RECORDS.  THE FIRST MESSAGE RECORD CONSISTS OF THE
*       TERMINAL OR *BTRAN* INPUT THAT INITIATED THE RECOVERABLE
*       TRANSACTION.  THE SECOND MESSAGE RECORD CONSISTS OF AN OUTPUT
*       MESSAGE THAT WILL BE SENT TO THE TERMINAL UPON SUCCESSFUL
*       COMPLETION OF THE RECOVERABLE TRANSACTION CONFIRMING
*       SUCCESSFUL COMPLETION TO THE TERMINAL OPERATOR.
*       THE THIRD TO NTH MESSAGE RECORDS (N .LE. 12) CONSIST OF USER
*       RECOVERY INFORMATION RECORDED BY THE TASK *RPUT* REQUEST.
*       THESE MESSAGES CAN BE RETRIEVED BY THE TASK *RGET* REQUEST.
```

COMMUNICATION RECOVERY FILE
(COMKTRF)



COMMUNICATION RECOVERY FILE
(COMKTRF)



COMMUNICATION RECOVERY FILE - CRF
(COMKTRF)

```
**          TRF HEADER RECORD.
*
*T  W1      42/ TRFI,18/
*T, W2      1/A,25/ ,12/ TRNM,12/ TRNW,12/ TRNR
*T, W3      60/ TRSD
*T, W4      60/ TRST
*T, W5      60/ TRTD
*T, W6      60/ TRTT
*
*           WORD 1.
*                 TRFI - LOGICAL FILE NAME.
*
*           WORD 2.
*                 A    - 1, IF ABNORMAL SHUTDOWN.
*                 TRNM - NUMBER OF MESSAGES PER RECOVERY UNIT.
*                 TRNW - MAXIMUM MESSAGE SIZE IN WORDS.
*                 TRNR - NUMBER OF RECOVERY UNITS.
*
*           WORD 3.
*                 TRSD - STARTUP DATE, YY/MM/DD.
*
*           WORD 4.
*                 TRST - STARTUP TIME, HH.MM.SS.
*
*           WORD 5.
*                 TRTD - TERMINATE DATE, YY/MM/DD.
*
*           WORD 6.
*                 TRTT - TERMINATE TIME, HH.MM.SS.


 TRFI      FIELD  0,59,18      LOGICAL FILE NAME
 TRRS      FIELD  1,59,59      1, IF ABNORMAL SHUTDOWN
 TRNM      FIELD  1,35,24      NUMBER OF MESSAGES FOR RECOVERY UNIT
 TRNW      FIELD  1,23,12      MAXIMUM MESSAGE SIZE IN WORDS
 TRNR      FIELD  1,11,0       NUMBER OF RECOVERY UNITS
 TRSD      FIELD  2,59,0       STARTUP DATE
 TRST      FIELD  3,59,0       STARTUP TIME
 TRTD      FIELD  4,59,0       TERMINATION DATE
 TRTT      FIELD  5,59,0       TERMINATION TIME
```

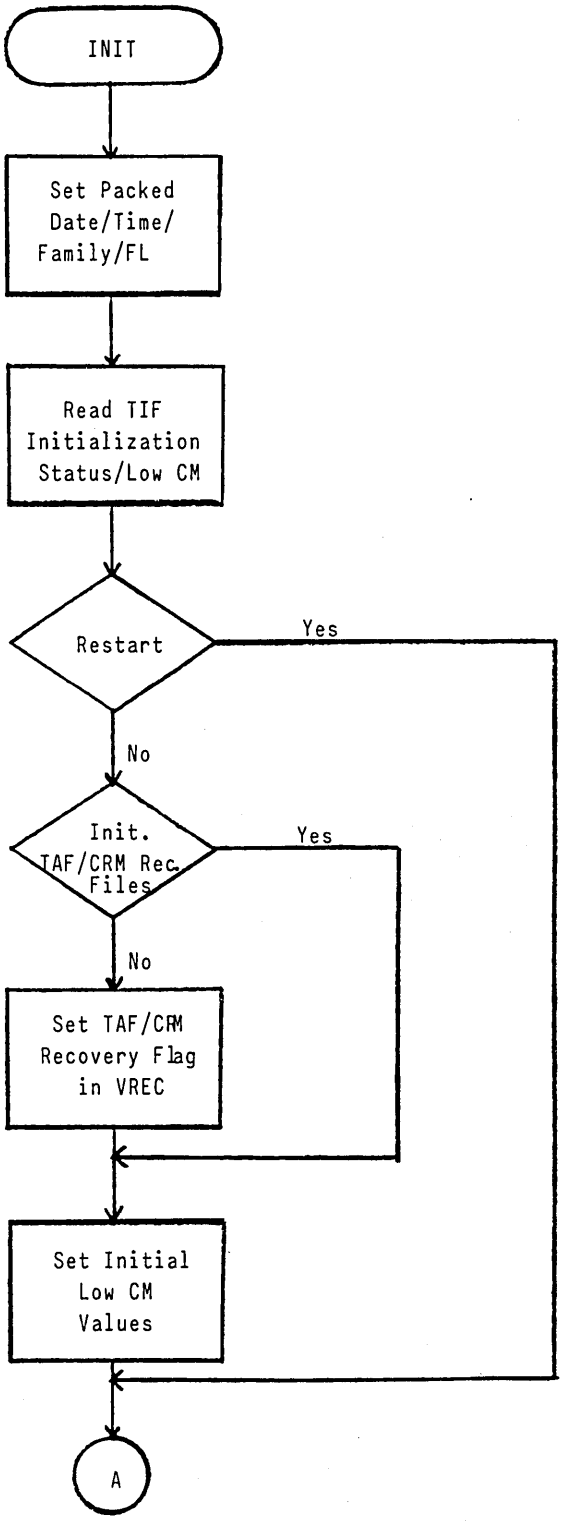
COMMUNICATION RECOVERY FILE - CRF
(COMKTRF)

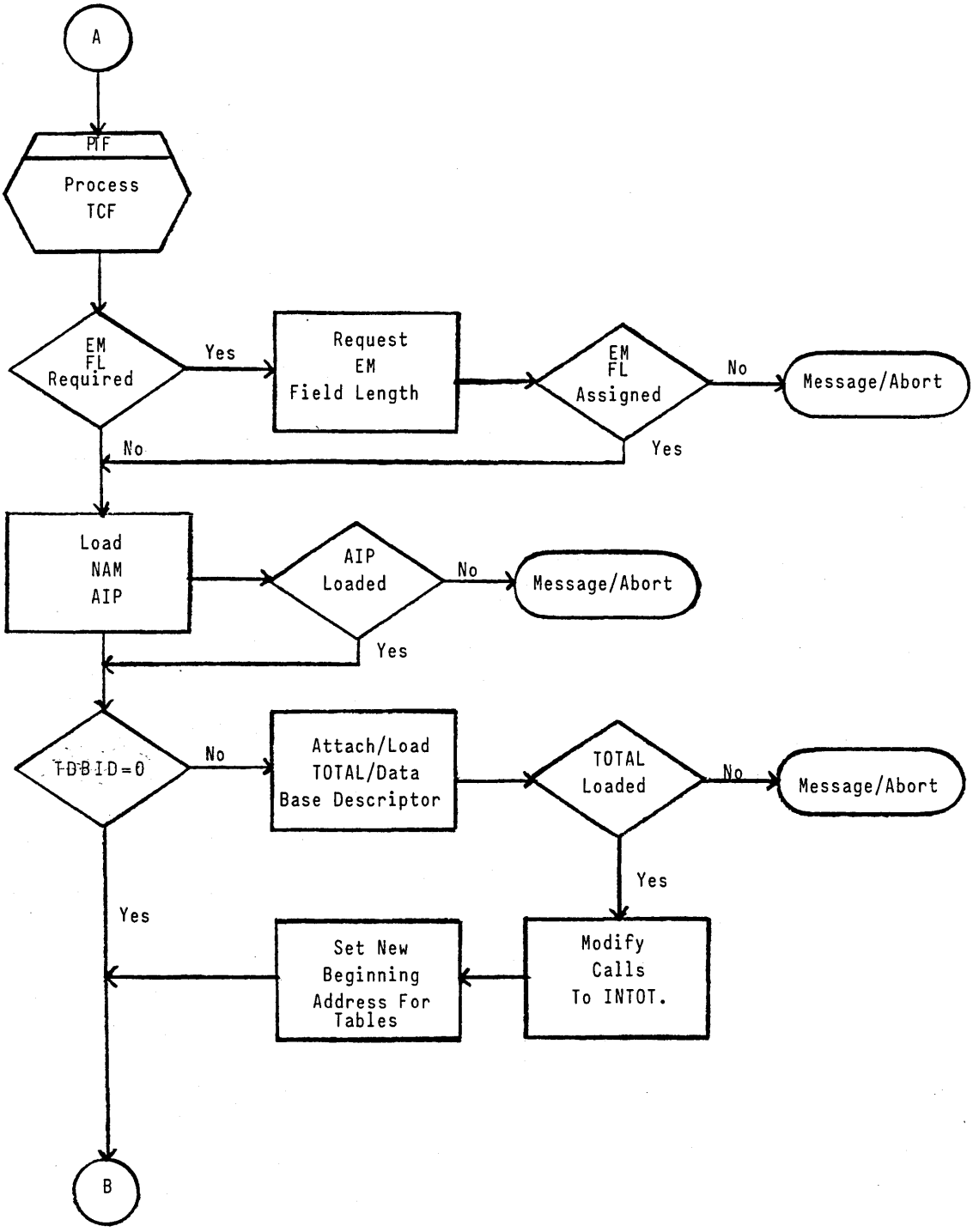
```
**        RECOVERY UNIT HEADER.
*
*T  W1    42/ TRUN,18/
*T, W2    30/ ,6/ TRCR,6/ TRCC,6/ TRTY,12/ TRCS
*T, W3    60/ TRCD
*T, W4    60/ TRCT
*T, W5    36/ ,24/ TRCN
*T, W6    60/ TROI
*T, W7    60/ TRNI
*
*         WORD 1.
*               TRUN - RECOVERY UNIT NAME.
*
*         WORD 2.
*               TRCR - 1, IF *CRM* ALLOWED.
*               TRCC  - 1, IF *CDCS* ALLOWED.
*               TRTY  - TRANSACTION TYPE.  (*TSTAT* KEYWORD *TRAN*)
*               TRCS  - TRANSACTION STEP.  (*TSTAT* KEYWORD *STEP*)
*
*         WORD 3.
*               TRCD - CURRENT STEP DATE, YY/MM/DD.
*
*         WORD 4.
*               TRCT - CURRENT STEP TIME, HH.MM.SS.
*
*         WORD 5.
*               TRCN - TRANSACTION SEQUENCE NUMBER.
*
*         WORD 6.
*               TROI - OLD *CRM* BEGIN-COMMIT IDENTIFIER.
*
*         WORD 7.
*               TRNI - NEW *CRM* BEGIN-COMMIT IDENTIFIER.


 TRUN     FIELD  0,59,18     RECOVERY UNIT NAME
 TRCR     FIELD  1,29,24     1, IF TRANSACTION CAN USE *CRM*
 TRCC     FIELD  1,23,18     1, IF TRANSACTION CAN USE *CDCS*
 TRTY     FIELD  1,17,12     TRANSACTION TYPE
 TRCS     FIELD  1,11,0      TRANSACTION STEP
 TRCD     FIELD  2,59,0      CURRENT STEP DATE
 TRCT     FIELD  3,59,0      CURRENT STEP TIME
 TRCN     FIELD  4,23,0      CURRENT TRANSACTION SEQUENCE NUMBER
 TROI     FIELD  5,59,0      OLD *CRM* BEGIN-COMMIT IDENTIFIERS
 TRNI     FIELD  6,59,0      NEW *CRM* BEGIN-COMMIT IDENTIFIERS
```

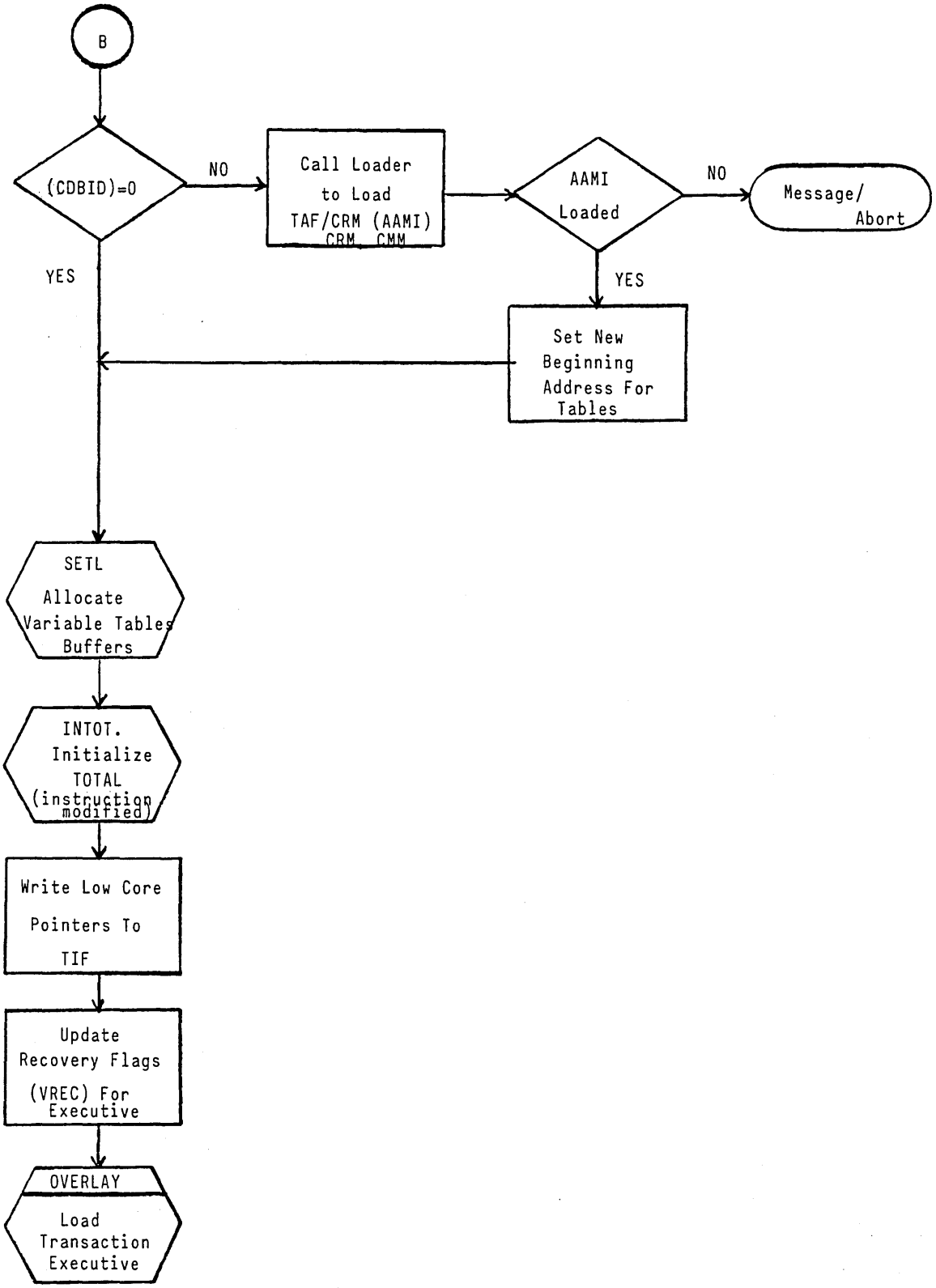
COMMUNICATION RECOVERY FILE - CRF
(COMKTRF)

```
**          MESSAGE HEADER.
*
*T  W1      36/ ,6/ TRMS,6/ TRMS,12/ TRML
*T, W2      60/ TRMT
*T, W3      60/ TRMD
*T, W4      60/ TRMH
*
*           WORD 1.
*                 TRMS - MESSAGE SOURCE.
*                           0 = NO MESSAGE.
*                           1 = RERUNNABLE TRANSACTION INPUT.
*                           2 = *SECURE* MESSAGE.
*                           3 = *RPUT* MESSAGE.
*                 TRMU - MESSAGE CHARACTER TYPE IN *NAM* UNITS.
*                 TRML - MESSAGE LENGTH IN UNITS OF *TRMU*.
*
*           WORD 2.
*                 TRMT - MESSAGE TIME, HH.MM.SS.
*
*           WORD 3.
*                 TRMD - MESSAGE DATE, YY/MM/DD.
*
*           WORD 4.
*                 TRMH - NAM APPLICATION BLOCK HEADER.


 TRMS       FIELD  0,23,18      MESSAGE SOURCE
 TRMU       FIELD  0,17,12      MESSAGE UNITS
 TRML       FIELD  0,11,0       MESSAGE LENGTH IN UNITS OF *TRMU*
 TRMT       FIELD  1,59,0       MESSAGE TIME
 TRMD       FIELD  2,59,0       MESSAGE DATE
 TRMH       FIELD  3,59,0       NAM APPLICATION BLOCK HEADER
```

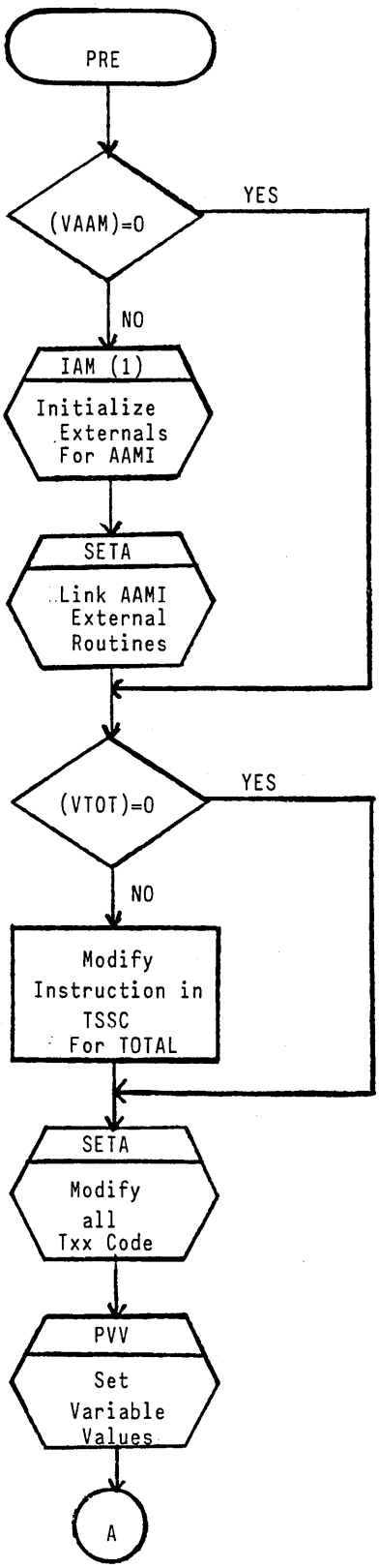
TAF1

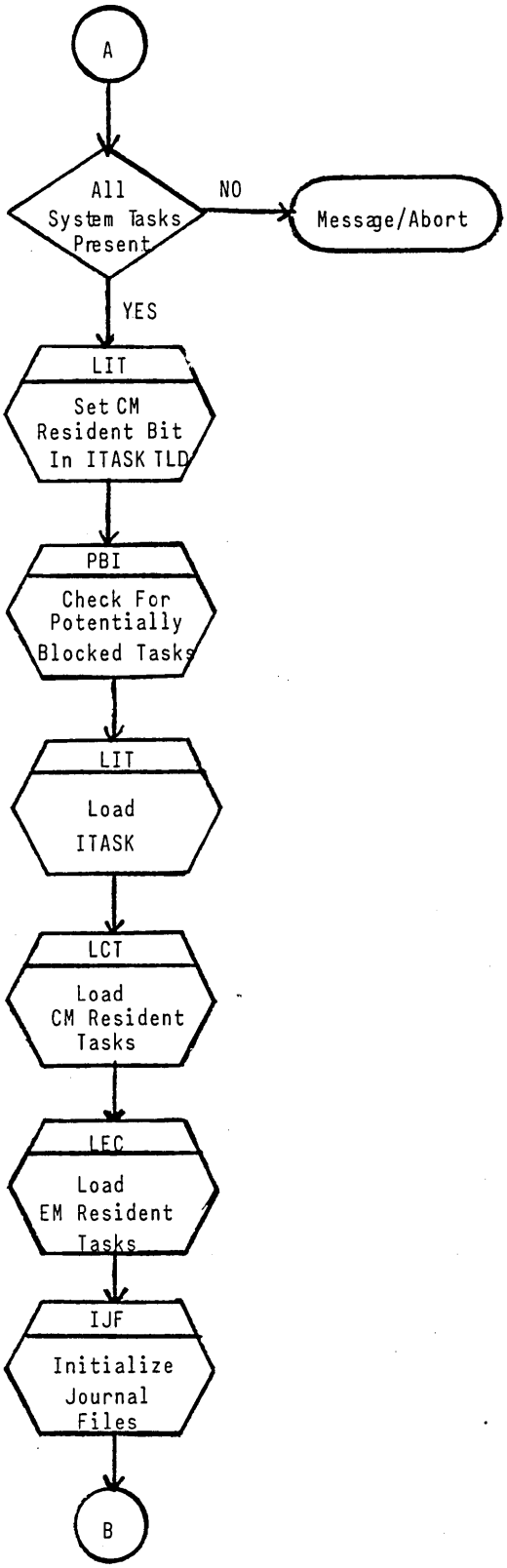

INIT
Set Packed Date/Time/ Family/FL
Read TIF Initialization Status/Low CM
Restart
Yes
No
Init. TAF/CRM Rec. Files
Yes
No
Set TAF/CRM Recovery Flag in VREC
Set Initial Low CM Values
A

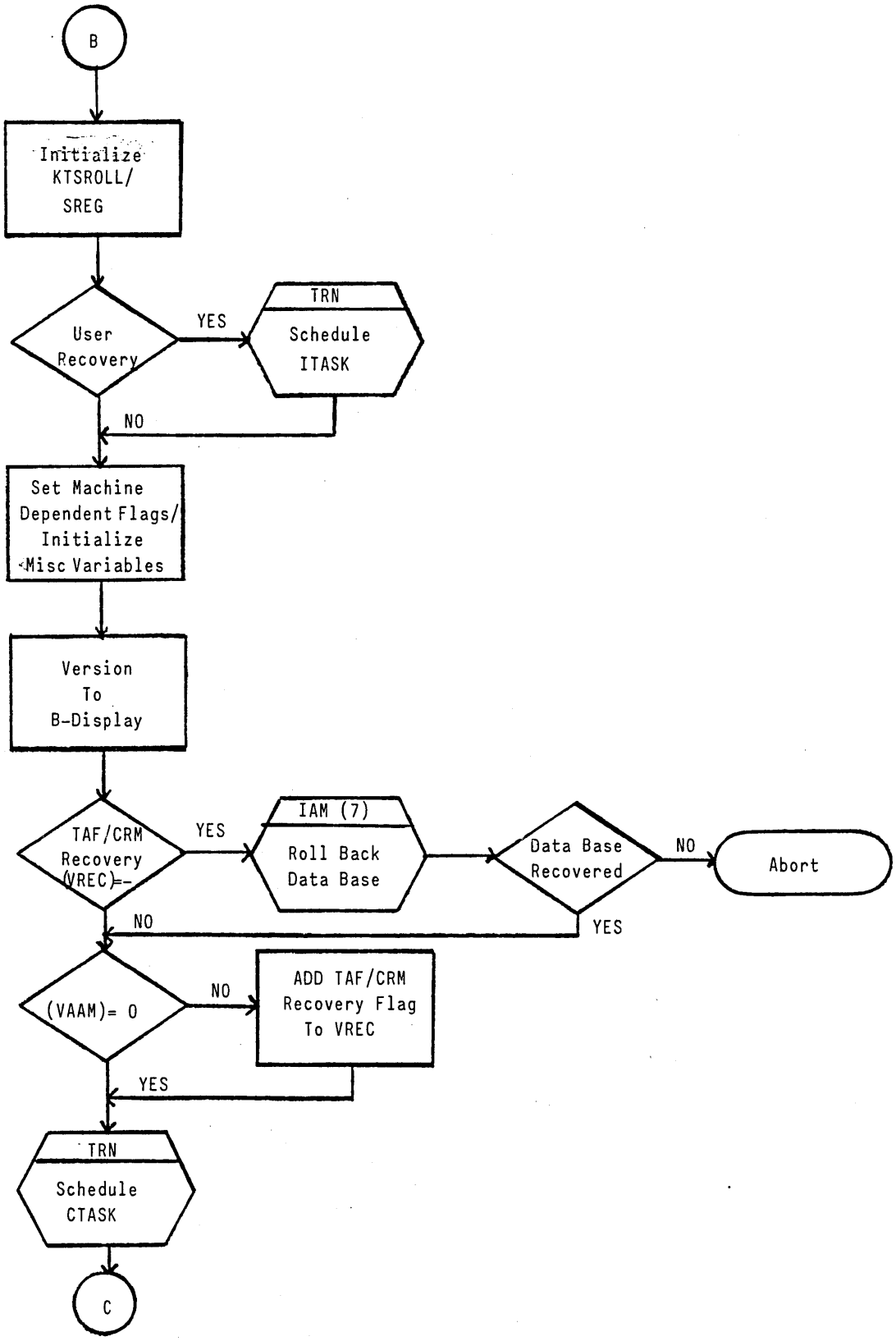

A
PIF
Process TCF
EM FL Required
Yes
Request EM Field Length
EM FL Assigned
No
Message/Abort
Yes
No
Load NAM AIP
AIP Loaded
No
Message/Abort
Yes
TDBID=0
No
Attach/Load TOTAL/Data Base Descriptor
TOTAL Loaded
No
Message/Abort
Yes
Yes
Modify Calls To INTOT.
Set New Beginning Address For Tables
B


B
(CDBID)=0
NO
Call Loader to Load TAF/CRM (AAMI) CRM CMM
AAMI Loaded
NO
Message/ Abort
YES
YES
Set New Beginning Address For Tables
SETL
Allocate Variable Tables Buffers
INTOT.
Initialize TOTAL (instruction modified)
Write Low Core Pointers To TIF
Update Recovery Flags (VREC) For Executive
OVERLAY
Load Transaction Executive

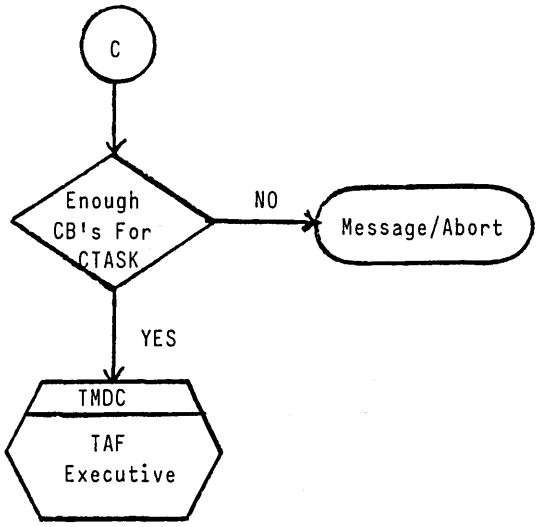
TAF Preset - PRE




A
All System Tasks Present
NO
Message/Abort
YES
LIT
Set CM Resident Bit In ITASK TLD
PBI
Check For Potentially Blocked Tasks
LIT
Load ITASK
LCT
Load CM Resident Tasks
LEC
Load EM Resident Tasks
IJF
Initialize Journal Files
B


B
Initialize KTSROLL/ SREG
User Recovery
YES
TRN
Schedule ITASK
NO
Set Machine Dependent Flags/ Initialize Misc Variables
Version To B-Display
TAF/CRM Recovery (VREC)=-
YES
IAM (7)
Roll Back Data Base
Data Base Recovered
NO
Abort
NO
YES
(VAAM)= 0
NO
ADD TAF/CRM Recovery Flag To VREC
YES
TRN
Schedule CTASK
C

TAF Preset - PRE (cont)



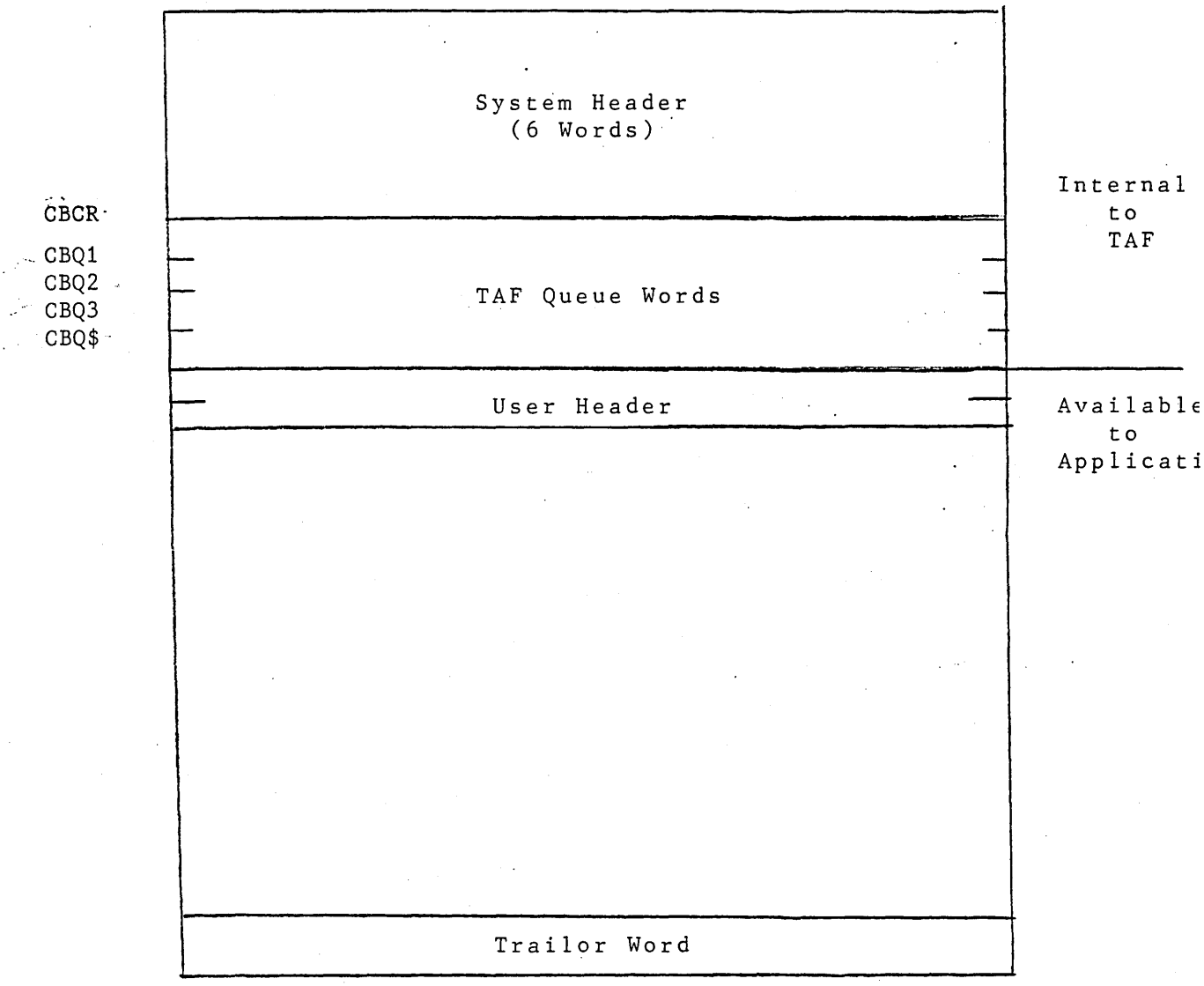
TABLES

COMMUNICATION BLOCK - CB
(COMKCBD)

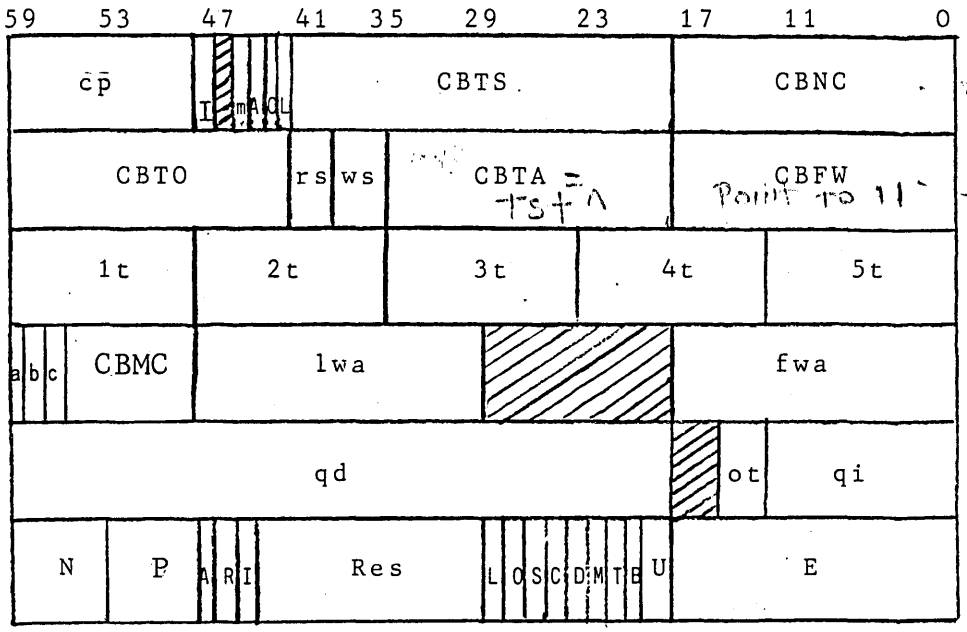
THE COMMUNICATION BLOCKS (CB) MAIN PURPOSE IS TO FUNCTION AS AN INPUT FILE FOR TAF. AS SOON AS TAF RECEIVES INPUT FROM A TERMINAL, IT PUTS THE INPUT IN A CB WHERE IT WILL REMAIN UNTIL TRANSACTION COMPLETION OR THE TASK DOES A WAIT FOR TERMINAL INPUT. A TRANSACTION CAN CONSIST OF MULTIPLE TASKS WITH THE COMMUNICATION BLOCK BEING THE MEANS TO PASS INPUT AND DATA FROM ONE TASK TO THE NEXT. THERE ARE THREE PARTS TO THE CB - THE SYSTEM HEADER, USER HEADER, AND INPUT FROM THE TERMINAL. THE SYSTEM HEADER CONTAINS CONTROL INFORMATION STRICTLY FOR TAF'S USAGE. THE USER HEADER IS ACCESSABLE BY THE TASK, BUT CANNOT BE CHANGED EXCEPT BY TAF. THE REST (66 WORDS) IS AVAILABLE FOR USAGE BY THE TASKS. THE LAST WORD CONTAINS THE PACKED DATE AND TIME. INITIALLY, THE INPUT FROM THE TERMINAL GETS STORED THERE AND PASSED TO THE FIRST TASK. THAT TASK THEN HAS THE OPTION OF PASSING THE INPUT ON TO THE NEXT TASK (IF THERE IS ONE) AS IS OR CHANGING IT. TAF ALWAYS COPIES THAT PART OF THE CB FROM THE TASK BACK INTO THE EXECUTIVE PORTION OF ITS FIELD LENGTH AND BACK OUT TO THE NEW TASK.

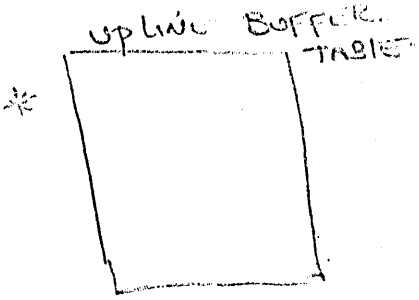
LOW CORE POINTER WORDS VCBRT (20B) AND VCBSA (21B) CONTAIN THE ADDRESS OF THE CB RESERVATION MAP WORD, AND THE FIRST WORD ADDRESS OF THE FIRST CB, RESPECTIVELY.

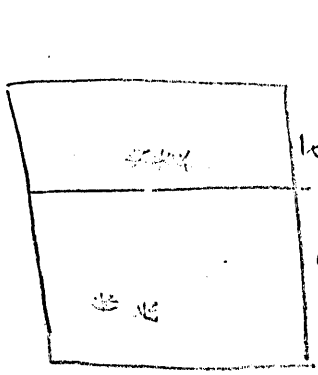
COMMUNICATION BLOCK
(COMKCBD)



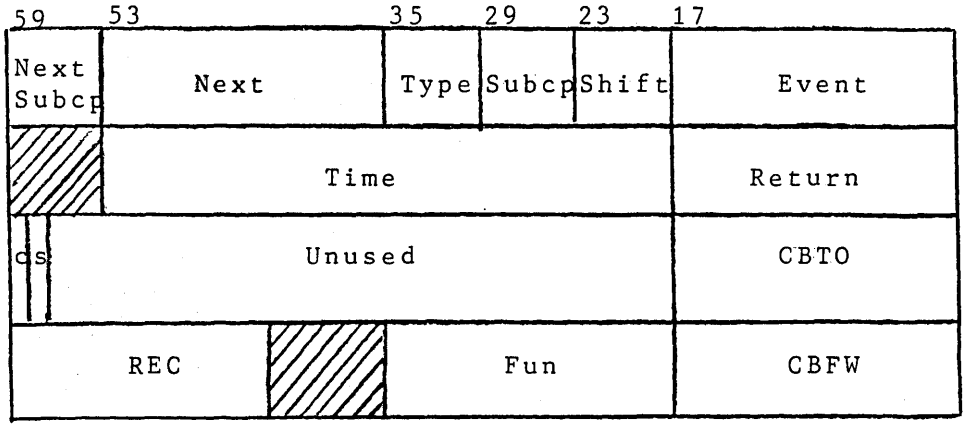
COMMUNICATION BLOCK SYSTEM HEADER







COMMUNICATION BLOCK SYSTEM HEADER
TAF WORK QUEUE ENTRY



```
***         *COMKCBD* GIVES THE DEFINITIONS FOR *TAF* COMMUNICATION
*           BLOCKS.  COMMUNICATION BLOCKS ARE USED TO PASS DATA
*           FROM *TAF* TO TASKS.  THIS DECK REQUIRES *COMKFLD*.
*
*           COMMUNICATION BLOCK SYSTEM HEADER.
*
*           THE COMMUNICATION BLOCK SYSTEM HEADER IS USED BY
*           *TAF* TO CONTROL PROCESSING RELATED TO A TRANSACTION.
*
*T  W1     12/  CP,1/I,1/R,1/M,1/A,1/C,1/L,24/  CBTS,18/ CBNC
*T, W2     18/     CBTO,3/RS,3/US,18/     CBTA,18/     CBFW
*T, W3     12/    1T,12/    2T,12/    3T,12/    4T,12/    5T
*T, W4     1/A,1/B,1/C,9/CBMC,18/LWA,12/ ,18/FWA
*T, W5     42/       QD,3/  ,3/OT,12/    QI
*T, CBCR   6/N,6/P,1/A,1/R,1/I,15/F,8/RFLG,4/U,18/E
*T, W7     6/NSUBCP,18/NEXT,6/TYPE,6/SUBCP,6/SHIFT,18/EVENT
*T, W8     6/ ,36/TIME,18/RETURN
*T, W9     1/C,1/S,40/  ,19/ CBTO
*T, W10    18/REC,6/ ,18/FUN,18/ CBFW
*
*           WORD 1.
*              CP     - CPU PRIORITY.
*              I      - 1, IF INITIAL LOAD REQUESTED.
*              R      - UNUSED.
*              M      - 1, IF MESSAGE SENT TO TERMINAL.
*              A      - 1, IF TRANSACTION CHAIN ABORTED.
*              C      - 1, IF TASK STARTED BY A *CALLRTN*.
*              L      - 1, IF TERMINAL TO REMAIND LOCK AFTER CEASE.
*              CBTS   - TRANSACTION SEQUENCE NUMBER.
*              CBNC   - FWA OF NEXT COMMUNICATION BLOCK.
*
*
*           WORD 2.
*              CBTO - TERMINAL ORDINAL.
*              RS   - TERMINAL *TAF* DATA MANAGER READ SECURITY.
*              WS   - TERMINAL *TAF* DATA MANAGER UPDATE SECURITY.
*              CBTA - FWA OF TERMINAL IN TERMINAL STATUS TABLE.
*              CBFW - COMMUNICATION BLOCK FWA.
*
*           WORD 3.
*              1T   - NEXT TASK SCHEDULE.
*              2T   - SECOND TASK IN CHAIN TO SCHEDULE.
*              3T   - THIRD TASK IN CHAIN TO SCHEDULE.
*              4T   - FOURTH TASK IN CHAIN TO SCHEDULE.
*              5T   - FIFTH TASK IN CHAIN TO SCHEDULE.
*
*           WORD 4.
*              A    - 1, IF VALID *DSDUMP* REQUEST.
*              B    - 1, IF DUMP EXCHANGE PACKAGE.
*              C    - 1, IF DUMP DATA BASE BUFFERS.
*              CBMC - NUMBER OF COMMUNICATION BLOCK FOR TRANSACTION.
*              LWA  - LAST WORD ADDRESS OF TASK DUMP.
*              FWA  - FIRST WORD ADDRESS OF TASK DUMP.
```

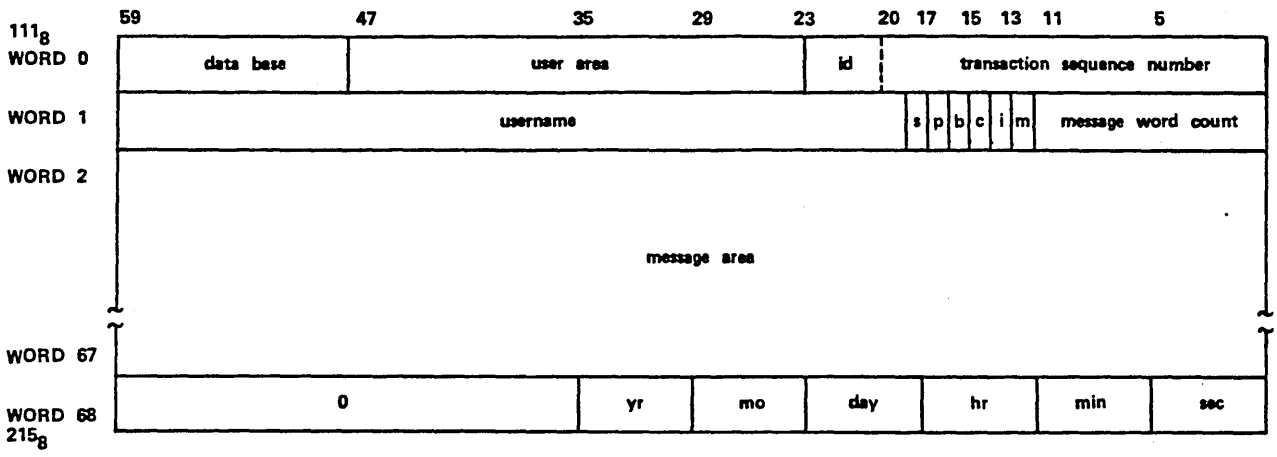
```
*          WORD 5.
*             QD    - QUEUE DESIGNATOR (SEE *K.DSDUMP*).
*             OT    - ORIGIN TYPE VALUE OF QUEUE DESTINATION.
*             QI    - QUEUE DESTINATION INDICATOR.
*
*          WORD 6 (CBCR).
*             N     - NEST LEVEL OF CURRENT TASK IF CALLED BY *CALLRTN*.
*             P     - SUBCONTROL POINT NUMBER OF LAST *CALLRTN* TASK.
*             A     - 1, IF *CALLRTN* TASK ABORTED.
*             R     - 1, IF *CALLRTN* TASK.
*             I     - 1, IF INITIAL TRANSFER TO C.B.
*             F     - RESERVED.
*             RFLG  - 1/L,1/O,1/S,1/C,1/D,1/M,1/T,1/B.
*             L     - 1, IF LOAD COMPLETE ON SCHEDULING TRANSACTION.
*             O     - 1, IF SYSTEM ORIGIN TRANSACTION.
*             S     - 1, IF TRANSACTION RESTARTED.
*             C     - 1, IF RECOVERABLE INPUT LOGGED.
*             D     - 1, IF *CDCS2* ALLOWED.
*             M     - 1, IF *CRM* ALLOWED.
*             T     - 1, IF RECOVERABLE TRANSACTION.
*             B     - 1, IF *BTRAN* TRANSACTION.
*             U     - USAGE BY DATA MANAGER.
*                   2 - *TOTAL* DATA MANAGER REQUESTS ALLOWED.
*                   4 - *AAM* FILE MANAGER REQUESTS ALLOWED.
*                   8 - *CDCS* CONNECTION INDICATOR.
*             E     - ENTRY ADDRESS IN ROLLOUT TABLE.
*
*          WORD 7 - WORD 10.
*          SEE TAF QUEUEING DESCRIPTIONS.
*                 (ONLY USED FOR QUEUEING DURING INPUT LOGGING)

*          DEFINITION OF BITS IN WORD *CBCR* INDICATING
*          DATA MANAGER USAGE.

 TOTDM     EQU     19          *TOTAL* DATA MANAGER
 AAMDM     EQU     20          *AAM* FILE MANAGER
 CDDM      EQU     21          *CDCS* CONNECTION INDICATOR
```

COMMUNICATION BLOCK SYSTEM HEADER FIELD DEFINITIONS

```
CBSD    FIELD  0,45,45    1, IF TRANSACTION SEND
CBAB    FIELD  0,44,44    1, IF TRANSACTION ABORT
CBLK    FIELD  0,42,42    1, IF TERMINAL REMAIND LOCK
CBTS    FIELD  0,41,18    TRANSACTION SEQUENCE NUMBER
CBIN    FIELD  0,47,47    1, IF INITIAL LOAD
CBNC    FIELD  0,17,0     NEXT COMMUNICATION BLOCK
CBTO    FIELD  1,59,42    TERMINAL ORDINAL
CBRS    FIELD  1,41,39    DATA MANAGER READ SECURITY
CBUS    FIELD  1,38,36    DATA MANAGER UPDATE SECURITY
CBTA    FIELD  1,35,18    TERMINAL ADDRESS
CBFW    FIELD  1,17,0     COMMUNICATION BLOCK FWA
CBTL    FIELD  2,59,0     TASK LIST
CBMC    FIELD  3,56,48    NUMBER OF COMMUNICATION BLOCKS
CBNL    FIELD  5,59,54    NEST LEVEL OF *CALLRTN* TASK
CBCP    FIELD  5,53,48    SUBCP NUMBER OF LAST *CALLRTN* TASK
CBRA    FIELD  5,47,47    1, IF *CALLRTN* TASK ABORTED
CBRF    FIELD  5,46,46    1, IF *CALLRTN* TASK
CBIT    FIELD  5,45,45    1, IF INITIAL TRANSFER TO C.B.
CBSC    FIELD  5,29,29    1, IF SCHEDULING COMPLETE
CBSO    FIELD  5,28,28    1, IF SYSTEM ORIGIN TRANSACTION
CBTR    FIELD  5,27,27    1, IF TRANSACTION RESTARTED
CBLC    FIELD  5,26,26    1, IF RECOVERABLE INPUT LOGGED
CBCS    FIELD  5,25,25    1, IF *CDCS2* ALLOWED
CBRM    FIELD  5,24,24    1, IF *CRM* ALLOWED
CBRT    FIELD  5,23,23    1, IF RECOVERABLE TRANSACTION
CBSB    FIELD  5,22,22    1, IF *BTRAN* TRANSACTION
CBQ1    FIELD  6,59,0     QUEUE ENTRY WORD 1
CBQ2    FIELD  7,59,0     QUEUE ENTRY WORD 2
CBQ3    FIELD  8,59,0     RECOVERY PARAMETERS
CBQ4    FIELD  9,59,0     RECOVERY PARAMETERS
```

```
*          *TAF* USES EVENT QUEUING TO PROCESS SOME OF ITS WORK.
*          THE FOLLOWING VARIABLES AND DEFINITIONS ARE USED FOR QUEUING.

*T  TAFQ   12/ ,6/LSUBCP,18/LAST,6/FSUBCP,18/FIRST
*
*          LSUBCP - LAST SUBCP ENTRY IN QUEUE.
*                   THIS IS ZERO IF NO SUBCP IS ASSOCAITED WITH *LAST*.
*          LAST   - FWA OF LAST ENTRY IN QUEUE.
*          FSUBCP - FIRST SUBCP IN QUEUE.
*                   THIS IS ZERO IF NO SUBCP IS ASSOCIATED WITH *FIRST*.
*          FIRST  - FWA OF NEXT ENTRY IN QUEUE.

 TAFQ      CON    0            *TAF* QUEUE POINTERS

*          EACH ENTRY IN THE QUEUE HAS THE FOLLOWING FORMAT -
*
*T         6/NSUBCP ,18/NEXT,6/TYPE,6/SUBCP,6/SHIFT,18/EVENT
*          6/ ,36/TIME,18/RETURN
*
*          NSUBCP - SUBCP OF NEXT ENTRY IN QUEUE.  ZERO IF NO
*                   SUBCP FOR *NEXT*.
*          NEXT   - FWA OF NEXT ENTRY IN QUEUE.  THIS IS ZERO FOR
*                   LAST ENTRY OF QUEUE.
*          TYPE   - TYPE OF EVENT.
*          SUBCP  - SUBCP OF QUEUED EVENT.  SUBCP IS ZERO IF NO
*                   SUBCP IS INVOLVED.
*          SHIFT -  SHIFT TO MOVE COMPLETE BIT TO BIT 59.
*                   A 1 INDICATES A COMPLETE EVENT.
*          EVENT  - FWA OF EVENT.
*          TIME   - TIME FOR EVENT TO COMPLETE IN MILLISECONDS.
*          RETURN - FWA OF PROCESSOR TO CALL WHEN EVENT IS COMPLETE.

*          *TAF* QUEUE WORK ENTRY FIELD DEFINITIONS.

 QWNS      FIELD  0,59,54      SUBCP OF NEXT ENTRY IN QUEUE
 QWNA      FIELD  0,53,36      FWA OF NEXT ENTRY RELATIVE TO NEXT SUBCP
 QWNT      FIELD  0,59,36      NEXT QUEUE ENTRY
 QWTY      FIELD  0,35,30      QUEUE TYPE
 QWSP      FIELD  0,29,24      SUBCP THAT NEEDS RESOURCE
 QWSH      FIELD  0,23,18      SHIFT TO POSITION COMPLETE BIT TO BIT 59
 QWEV      FIELD  0,17,0       FWA OF RESOURCE EVENT
 QWTM      FIELD  1,53,18      MILLISECONDS FOR EVENT TO COMPLETE
 QWPR      FIELD  1,17,0       QUEUE COMPLETE PROCESSOR

*          *TAF* AUTOMATIC RECOVERY QUEUE ENTRY DEFINITIONS.

 QRQ1      FIELD  0,59,0       *TAF* QUEUE WORD 1
 QRQ2      FIELD  1,59,0       *TAF* QUEUE WORD 2
 QRTC      FIELD  2,59,59      RECOVERY PROCESSING COMPLETE
 QRST      FIELD  2,58,58      RECOVERY STARTED COMPLETE
 QRTO      FIELD  2,17,0       TERMINAL ORDINAL FOR RECOVERY PROCESSING
 QREC      FIELD  3,59,0       RECOVERY REQUEST
```

COMMUNICATION BLOCK USER AREA

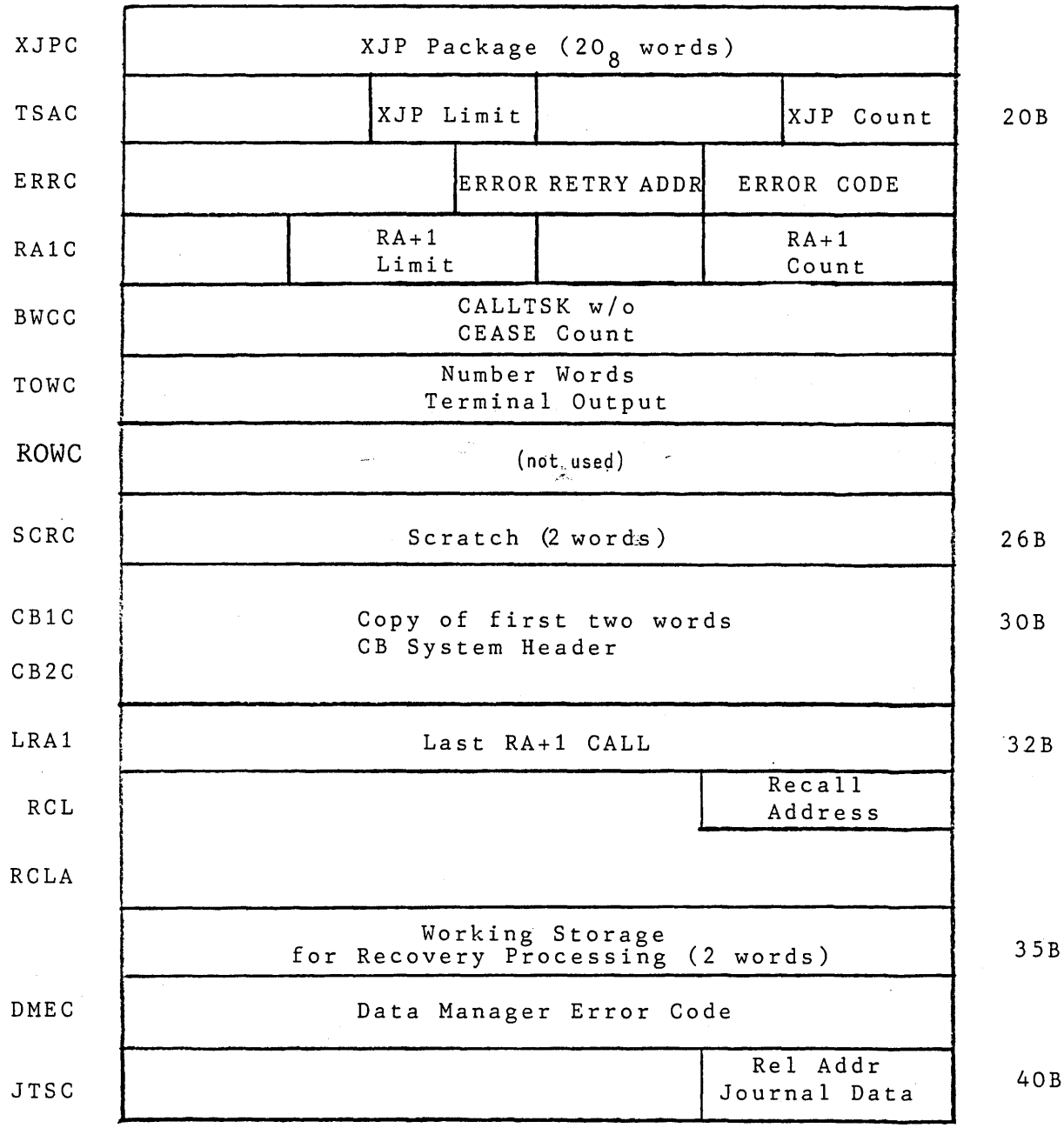


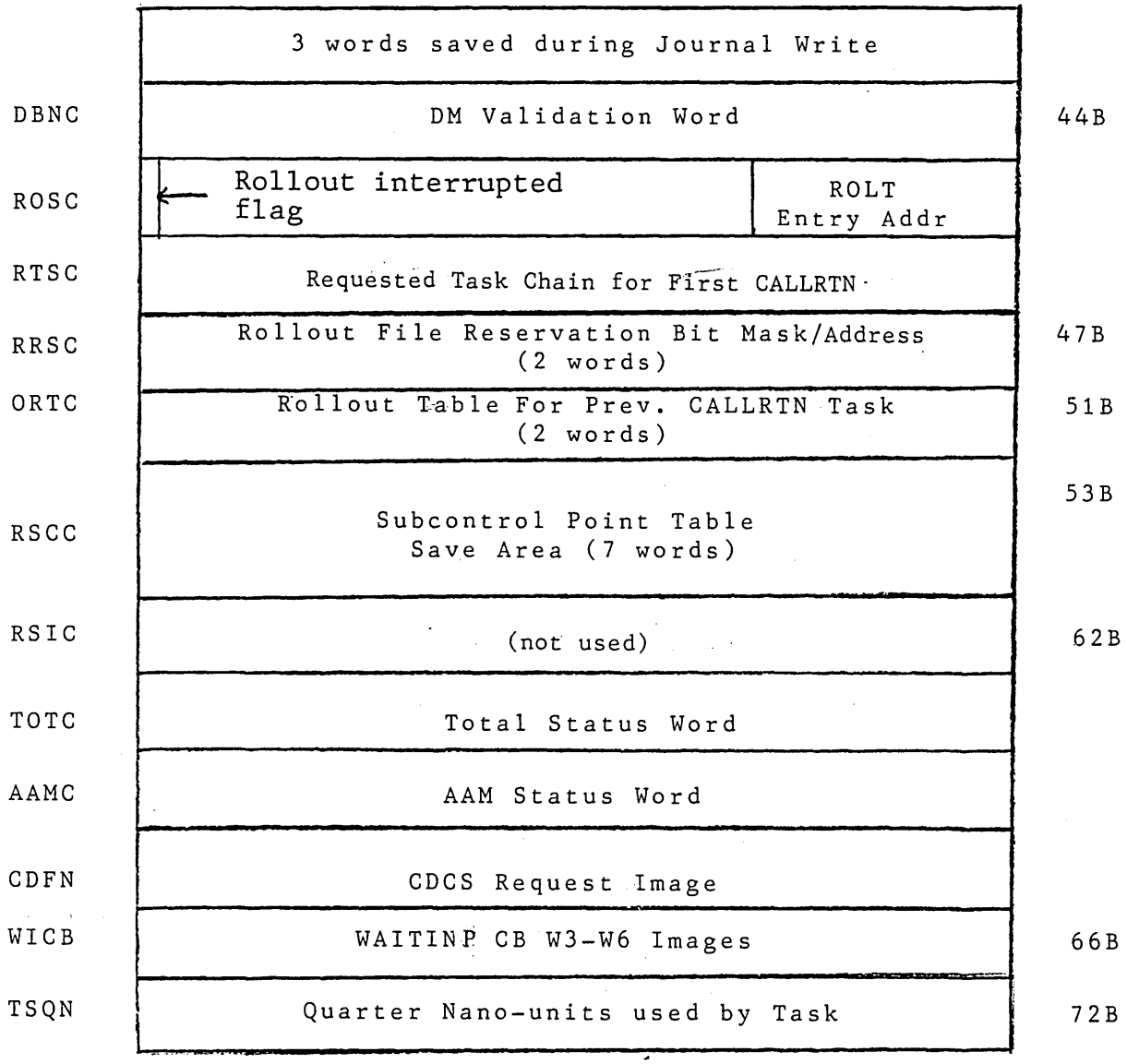
```
*         USER HEADER IS LOGICAL EXTENSION OF SYSTEM HEADER.

 CBWC     FIELD  11,11,0     WORD COUNT OF INPUT DATA

**        COMMUNICATIONS BLOCK USER HEADER.
*
*T  W1    12/      DB,24/        UA,24/        SEQ
*T, W2    42/ TN,1/S,1/P,1/B,1/C,1/I,1/M,12/WC
*
*         WORD 1.
*            DB    - DATA BASE TERMINAL IS VALIDATED TO USE.
*            UA    - USER AREA.
*            SEQ   - TRANSACTION SEQUENCE NUMBER.
*
*         WORD 2.
*            TN    - TERMINAL NAME.
*            S     - 1, IF SYSTEM ORIGIN TRANSACTION.
*            P     - 1, IF PARITY ERROR OCCURED ON TERMINAL INPUT.
*            B     - 1, IF BATCH INPUT.
*            C     - 1, IF CDCS ABORTED.
*            I     - 1, IF IDLE DOWN.
*            M     - 1, IF MULTIPLE INPUT.
*            WC    - WORD COUNT OF INPUT DATA.

*         DEFINITIONS OF COMMUNICATION BLOCK LENGTHS.

 CMBHL    EQU    10           SYSTEM HEADER LENGTH
 CBCR     EQU    5            WORD SIX OF SYSTEM HEADER
 CMBRL    EQU    2            USER HEADER LENGTH
 CBDL     EQU    57           DATA INPUT LENGTH
 CBUL     EQU    9            USER AREA LENGTH
          ERRNG  CBUL-4       IF *CBUL* NOT LARGE ENOUGH FOR RECOVERY
 CBTL     EQU    1            TIME/DATE LENGTH
 CMBL     EQU    CMBHL+CMBRL+CBDL+CBUL+CBTL  TOTAL LENGTH
 CBDCB    EQU    CBDL*60      LENGTH OF DATA INPUT IN BITS
 MAXINB   EQU    CBDCB*NCBC   MAXIMUM INPUT FOR TERMINAL IN BITS
 CBCH     EQU    CBDL-12-3    PARAMETER BLOCK HEADER (FOR *CDCS* ONLY)

*         DEFINITIONS FOR TASK COMMUNICATION BLOCK.

          LOC    111B
 SUAC     BSS    0            FWA OF USER AREA
 TRSQ     BSS    1            SEQUENCE NUMBER
 TNAM     BSS    1            TERMINAL/USER NAME
 TMSW     BSS    CBDL         DATA INPUT MESSAGE
 TDUA     BSS    CBUL         USER DEFINED AREA
 TIMD     BSS    CBTL         PACKED TIME/DATE
          LOC    *O

*         COMMUNICATION BLOCK USER HEADER DEFINITIONS.

 CBSY     FIELD  TNAM,17,17  1, IF SYSTEM ORIGIN TRANSACTION
 CBPR     FIELD  TNAM,16,16  1, IF PARITY ERROR
 CBBT     FIELD  TNAM,15,15  1, IF BATCH TRANSACTION
 CBCD     FIELD  TNAM,14,14  1, IF CDSC ABORT
 CBID     FIELD  TNAM,13,13  1, IF IDLE DOWN
```

TASK SYSTEM AREA - TSA
(COMKTSA)

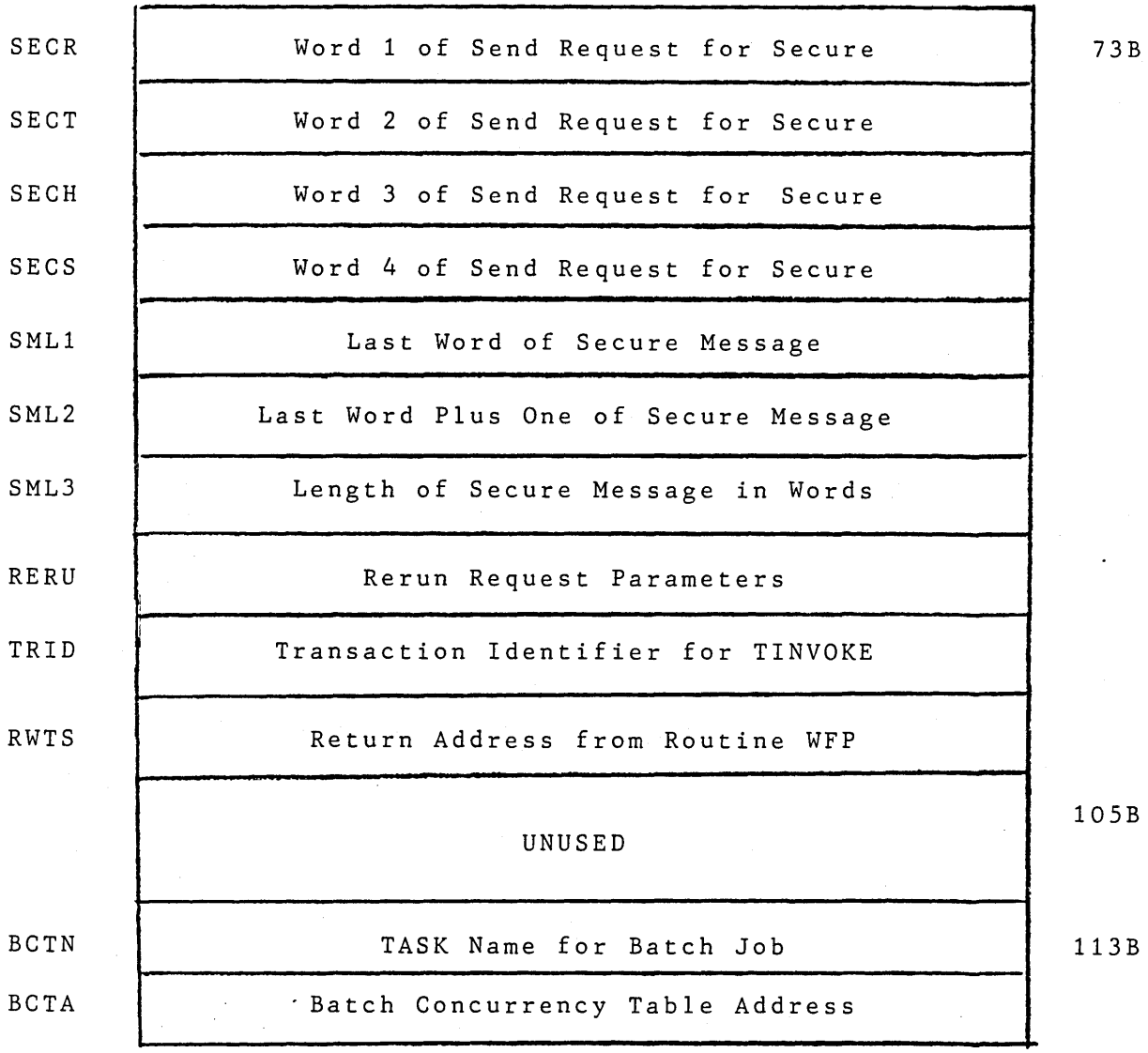
THE TASK SYSTEM AREA (TSA) IS LIKE THE CONTROL POINT AREA IN NOS. IT CONTAINS THE TASK'S EXCHANGE PACKAGE AND SYSTEM CONTROL INFORMATION. SINCE IT IS LOCATED IN THE TASK'S NEGATIVE FIELD LENGTH, THE EXECUTIVE IS THE ONLY ONE THAT CAN READ AND WRITE IT.

TASK SYSTEM AREA
(COMKTSA)




3 words saved during Journal Write
DBNC
DM Validation Word
44B
ROSC
Rollout interrupted flag
ROLT Entry Addr
RTSC
Requested Task Chain for First CALLRTN
RRSC
Rollout File Reservation Bit Mask/Address (2 words)
47B
ORTC
Rollout Table For Prev. CALLRTN Task (2 words)
51B
RSCC
Subcontrol Point Table Save Area (7 words)
53B
RSIC
(not used)
62B
TOTC
Total Status Word
AAMC
AAM Status Word
CDFN
CDCS Request Image
WICB
WAITINP CB W3-W6 Images
66B
TSQN
Quarter Nano-units used by Task
72B

TASK SYSTEM AREA (Continued)

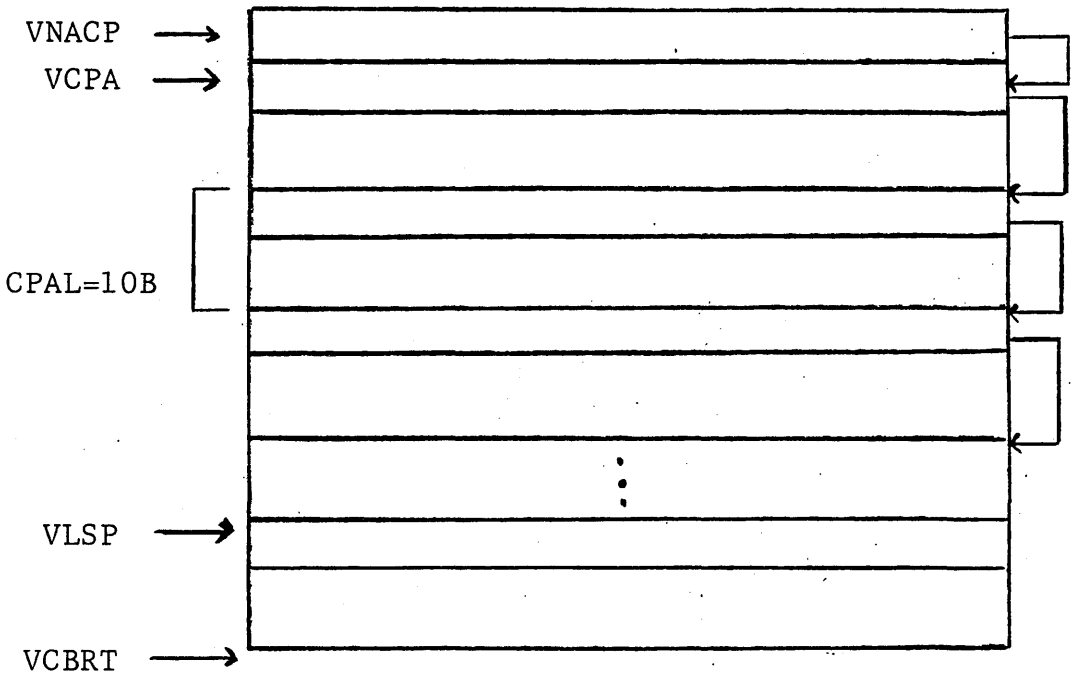


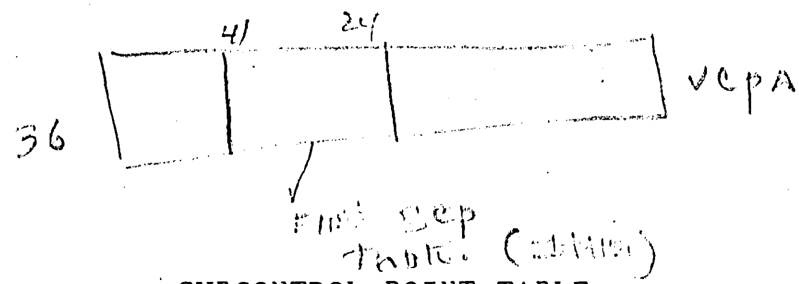
SUBCONTROL POINT TABLE - SUBCP
(COMKSCD)

THE SUBCONTROL POINT IS TO TAF WHAT CONTROL POINTS ARE TO NOS. A TASK THAT EXECUTES AT A SUBCP IS QUARANTEED MEMORY PROTECTION AND SYSTEM SECURITY. THE EXECUTIVE (TAF) PROCESSES ALL RA+1 REQUESTS WHICH GIVES TAF COMPLETE CONTROL OVER ALL TASKS. TAF CAN LOAD WHATEVER TASK IT WANTS TO AND DO ANY STORAGE MOVES WITHIN ITS FIELD LENGTH NECESSARY TO MAKE ROOM FOR THAT TASK. TAF CAN ALSO REQUEST ADDITIONAL FIELD LENGTH FROM THE SYSTEM IN ORDER TO LOAD A TASK. TAF DOES NOT HAVE THE RESTRICTION THAT THE FIELD LENGTH ASSOCIATED WITH A SUBCP HAS TO BE IN THE SAME ORDER AS THE SUBCP'S THEMSELVES, IE., SUBCP 3'S FL CAN BE AFTER SUBCP 5'S FL. BY EXAMINING THE SUBCP, THE ANALYST CAN DETERMINE WHICH COMMUNICATION BLOCK A TASK IS EXECUTING WITH AND WHICH ONES ARE WAITING TO EXECUTE.

LOW CORE POINTER WORD VCPA (36B) CONTAINS THE ADDRESS OF THE FIRST SUBCONTROL POINT TABLE (ITASK'S). WORD VNACP (45B) CONTAINS THE ADDRESS OF THE WORD THAT CONTAINS THE ADDRESS OF THE FIRST FREE SUBCONTROL POINT TABLE ENTRY.

SUBCONTROL POINT TABLE LINKAGE





3RD WORD 0-17 IS FORWARD LINK

SUBCONTROL POINT TABLE
(VCPA - COMKSCD)

entry ADDRESS

5/3 S

59							52	47		35	17	0
s	sr	l	cm	p	c	b		fc		fl	ra	
t	d	c	r	a	t	p	d	unused	rc	ep	cc	
n								unused		ls	ns	
x		i						unused		tcba	cba	

```
*          *TAF* SUBCP TABLE ENTRIES.
*
*T  W1     1/S,1/R,1/L,1/C,1/P,1/B,18/FC,18/FL,18/RA
*T, W2     1/S,1/D,1/C,1/R,1/A,1/T,1/P,1/D,13/  ,3/NC,18/EP,18/CC
*T, W3     12/      NM,12/         ,18/      LS,18/      NS
*T, WN     1/X,1/ ,1/ ,1/ ,1/ ,1/I, 18/ ,18/ TCBA,18/ CBA
*
*          WORD 1.
*             S   - 1, IF STOARGE MOVE LOCKED OUT.
*             R   - 1, IF SUBCP IS RELEASABLE.
*             L   - 1, IF TASK WILL REQUEST COMMUNICATION BLOCK.
*             C   - 1, IF IN *CMM* STATUS.
*             P   - 1, IF PAUSE BY MEMORY MANAGER (FUTURE USE).
*             B   - 1, IF IN USE BY BATCH CONCURRENCY.
*             FC  - FREE CORE AFTER SUBCP.
*             FL  - SUBCP FL.
*             RA  - SUBCP RA.
*
*          WORD 2.
*             S    - 1, IF SYSTEM TASK.
*             D    - 1, IF TASK CODE IS REUSABLE.
*             C    - 1, IF TASK IS CM RESIDENT.
*             R    - 1, IF TASK IN RECALL.
*             A    - 1, IF TASK IS TO BE ABORTED.
*             T    - 1, IF TASK IS TO BE TERMINATED.
*             P    - 1, IF CDCS ABORTED.
*             D    - 1, IF TASK IS DROPED BY THE OPERATOR.
*             NC   - NUMBER OF COMMUNICATION BLOCKS AT SUBCP.
*             EP   - FWA OF TASK ENTRY POINT.
*             CC   - FWA OF STATUS WORD FOR ACTIVE COMMUNICATION BLOCK.
*
*          WORD 3.
*             NM   - TASK DIRECTORY INDEX.
*             LS   - LAST SUBCP.
*             NS   - NEXT SUBCP.
*
*          WORD 4 TO 10B ARE USED FOR COMMUNICATION BLOCK STATUS WORDS.
*             X    - 1, IF COMMUNICATION BLOCK PRESENT.
*             I    - 1, IF INITIAL COMMUNICATION BLOCK LOAD.
*             TCBA - TASK COMMUNICATION BLOCK FWA (IF NOT DEFAULT).
*             CBA  - FWA OF COMMUNICATION BLOCK.

*          CPAL MUST BE EVEN MULTIPLE OF 10B.

 CPAL     EQU    10B          LENGTH OF A SUBCP ENTRY
 SCPAL    EQU    CPAL/10B+2   SHIFT TO CONVERT SUBCP FWA TO NUMBER
 CPAHL    EQU    3            LENGTH OF HEADER
 CPACL    EQU    CPAL-CPAHL   MAXIMUM NUMBER OF STATUS WORDS
```

SUBCONTROL POINT TABLE DEFINITIONS
(CONTINUED)

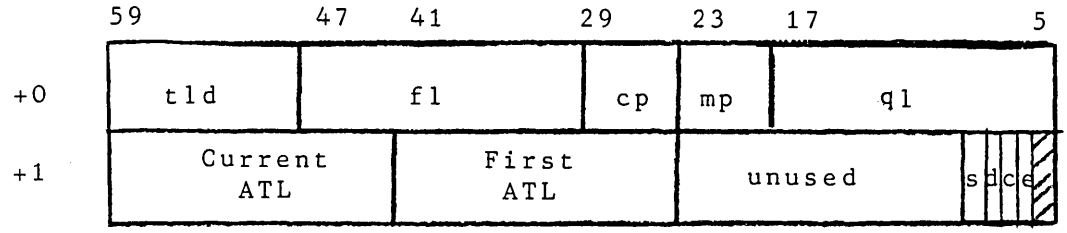
```
*         SUBCP FIELD DEFINITIONS.

 SCRL     FIELD  0,58,58     1, IF SPACE RELEASABLE
 SCSC     FIELD  0,57,57     1, IF TASK REQUESTED COMMUNICATION BLOCK
 SCCM     FIELD  0,56,56     1, IF IN *CMM* STATUS
 SCPU     FIELD  0,55,55     1, IF PAUSE FOR STORAGE REQUESTS
 SCBC     FIELD  0,54,54     1, IF IN USE BY BATCH CONCURRENCY
 SCFC     FIELD  0,53,36     FREE SPACE AFTER SUBCP
 SCFL     FIELD  0,35,18     TASK FIELD LENGTH
 SCRA     FIELD  0,17,0      TASK RA
 SCST     FIELD  1,59,59     1, IF SYSTEM TASK
 SCTR     FIELD  1,58,58     1, IF TASK IS REUSABLE
 SCCR     FIELD  1,57,57     1, IF TASK IS CM RESIDENT
 SCRC     FIELD  1,56,56     1, IF TASK IS IN RECALL
 SCTA     FIELD  1,55,55     1, IF TASK IS TO BE ABORTED
 SCTM     FIELD  1,54,54     1, IF TASK IS TO BE TERMINATED
 SCCD     FIELD  1,53,53     1, IF CDCS ABORTED
 SCDR     FIELD  1,52,52     1, IF DROPED BY OPERATOR
 SCNC     FIELD  1,38,36     NUMBER OF C.B.-S AT SUBCP
 SCEP     FIELD  1,35,18     FWA OF TASK ENTRY POINT
 SCCC     FIELD  1,17,0      FWA OF STATUS WORD FOR ACTIVE C.B.
 SCNM     FIELD  2,59,48     TASK DIRECTORY INDEX
 SCLS     FIELD  2,37,18     LAST SUBCP TABLE
```

REQUESTED TASK LIST - RTL

THE REQUESTED TASK LIST (RTL) CONTAINS ALL TASKS WAITING TO BE SCHEDULED, LIKE THE NOS INPUT AND ROLLOUT QUEUE. EVERYTIME THE SCHEDULER GETS CALLED, IT EVALUATES ALL TASKS IN THE RTL AND FINDS THE TASK WITH THE HIGHEST QUEUE PRIORITY AND SMALLEST FIELD LENGTH. IF SUFFICIENT FIELD LENGTH CAN BE OBTAINED TO LOAD THE TASK, THE REQUESTED TASK GETS 'MOVED' FROM THE RTL TO THE TASK LOAD STACK.

LOW CORE POINTER WORD VRTLW (53B) CONTAINS THE ADDRESS OF THE RESERVATION MASK WORD FOR THIS TABLE. THE RTL ENTRIES FOLLOW THE RESERVATION WORD IN CORE.

REQUESTED TASK LIST - RTL



If task rollin request

REQUESTED TASK LIST - RTL

```
**        RTL - REQUESTED TASK LIST.
*
*T  W1    12/   NAME,18/    FL,6/  CP,6/  MP,18/    L
*T, W2    18/      CR,18/      1S,18/ ,1/S,1/D,1/C,1/E,2/
*T, W2    18/      CR,18/      1S,6/  ,18/      RE
*
*         WORD 1.
*            NAME - TASK DIRECTORY INDEX.
*            FL   - FIELD LENGTH.
*            CP   - CURRENT PRIORTY.
*            MP   - MAXIMUM PRIORTY (FUTURE USE).
*            L    - QUEUE LENGTH LIMIT.
*
*         WORD 2 (IF NOT FOR TASK ROLLIN REQUEST).
*            CR   - CURRENT *ATL* ENTRY.
*            1S   - FIRST *ATL* ENTRY.
*            S    - SYSTEM TASK.
*            D    - NON DESTRUCTIVE CODE.
*            C    - CM RESIDENT.
*            E    - ECS RESIDENT.
*
*         WORD 2 (IF FOR A TASK ROLLIN REQUEST).
*            CR   - CURRENT *ATL* ENTRY.
*            1S   - FIRST *ATL* ENTRY.
*            RE   - ROLLOUT TABLE ENTRY ADDRESS OF TASK TO ROLL IN.


 RTLLE    EQU    2           LENGTH OF A *RTL* ENTRY
 NRTL     EQU    40          NUMBER OF *RTL* ENTRIES (MUST BE .LE. 47).

 RTLW     ALLOC  NRTL,47     RESERVATION WORD FOR *RTL*
 RTL      BSSZ   NRTL*RTLLE  *RTL* ENTRIES
 RTLL     EQU    *-RTL       LENGTH OF *RTL*
```

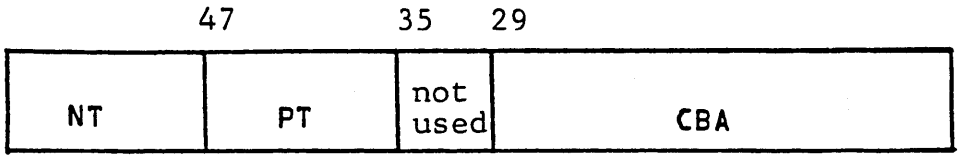
ACTIVE TRANSACTION LIST - ATL

THE ACTIVE TRANSACTION LIST (ATL) CORRELATES THE RTL ENTRIES TO THE COMMUNICATION BLOCKS THAT ARE WAITING FOR THOSE TASKS. WHEN A TASK FROM THE RTL GETS LOADED, TAF CHAINS THROUGH THE ATL'S TO FIND ALL THE CB'S QUEUED ON THAT TASK. THERE CAN BE A MAXIMUM OF QL (QUEUE LIMIT DEFINED FOR THE TASK) ATL ENTRIES QUEUED ON ONE RTL.

LOW CORE POINTER WORD VATL (16B) CONTAINS THE ADDRESS OF THE FIRST ATL ENTRY.

```
**          ATL - ACTIVE TRANSACTION LIST.
*
*T          12/    NT,12/     PT,6/    ,30/        CBA
*
*           NT     - NEXT TASK IN QUEUE CHAIN (BIASED BY +1).
*           PT     - PREVIOUS TASK IN QUEUE CHAIN (BIASED BY +1).
*           CBA    - ADDRESS OF COMMUNICATIONS BLOCK.

 ATLL       EQU    1           WORDS PER *ATL* ENTRY
```

ACTIVE TRANSACTION LIST - ATL



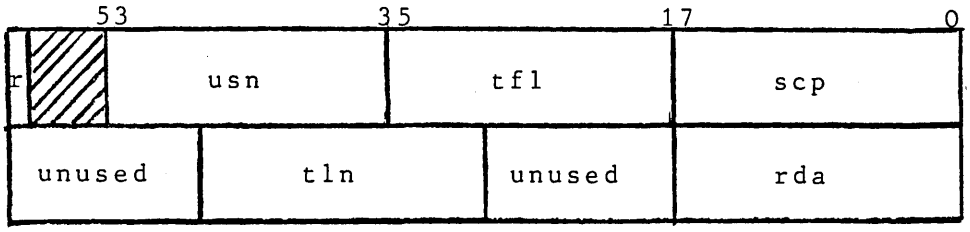
TASK LOAD STACK - TLS

IF THE TASK SELECTED FROM THE RTL NEEDS TO BE LOADED (IE., NOT IN CORE AND REUSABLE), ROUTINE RCP IS CALLED TO FIND THE FIELD LENGTH REQUIRED AND A FREE SUBCP TABLE ENTRY. IF THE SUBCP TABLE ENTRY AND FIELD LENGTH ARE AVAILABLE, TAF MOVES ALL CB'S LINKED THROUGH THE ATL'S TO THAT TASK (RTL) INTO THE SUBCP TABLE ENTRY. TAF THEN GENERATES AN ENTRY IN THE TLS TO LOAD THE TASK. IT IS FROM THIS TABLE THAT TAF ACTUALLY DOES THE I/O TO LOAD A TASK FROM EITHER DISK OR EXTENDED MEMORY.

THERE IS NO LOW CORE POINTER WORD FOR THE TLS. THE TLS IS HARD CODED BEGINNING AT THE TAG CCC.

```
**        TLS - TASK LOAD STACK.
*
*T  W1    1/R,5/  ,18/      USN,18/      TFL,18/      SCP
*T, W2    30/      TLN ,30/      RDA
*
*         WORD 1.
*            R    - 1, IF TASK ROLLIN.
*            USN  - ADDRESS (-2) OF USER NUMBER FOR TASK LIBRARY.
*            TFL  - TASK FIELD LENGTH.
*            SCP  - FWA OF SUBCP TABLE ENTRY.
*
*         WORD 2.
*            TLN  - ADDRESS OF TASK LIBRARY NAME.
*            RDA  - RANDOM DISK ADDRESS OF TASK.


 CCC      BSSZ   5*2+1       FIVE ENTRIES + ZERO WORD TERMINATOR
 LTLRE    EQU    *-3         FWA OF LAST TASK LOAD REQUEST STACK ENTRY
```

TASK LOAD STACK - TLS



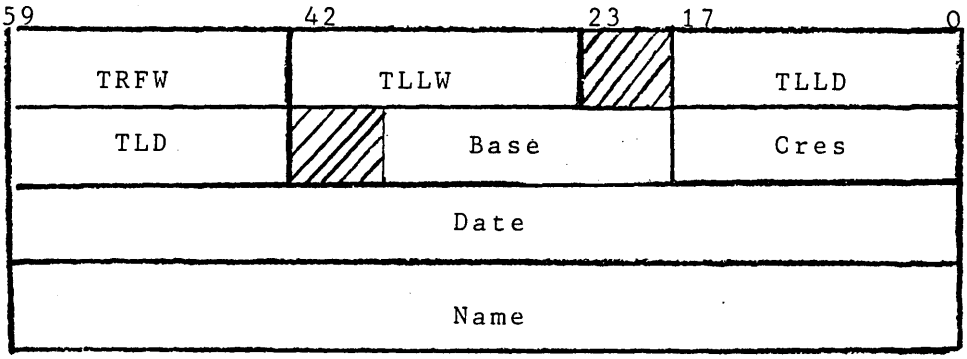
TASK LIBRARY DIRECTORY
(COMKTLD)

THE TASK LIBRARY DIRECTORY (TLD) CONTAINS ALL CONTROL INFORMATION FOR EACH TASK. TAF ATTACHES ALL TASK LIBRARIES AND READS THE DIRECTORIES INTO ITS FIELD LENGTH, LEAVING SPACE FOR TLDL (10 IS DEFAULT) EXTRA TASKS TO BE ADDED DURING PRODUCTION. EACH TIME A TASK IS REQUESTED, THE TLD ASSOCIATED WITH THE DATA BASE FOR THE TERMINAL IS SEARCHED TO FIND THE CALLED TASK. IF THE TASK IS NOT FOUND, TAF SEARCHES THE SYSTEM TLD. IF THE TASK IS IN NEITHER OF THOSE TLD'S, THE TRANSACTION IS ABORTED. FROM THE TLD ENTRY, TAF KNOWS WHETHER TO LOAD THE TASK FROM DISK OR EXTENDED MEMORY AND AT WHAT POSITION TO INITIATE THE READ.

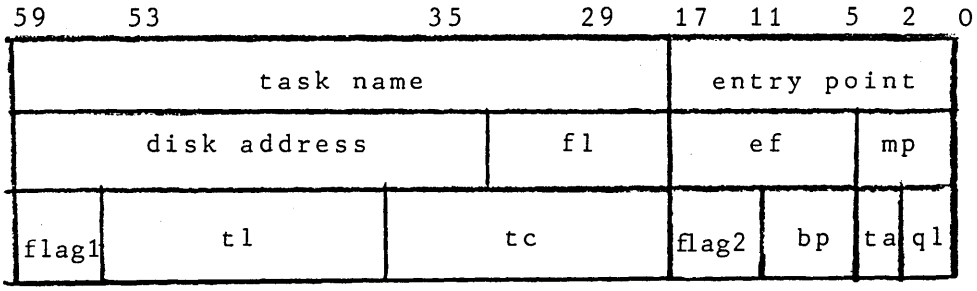
LOW CORE POINTER WORD VTLD (22B) CONTAINS THE ADDRESS OF THE FIRST TLD ENTRY IN THE FIRST TASK LIBRARY (TASKLIB).

TASK LIBRARY DIRECTORY
(COMKTLD)

Header



Entry



TASK LIBRARY DIRECTORY - TLD
(COMKTLD)

```
**        TASK DIRECTORY HEADER IN *TAF*.
*
*T VTLD-4 18/ TRFWA,18/ TLLW, 6/ ,18/ TLLD
*T,VTLD-3 18/ TLDN, 6/ ,18/ BASE, 18/ CRES
*T,VTLD-2 60/ DATE
*T,VTLD-1 60/ NAME
*
*         VTLD-4.
*                TRFWA = FWA OF CORRESPONDING TRANSACTION DIRECTORY.
*                TLLW  = LWA OF LAST TLD (EXISTS IN FIRST TLD ONLY).
*                TLLD  = LENGTH OF TASK DIRECTORY + EXPANSION AREA.
*
*         VTLD-3.
*                TLDN  = RECORD NAME, *TLD*.
*                BASE  = LENGTH OF SORTED PART OF TASK DIRECTORY.
*                CRES  = NUMBER OF CM RESIDENT TASKS.
*
*         VTLD-2.
*                DATE  = DATE OF LAST MODIFICATION.  YY/MM/DD.
*
*         VTLD-1.
*                NAME  = LIBRARY FILE NAME.
*E


 TRFW     FIELD  -4,59,42    FWA OF CORRESPONDING *TRD*
 TLLW     FIELD  -4,41,36    LWA OF LAST *TLD* ENTRY
 TLLT     FIELD  -4,17,0     LENGTH OF TASK DIRECTORY + EXPANSION AREA
 TLLB     FIELD  -3,35,18    LENGTH OF BASE (SORTED) DIRECTORY
 TLLN     FIELD  -1,59,0     LIBRARY FILE NAME

 TLDLH    EQU    4           LENGTH OF TASK LIBRARY HEADER
```

TASK LIBRARY DIRECTORY - TLD
(COMKTLD)

```
**        TASK DIRECTORY ENTRY.
*
*T  W1    42/ NAME,18/ EP
*T, W2    30/ DA,12/ FL,12/ EF,6/ MP
*T, W3    6/ FLG1,18/ TL,18/ TC,6/ FLG2,6/ BP,3/ TA,3/ QL
*
*         WORD 1.
*                NAME - TASK NAME.
*                EP   - ENTRY POINT.
*
*         WORD 2.
*                DA   - DISK ADDRESS.
*                FL   - FIELD LENGTH/100B REQUIRED BY TASK.
*                EF   - EXPANDABLE FIELD LENGTH/100B FOR TASK.
*                MP   - MAXIMUM PRIORITY.
*
*         WORD 3.
*                FLG1 - 1/S,1/D,1/C,1/E,1/O,1/L.
*                S  - SYSTEM TASK.
*                D  - DESTRUCTIVE CODE.
*                C  - CM RESIDENT.
*                E  - ECS RESIDENT LIBRARY COPY.
*                O  - TASK TURNED OFF.
*                L  - TASK LOGICALLY DELETED. (*DL* PARAMETER)
*                TL - NUMBER OF TIMES TASK WAS LOADED.
*                TC - NUMBER OF TIMES TASK WAS CALLED.
*                FLG2 - 1/B,1/Q,1/R,3/0.
*                B  - SOLICITED C.B. LOAD WAS REQUESTED.
*                     (*SC* PARAMETER).
*                Q  - QUEUING FORCED FOR ADDITIONAL ACTIVE
*                     COPIES.
*                R  - REDUCE FL. (CM RESIDENT TASK ONLY).
*                BP   - BASE PRIORITY.
*                TA   - TASK ACTIVE STATUS. THIS FIELD ONLY APPLIES
*                       TO TASKS WHICH HAVE THE Q-ATTRIBUTE DECLARED.
*                       IN ADDITION IT DOES NOT APPLY TO CM RESIDENT
*                       TASKS.
*                QL   - QUEUE LENGTH LIMIT.
```

```
**        TASK DIRECTORY ENTRY (CONTINUED)

 TLTN     FIELD  0,59,18     TASK NAME
 TLEP     FIELD  0,17,0      ENTRY POINT

 TLDA     FIELD  1,59,30     DISK ADDRESS
 TLFL     FIELD  1,29,18     FIELD LENGTH REQUIRED BY TASK/100B
 TLEF     FIELD  1,17,6      EXPANDABLE FIELD LENGTH FOR TASK/100B
 TLMP     FIELD  1,5,0       MAXIMUM PRIORITY

 TLST     FIELD  2,59,59     SYSTEM TASK
 TLND     FIELD  2,58,58     DESTRUCTIVE CODE
 TLCM     FIELD  2,57,57     CM RESIDENT
 TLEC     FIELD  2,56,56     ECS RESIDENT LIBRARY COPY
 TLTO     FIELD  2,55,55     TASK TURNED OFF BY OPERATOR
 TLDL     FIELD  2,54,54     TASK LOGICALLY DELETED
 TLTL     FIELD  2,53,36     NUMBER OF TIMES TASK WAS LOADED
 TLTC     FIELD  2,35,18     NUMBER OF TIMES TASK WAS CALLED
 TLSC     FIELD  2,17,17     SOLICITED COMMUNICATION BLOCK LOAD
 TLQU     FIELD  2,16,16     QUEUING FORCED FOR ADDITIONAL COPIES
 TLRE     FIELD  2,15,15     REDUCE FL
 TLX1     FIELD  2,14,12     RESERVED FOR CDC
 TLBP     FIELD  2,11,6      BASE PRIORITY
 TLTA     FIELD  2,5,3       TASK ACTIVE STATUS
 TLQL     FIELD  2,2,0       QUEUE LENGTH LIMIT

*         TO CHANGE THE LENGTH OF A TASK LIBRARY DIRECTORY
*         ENTRY, IT IS NOT SUFFICIENT JUST TO CHANGE THE VALUE
*         OF THE NEXT SYMBOL.  IT IS ALSO NECESSARY TO CHANGE
*         SOME CODE IN THE AFFECTED DECKS.

 TLDLE    EQU    3           LENGTH OF A TLD ENTRY
 TLDMT    EQU    600         MAXIMUM TASKS IN LIBRARY
 TLDMN    EQU    10          MAXIMUM ADDITIONAL TASKS ON *TT* RUN
 TLDL     EQU    TLDMN*TLDLE SPACE NEEDED FOR ADDITIONAL TASKS
```

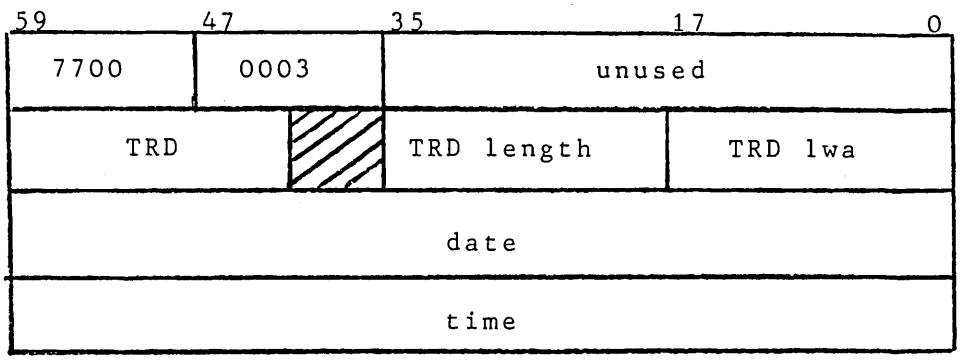
TRANSACTION DIRECTORY
(COMKTLD)

THIS DIRECTORY IS CREATED AND MAINTAINED BY LIBTASK. IT IS READ INTO TAF'S FIELD LENGTH DURING INITIALIZATION. IT CONTAINS INFORMATION RELATIVE TO A TRANSACTION LIKE THE NAME, THE TLD OFFSETS FOR THE TASKS THAT MAKE UP THE TRANSACTION, AND WHICH DATA MANAGER THIS TRANSACTION USES.

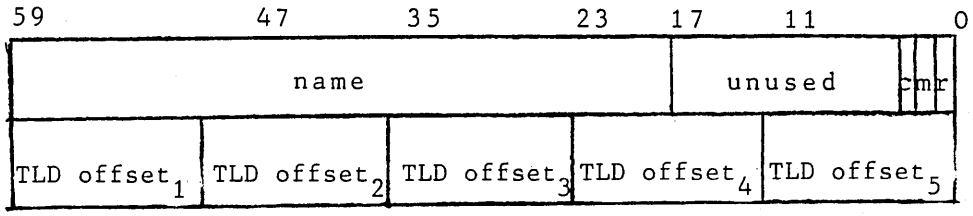
THE ADDRESS OF THE TRD OF A TASK LIBRARY CAN BE FOUND IN THE FIRST WORD OF THE TLD HEADER FOR THE PARTICULAR TASK LIBRARY.

TRANSACTION DIRECTORY
(COMKTLD)

Header



Entry



TRANSACTION DIRECTORY - TRD
(COMKTLD)

```
**          TRANSACTION DIRECTORY HEADER.
*
*T TRFW-4 12/ 7700,12/ ,36/
*T,TRFW-3 18/ TRDN,6/ ,18/ TRDL,18/ TRLW
*T,TRFW-2 60/ DATE
*T,TRFW-1 60/ TIME
*
*           TRFW-3.
*                   TRDN = RECORD NAME, *TRD*.
*                   TRDL = LENGTH OF DIRECTORY IN *LIBTASK*.
*                   TRLW = LWA OF DIRECTORY.
*
*           TRFW-2.
*                   DATE = LAST MODIFICATION DATE.  YY/MM/DD.
*
*           TRFW-1.
*                   TIME = LAST MODIFICATION TIME.  HH.MM.SS.
*E


 TDDN      FIELD  -3,59,42     RECORD NAME
 TDLD      FIELD  -3,35,18     *TRD* LENGTH
 TDLW      FIELD  -3,17,0      LWA OF DIRECTORY
 TDDT      FIELD  -2,59,0      MODIFICATION DATE
 TDTM      FIELD  -1,59,0      MODIFICATION TIME

 TRDLH     EQU    4            LENGTH OF *TRD* HEADER
```

TRANSACTION DIRECTORY - TRD

```
**          TRANSACTION DIRECTORY ENTRY.
*
*           (IN *LIBTASK*).
*T  W1      42/ TRNAME,5/ ,7/ ,1/D,2/ ,1/C,1/M,1/R
*T, W2      42/ TSK1,18/0
*T, W3      42/ TSK2,18/0
*T, W4      42/ TSK3,18/0
*T, W5      42/ TSK4,18/0
*T, W6      42/ TSK5,18/0
*
*           WORD 1.
*                 TRNAME - TRANSACTION UNIT NAME.
*                 D - 1 IF TRANSACTION LOGICALLY DELETED (IN LIBTASK).
*                 C - 1 IF TRANSACTION MAY USE *CDCS*.
*                 M - 1 IF TRANSACTION MAY USE *CRM*.
*                 R - 1 IF TRANSACTION IS RECOVERABLE.
*           WORD 2.
*                 TSK1    - FIRST TASK TO SCHEDULE.
*           WORD 3.
*                 TSK2    - SECOND TASK TO SCHEDULE.
*           WORD 4.
*                 TSK3    - THIRD TASK TO SCHEDULE.
*           WORD 5.
*                 TSK4    - FOURTH TASK TO SCHEDULE.
*           WORD 6.
*                 TSK5    - FIFTH TASK TO SCHEDULE.
*
*           (IN *TAF*).
*T  W1      42/ TRNAME,1/ ,1/ ,1/ ,1/ ,1/ ,8/ ,2/ ,1/C,1/M,1/R
*T, W2      12/ IND1,12/ IND2,12/ IND3,12/ IND4, 12/ IND5
*
*           WORD 1.
*                 SEE ABOVE.
*
*           WORD 2.
*                 IND1 - IND5, OFFSETS FROM START OF TLD (VTLD)
*                              CORRESPONDING TO TASKS TSK1 - TSK5
*                              ABOVE.
```

TERMINAL STATUS TABLE
(COMKTST)

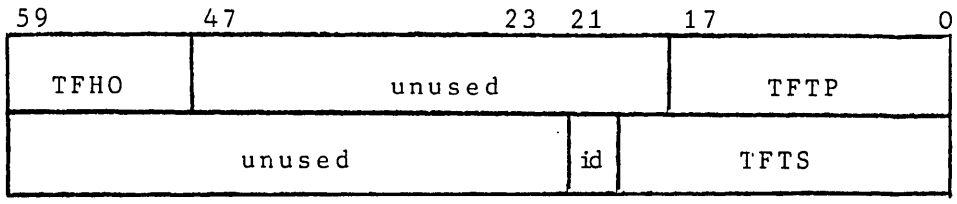
TWO TABLES ARE USED TO CONTROL USERS/TERMINALS, THE TERMINAL FILE TABLE (TTFT), AND THE TERMINAL STATUS TABLE (TST). THE TTFT INDICATES THE NETWORK FILE CONTAINING THE USER/TERMINAL NAME. IT IS ALSO THE TABLE THAT ASSOCIATES USERS/TERMINALS IN THE TST WITH THE CORRECT COMMUNICATION RECOVERY FILE (CRF). THIS TABLE IS USED TO FIND THE CORRECT CRF FROM WHICH TO DO I/O FOR RECOVERY REQUESTS. THE TST CONTAINS STATUS INFORMATION FOR EACH USER/TERMINAL. THERE ARE AS MANY ENTRIES IN THE TST AS TRANSACTION TERMINAL DEFINITIONS IN ALL THE NETWORK FILES USED BY TAF.

THUS, THE TST MAY BE LOGICALLY DIVIDED INTO SUBSECTIONS, EACH REFLECTING THE RESPECTIVE NETWORK FILE. THE TERMINAL FILE TABLE DEFINES THE LENGTH OF EACH SUBSECTION.

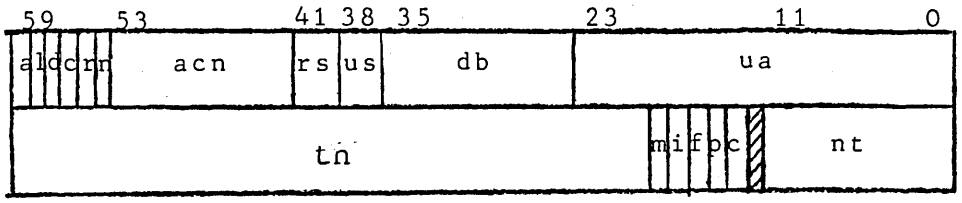
LOW CORE POINTER VTST CONTAINS THE ADDRESS OF THE FIRST ENTRY IN THE TST. THE TTFT IMMEDIATELY PRECEEDES THE TST.

TERMINAL STATUS TABLE
(COMKTST)

TTFT Entry



TST Entry



TERMINAL FILE TABLE - TTFT
(COMKTST)

```
*          *TTFT* - TERMINAL FILE TABLE.
*
*T  W1     12/ TFHO,30/ ,18/ TFTP
*T, W2     36/ ,3/TFID,21/TFTS
*
*          WORD 1.
*                 TFHO - HIGHEST *TST* ORDINAL FOR NETWORK FILE(N).
*                 TFTP - FWA OF *TAF* RECOVERY FILE TABLE.
*                        0, IF *IPTAR* EQUALS ZERO.
*          WORD 2.
*                 TFID - IDENTIFIER FOR NETWORK FILE.
*                 TFTS - CURRENT TRANSACTION SEQUENCE NUMBER.
*
*          IF THERE ARE LESS THAN 8 NETWORK FILES IN USE THE HEADER IS
*          ZERO FILLED.  THE HEADER HAS NO USE FOR *TAFTS*.

*          *TTFT* - TERMINAL FILE TABLE DEFINITIONS.

 TFHO      FIELD  0,59,48      HIGHEST *TST* ORDINAL FOR NETWORK FILE
 TFTP      FIELD  0,17,0       POINTER TO *TTRF* TABLE FOR NETWORK FILE
 TFID      FIELD  1,23,21      NETWORK FILE IDENTIFIER
 TFTS      FIELD  1,20,0       CURRENT TRANSACTION SEQUENCE NUMBER

 TTFTE     EQU    TFTSW+1      *TTFT* ENTRY LENGTH
 TTFTL     EQU    TTFTE*8      *TTFT* TABLE LENGTH
```

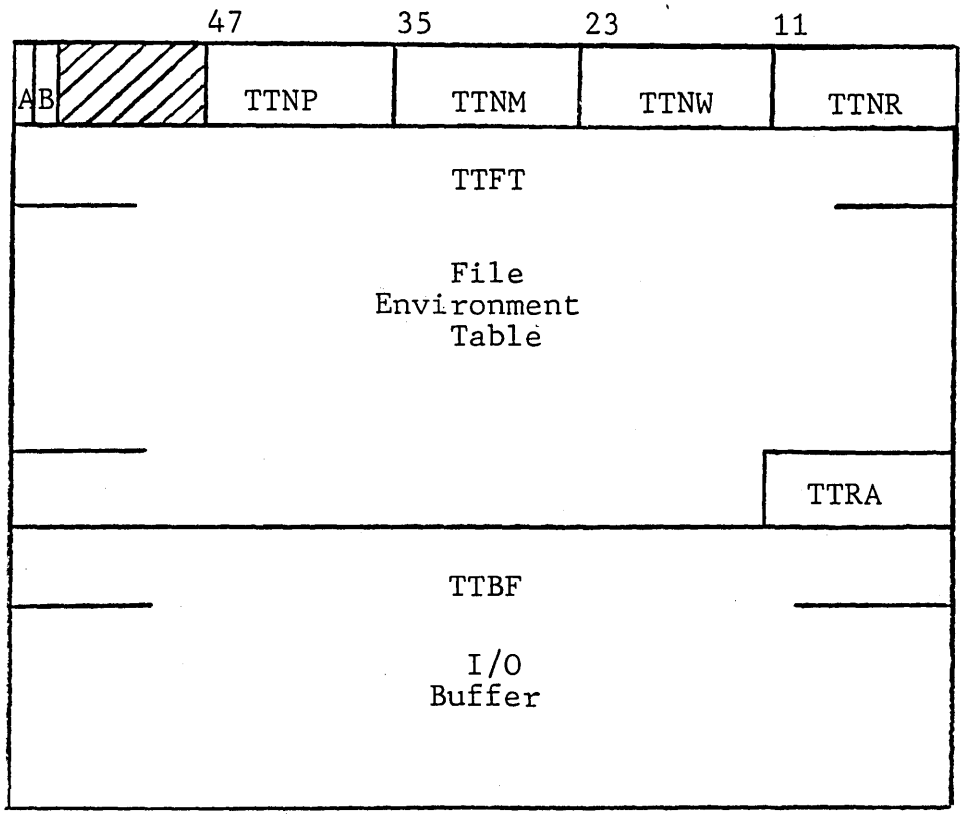
```
*          *TST* - TERMINAL STATUS TABLE.
*
*T  W1     1/A,1/L,1/D,1/C,1/R,1/N,12/ACN,3/RS,3/US,12/DB,24/UA
*T, W2     42/  TN,1/M,1/I,1/F,1/P,1/C,1/ ,12/ NT
*
*          WORD 1.
*                 A     - 1, IF USER ACTIVE.
*                 L     - 1, IF TERMINAL LOGGED IN.
*                 D     - 1, IF TERMINAL DOWN.
*                 O     - 1, IF TERMINAL ON/OFF.
*                 C     - 1, IF CON/REQ SUPERVISORY MESSAGE RECEIVED.
*                 R     - 1, IF AUTOMATIC RECOVERY REQUIRED.
*                 N     - 0, IF NON RECOVERABLE TRANSACTION RUN.
*                 RS    - DATA BASE READ SECURITY LEVEL (UNUSED).
*                 US    - DATA BASE UPDATE SECURITY LEVEL (UNUSED).
*                 DB    - DATA BASE NAME.
*                 UA    - USER AREA.
*                 ACN   - APPLICATION CONNECTION NUMBER.
*
*          WORD 2.
*                 TN    - USER/TERMINAL NAME.
*                 M     - 1, IF MULTIPLE BLOCK INPUT.
*                 I     - 1, IF INPUT WANTED.
*                 F     - 1, IF LAST MESSAGE BLOCK SENT.
*                 P     - 1, IF CONNECTION IS TO BE POSTPONED.
*                 C     - 1, IF CDCS ABORT.
*                 NT    - NUMBER OF TRANSACTIONS.

*          *TST* - TERMINAL STATUS TABLE DEFINITIONS.

 TSAU     FIELD  0,59,59      1, IF ACTIVE USER
 TSLI     FIELD  0,58,58      1, IF TERMINAL LOGGED IN
 TSTD     FIELD  0,57,57      1, IF TERMINAL DOWN
 TSTO     FIELD  0,56,56      1, IF TERMINAL OFF
 TSCR     FIELD  0,56,56      1, IF CON/REQ MESSAGE RECEIVED.
 TSAR     FIELD  0,55,55      1, IF AUTOMATIC RECOVERY REQUIRED
 TSNR     FIELD  0,54,54      0, IF NON RECOVERABLE TRANSACTION RUN
 TSCN     FIELD  0,53,42      APPLICATION CONNECTION NUMBER
 TSRS     FIELD  0,41,39      DATA BASE READ SECURITY LEVEL
 TSUS     FIELD  0,38,36      DATA BASE UPDATE SECURITY LEVEL
 TSDB     FIELD  0,35,24      DATA BASE NAME
 TSUA     FIELD  0,23,0       USER AREA
 TSRB     FIELD  0,0,0        1, IF USER RECOVERY BIT
 TSTN     FIELD  1,59,18      TERMINAL NAME
 TSMB     FIELD  1,17,17      1, IF MULTIPLE BLOCK INPUT
 TSIW     FIELD  1,16,16      1, IF INPUT WANTED
 TSMS     FIELD  1,15,15      1, IF LAST MESSAGE BLOCK SENT
 TSCP     FIELD  1,14,14      1, IF CONNECTION IS TO BE POSTPONED
 TSCD     FIELD  1,13,13      1, IF CDCS ABORT
 TSTC     FIELD  1,11,0       TRANSACTION COUNT

 TSTLLE   EQU    TSTCW+1      *TST* ENTRY LENGTH
```

COMMUNICATION RECOVERY FILE TABLE - TTRF
(COMKTRF)

THE COMMUNICATION RECOVERY FILE TABLE DEFINES THE ATTRIBUTES OF THE COMMUNICATION RECOVERY FILE (CRF), AND CONTAINS THE FET FROM WHICH I/O IS DONE ON THE FILE. ATTRIBUTES INCLUDE MAXIMUM SIZE AND NUMBER OF USER (RPUT/RGET) MESSAGES FOR THIS FILE.

THIS TABLE IS POINTED TO BY THE TERMINAL FILE TABLE (TTFT).

COMMUNICATION RECOVERY FILE TABLE
(COMKTRF)



COMMUNICATION RECOVERY FILE TABLE - TTRF
(COMKTRF)

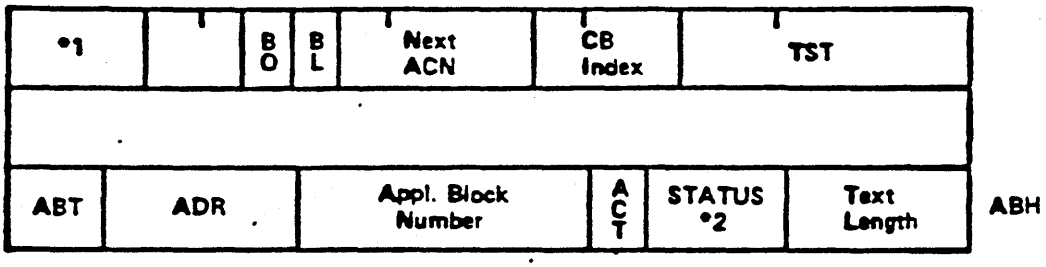
```
*          *TTRF* - COMMUNICATION RECOVERY FILE FIELD DEFINITIONS.

 TTLK      FIELD  0,59,59      1, IF FILE LOCKED
 TTEV      FIELD  0,58,58      1, IF FILE UNLOCK EVENT
 TTNP      FIELD  0,47,36      NUMBER OF PRU-S IN USER MESSAGE
 TTNM      FIELD  0,35,24      MAXIMUM NUMBER OF USER MESSAGES
 TTNW      FIELD  0,23,12      MAXIMUM WORDS IN USER MESSAGE
 TTNR      FIELD  0,11,0       NUMBER OF RECOVERY UNITS
 TTFT      FIELD  1,59,0       RECOVERY FILE FET
 TTRA      FIELD  9,17,0       RANDOM ADDRESS
 TTBF      FIELD  10,59,0      FWA OF BUFFER
```

NAM COMMUNICATION TABLE - NCT
(COMKNWC)

THE NAM COMMUNICATION TABLE (NCT) CONTAINS DETAILED INFORMATION ABOUT THE STATUS OF EACH TERMINAL WHILE IT IS LOGGED IN. THE APPLICATION CONNECTION NUMBER (ACN) IN THE TST ENTRY IS USED AS AN OFFSET INTO THIS TABLE.

LOW CORE POINTER WORD VNCT CONTAINS THE ADDRESS OF THE FIRST NCT TABLE ENTRY (ACN ZERO).

NAM COMMUNICATION TABLE - NCT
(COMKNWC)



NAM COMMUNICATION TABLE - NCT
(COMKNWC)

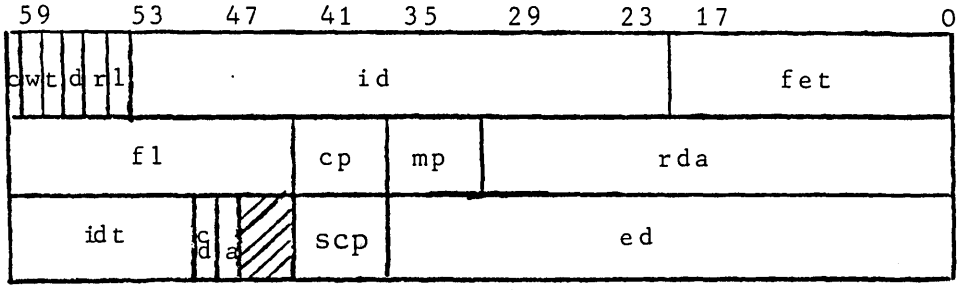
```
**          NCT - NAM COMMUNICATION TABLE.
*
*T, W1      9/NCTFLAG,6/ ,3/TNBO,3/TNBL,12/TNCN,9/TNCB,18/TNTS
*T, W2      8/PFC,1/EB,1/RB,6/SFC,44/PARM
*T, W3      6/ABT,12/ADR,18/ABN,4/ACT,8/STATUS,12/TLC
*
*
*           NCTFLAG = 1/TNDS,1/TNQS,1/TNSR,1/TNSL,1/TNSE,1/TNTF,1/TNBD,
*                     1/TNBK,1/TNSM
*
*           TNDS = 1 IF TERMINAL HAS STOP ON DOWN LINE CONNECTION.
*           TNQS = 1 IF QUEUED SUPERVISORY MESSAGE.
*           TNSR = 1 IF TASK SEND MESSAGE WITH RECALL.
*           TNSL = 1 IF INPUT FROM TERMINAL EXCEEDED LIMIT.
*           TNSE = 1 IF TASK IS ROLLED OUT ON SEND.
*           TNTF = 1 IF TERMINAL TEMPOFF.
*           TNBD = 1 IF BLOCK DELIVERED.
*           TNBK = 1 IF TERMINAL BREAK.
*           TNSM = 1 IF SUPERVISORY MESSAGE RETURN AFTER SEND.
*           TNBO = OUTSTANDING OUTPUT BLOCKS.
*           TNBL = OUTPUT BLOCK LIMIT.
*           TNCN = NEXT ACN IN QUEUE.
*           TNCB = INDEX OF COMMUNICATION BLOCK CONNECTION TO TERMINAL.
*           TNTS = ADDRESS OF TST ENTRY FOR CONNECTION.
*
*           STATUS = 1/IBU,3/RFE,1/NFE,1/XPT,1/CAN,1/BIT

 NCTA       CON     0           HEAD OF SUPERVISORY MESSAGE QUEUE
 NCTB       CON     0           END OF SUPERVISORY MESSAGE QUEUE
 TNCTL      EQU     3           LENGTH OF NCT ENTRY

 TNQS       FIELD   ,58,58      QUEUED SUPERVISORY MESSAGE
 TNSR       FIELD   ,57,57      TASK SEND WITH RECALL
 TNSL       FIELD   ,56,56      INPUT EXCEEDED LIMIT
 TNSE       FIELD   ,55,55      TASK ROLLED OUT ON SEND
 TNTF       FIELD   ,54,54      TERMINAL TEMPOFF
 TNBD       FIELD   ,53,53      BLOCK DELIVERED
 TNBK       FIELD   ,52,52      TERMINAL BREAK
 TNSM       FIELD   ,51,51      SUPERVISORY MESSAGE INDICATOR
 TNBO       FIELD   ,44,42      OUTSTANDING OUTPUT BLOCK
 TNBL       FIELD   ,41,39      OUTPUT BLOCK LIMIT
 TNCN       FIELD   ,38,27      NEXT ACN IN QUEUE
 TNCB       FIELD   ,26,18      COMMUNICATION BLOCK INDEX
 TNTS       FIELD   ,17,0       TST ENTRY FWA
 TNAH       FIELD   2,59,0      APPLICATION BLOCK HEADER
```

TASK ROLLOUT TABLE - ROLT

THE ROLLOUT TABLE (ROLT) IS USED FOR WRITING A TASKS FIELD LENGTH TO DISK (FILE KTSROLL) DURING PROCESSING OF CERTAIN REQUESTS. THESE REQUESTS TYPICALLY TAKE SOME TIME TO PROCESS, THUS MAKING ROLLOUT OF THE TASK A NECESSARY FUNCTION. IF TAF DECIDES TO ROLL A TASK OUT, IT CREATES A ROLLOUT TABLE ENTRY, BUT IN MOST CASES WILL NOT ROLL THE TASK OUT IMMEDIATELY. THE DELAY IS ADDED TO INCREASE THE CHANCE OF THE REQUEST THAT CAUSED THE ROLLOUT TO COMPLETE. EVEN ONCE THE ROLLOUT IS INITIATED, IF THE REQUEST COMPLETES, TAF STOPS THE ROLLOUT IN PROGRESS. IT JUST ZEROS OUT THE ROLT ENTRY AND FORGETS IT EVER STARTED THE ROLLOUT. AS LONG AS THE SUBCP TABLE ENTRY IS STILL INTACT, THE TASK CAN BE RESTARTED.

THE ROLLOUT TABLE ALLOCATION MAP AND ITS ENTRIES CAN BE FOUND IN TAF'S LOW CORE AT TAG TROM.

TASK ROLLOUT TABLE - ROLT



TASK ROLLOUT TABLE - ROLT

```
**          ROLT - ROLLOUT TABLE.
*
*T  W1      1/C,1/W,1/T,1/D,1/R,1/L,36/  ID,18/ FET
*T, W2      18/     FL,6/   CP,6/   MP,30/      RDA
*T, W3      12/IDT,1/C,1/A,4/0,6/SCP,36/ ED
*
*           WORD 1.
*              C       - 1, IF ROLLOUT COMPLETE.
*              W       - 1, IF WAIT FOR TERMINAL INPUT.
*              T       - 1, IF TIMED ROLLOUT.
*              D       - 1, IF DATA MANAGER REQUESTED ROLLOUT.
*              R       - 1, IF DO NOT RELOAD COMMUNICATION BLOCK.
*              L       - 1, IF RESTART TASK AT RECALL ENTRY POINT.
*              ID      - EVENT DESCRIPTOR.
*                        6/0,18/C.B. ADDRESS,12/0   FOR CALLRTN ROLL.
*                        24/TIME,12/TST ORDINAL FOR WAITINP ROLL.
*                        24/TIME,12/0      FOR MEMORY REQUEST ROLL.
*                        24/TIME,12/0  FOR WAIT ROLL.
*              FET     - FET ADDRESS OF ROLLOUT FILE.
*
*           WORD 2.
*              RDA     - DISK ADDRESS OF ROLLOUT FILE.
*              FL      - FIELD LENGTH REQUIRED FOR ROLLIN.
*              CP      - CURRENT PRIORITY.
*              MP      - MAXIMUM PRIORITY (FUTURE USE).
*
*           WORD 3.
*              IDT     - IDENTIFICATION OF TYPE OF EVENT.
*              C       - *CDCS* CONNECTED FLAG.
*              A       - ABORT TASK FLAG. (*CDCS* DOWN)
*              SCP     - SUBCONTROL POINT NUMBER.
*              ED      - EVENT DESCRIPTOR.
*                        18/ACN,18/ABN   FOR SEND IN *TAFNAM*.
*                        12/0,24/SEQ   FOR WAIT INPUT.
*                        18/0,18/FL   FOR MEMORY REQUEST.
*                        36/0     FOR CALLRTN ROLL.
*                        36/0     FOR WAIT ROLL.
```

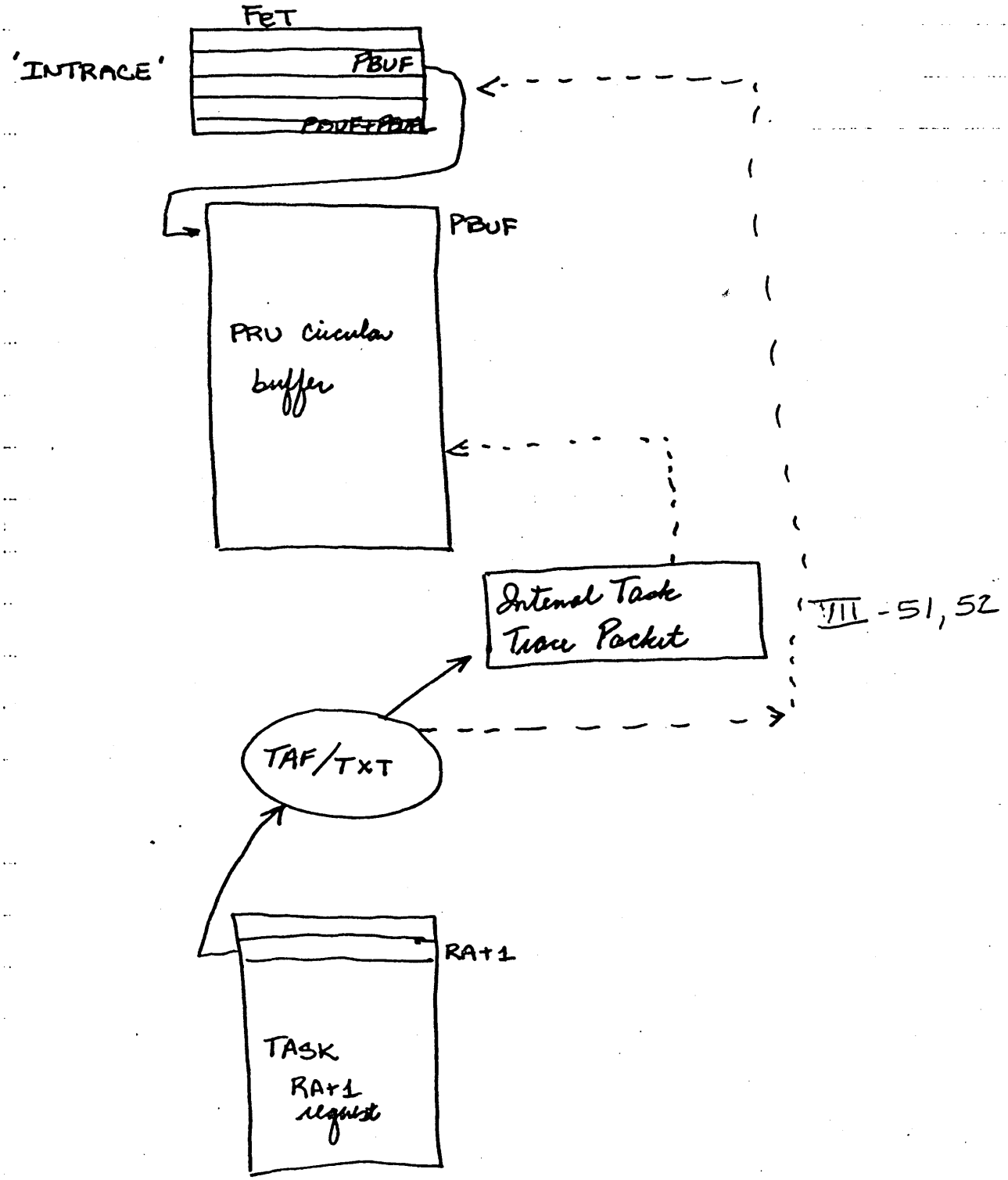
```
 TROLE    EQU    3            LENGTH OF A ROLLOUT TABLE ENTRY
 NROL     EQU    50           NUMBER OF ROLLOUT TABLE ENTRIES

 TROM     ALLOC  NROL,47      ROLLOUT TABLE ALLOCATION MAP

 TROL     BSSZ   TROLE*NROL   ROLLOUT TABLE

*           SHIFT COUNTS FOR *ROLT* FIELDS.

 RTBWI    EQU    58           WAIT FOR TERMINAL INPUT
 RTBTR    EQU    57           TIMED ROLLOUT
 RTBDM    EQU    56           DATA MANAGER ROLLOUT
 RTBID    EQU    53           ROLLOUT ID
 RTBCB    EQU    35           ROLLOUT CONTROL BLOCK INDEX
 RTBCP    EQU    23           SUBCP

 RTTL     FIELD  0,53,30      ROLLOUT TIME IN SECONDS
 RTCD     FIELD  2,47,47      *CDCS* CONNECTED FLAG
 RTAB     FIELD  2,46,46      ABORT TASK FLAG
 RTTS     FIELD  2,23,0       SEQUENCE NUMBER FOR *WAITINP*

*          WORD DEFINITIONS FOR *ROLT*.

 RTWEV    EQU    2            ROLT EVENT WORD

*          *ROLT* ID TYPES.

 EVDM     EQU    1            DATA MANAGER LOCKED RECORD EVENT
 EVTO     EQU    2            TERMINAL OUTPUT THRESHOLD ROLLOUT
 EVCR     EQU    3            CALL TASK WITH RETURN ROLLOUT
 EVWD     EQU    4            WAITING TO USE THE DATA MANAGER
 EVWI     EQU    5            WAIT FOR TERMINAL INPUT
 EVRL     EQU    6            REQUEST FIELD LENGTH
```

INTERNAL TASK TRACE PACKET - ITTP

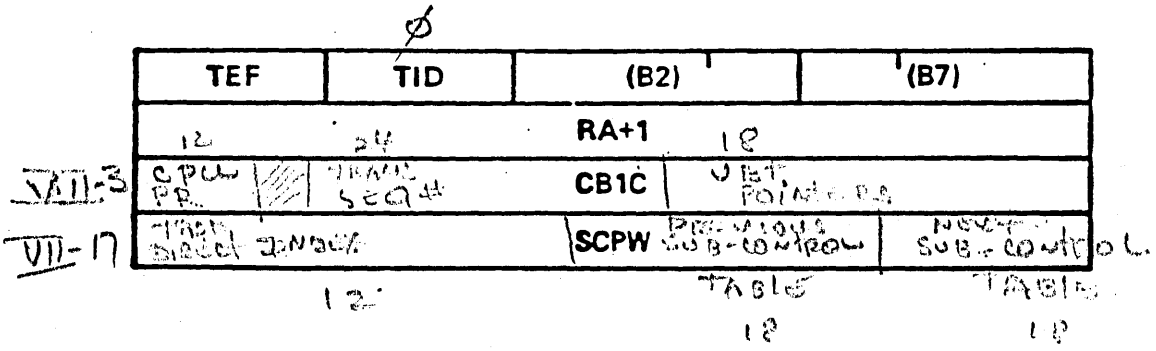
THE INTERNAL TASK TRACE PACKET (ITTP) IS CONTAINED IN A CIRCULAR BUFFER (PBUF). EACH ENTRY IS FOUR WORDS LONG AND CONTAINS INFORMATION ABOUT THE TASK THAT HAS JUST ISSUED AN RA+1 REQUEST.

THE TRACE BUFFER FET CAN BE FOUND IN TAF'S LOW CORE AT TAG INTRACE.

Internal Trace Packet Building



INTERNAL TASK TRACE PACKET - ITTP



INTERNAL TASK TRACE PACKET - ITTP

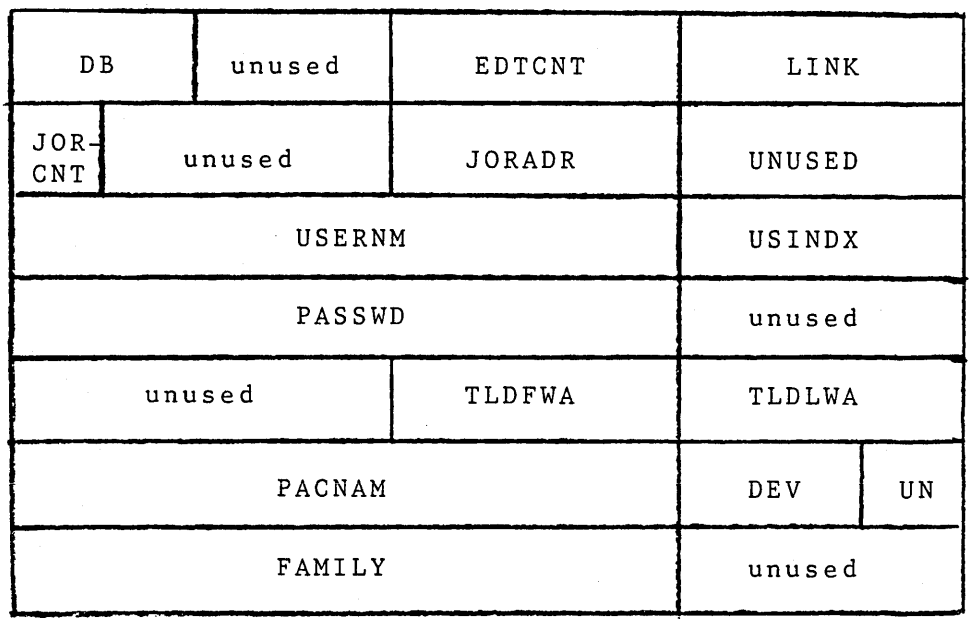
```
**          ITTP - INTERNAL TASK TRACE PACKET.
*
*T W1       12/    TEF,12/    TID,18/    (B2),18/    (B7)
*T,W2       60/    RA+1
*T,W3       60/    CB1W
*T,W4       60/    SCPW
*
*           WORD 1.
*              TEF    - 2000B+ERROR FLAG RETURNED FROM SUBCP ACTIVATION.
*              TID    - TASK TRACE PACKET IDENTIFIER (SET TO ZERO).
*              (B2)   - START OF SYSTEM AREA PRECEDING THE RA OF THE TASK.
*              (B7)   - ADDRESS OF SUBCP TABLE.
*
*           WORD 2.
*              RA+1   - CONTENTS OF RA+1 IN THE TASK FL.
*
*           WORD 3.
*              CB1W   - FIRST WORD OF C.B. KEPT IN THE SYSTEM AREA
*                       PRECEDING THE RA OF THE TASK.
*
*           WORD 4.
*              SCPW   - THIRD WORD OF SUBCP TABLE.
*                       (SEE DEFINITION IN *COMKSCD*.)

 ITTPL      EQU    4           LENGTH OF AN INTERNAL TASK TRACE PACKET
            ERRNG  ITTPL-4     TRACE PACKET SIZE REDEFINED TO LESS THAN 4
```

ELEMENT DESCRIPTOR TABLE - EDT
(TAF1)

THE EDT CONTAINS INFORMATION ABOUT JOURNAL FILES AND TASK LIBRARIES ASSOCIATED WITH A DATA BASE. IT IS USED BY THE EXECUTIVE WHEN SEARCHING FOR A TASK TO LOAD FROM A PARTICULAR LIBRARY.

LOW CORE POINTER WORD VEDT CONTAINS THE ADDRESS OF THE EDT FOR THE FIRST DATA BASE SPECIFIED IN THE TAF CONFIGURATION FILE (TCF).

ELEMENT DESCRIPTOR TABLE - EDT
(TAF1)



ELEMENT DESCRIPTOR TABLE - EDT
(TAF1)

```
**        EDT - ELEMENT DESCRIPTION TABLE.
*
*T VEDT1  12/  DB,12/    ,18/EDTCNT,18/  LINK
*T,VEDT2  6/JORCNT,18/       ,18/JORADR,18/TRCADR
*T,VEDT3  42/         USERNM,18/USINDX
*T,VEDT4  42/         PASSWD,18/
*T,VEDT5  24/0,18/TLDFWA,18/TLDLWA
*T,VEDT6  42/ PACNAM,12/ DEV,6/ UN
*T,VEDT7  42/FAMILY,18/
*
*         WORD 1.
*            DB      - DATA BASE NAME.
*            EDTCNT  - NUMBER OF EDT-S (PRESENT ONLY IN FIRST HEADER).
*            LINK    - POINTER TO NEXT EDT.
*
*         WORD 2.
*            JORCNT  - NUMBER OF JOURNAL FILES (MAXIMUM OF 3 PER DB).
*            JORADR  - ADDRESS OF FIRST JOURNAL FILE FET.
*            TRCADR  - ADDRESS OF TRACE FILE FET.
*
*         WORD 3.
*            USERNM  - USER NUMBER (USED TO ATTACH MULTIPLE TLD-S).
*            USINDX  - USER INDEX (TO ATTACH JOURNAL AND
*                                  DATA BASE FILES).
*
*         WORD 4.
*            PASSWD  - PASSWORD.
*
*         WORD 5.
*            TLDFWA  - FWA OF DBTASKL (NAME OF PARTICULAR TLD).
*            TLDLWA  - LWA OF DBTASKL.
*
*         WORD 6.
*            PACNAM  - PACK NAME OF AUXILIARY DEVICE ON WHICH THE
*                      TASK LIBRARY RESIDES.
*            DEV     - DEVICE TYPE THE FILE WILL BE RESIDING
*                      ON (DI,DJ,...,).
*            UN      - NUMBER OF UNITS OF THE TYPE SPECIFIED IN THE
*                      DEVICE TYPE FIELD.
*
*         WORD7.
*            FAMILY  - USER FAMILY NAME.
```

LOW CORE POINTERS

VNSCP (10)	Number of subcontrol points (24-41)
VNCMB (11)	Number of communication blocks (24-41)
VTST (12)	Start of terminal status table (FWA/24-41 LWA/0-17)
VNTST (13)	Number of terminals (24-41)
VNSIN (14)	Number of CB's reserved for small input (24-41)
VLSP (15)	Address of last subcontrol point (24-41)
VATL (16)	Address of Active Transaction List (24-41)
VFSCP (17)	Start of subcontrol point allocatable storage (24-41)
VCBRT (20)	Start of communication block storage allocation bit maps (24-41)
VCBSA (21)	Start of communication blocks (24-41)
VTLD (22)	Start of task library directory (FWA/24-41, LWA/0-17)
VEDT (23)	Base address of element descriptor tables (24-41)
VPOTT (24)	Unused
VMFL (25)	Maximum field length for transaction subsystem (0-17)
VTFL (26)	Task library file name
VREC (27)	Recovery flag
VCRAT (30)	Start of copied record address table (unused)
VECS (31)	EM field length (0-17)
VECSC (32)	Current next available EM address (0-17)
VCRS (33)	CRAS terminal name (unused)

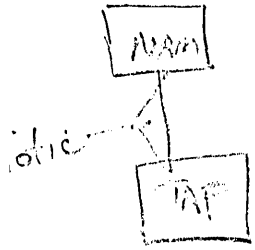
LOW CORE POINTERS (con't)

VRLAT (35)	Rollout file allocation map (24-41)
VCPA (36)	Address of first subcontrol point table (24-41)
VTOT (37)	TOTAL data manager initialization flag
VAAM (41)	TAF/CRM data manager initialization flag (30-47) and FWA of routine to process CRM requests (0-17)
VAAQ (42)	FWA of FETs for TAF/CRM input and output queues (24-41 and 0-17, respectively)
VAMB (43)	FWA of AAM record buffer (0-17) and address of first logical name table (TLNT entry (24-41)
VINT (44)	Initialization complete flag (Bit 0)
VNACP (45)	Address of pointer to free subcontrol point (24-41)
VOEP (46)	Overlay entry point name list (24-41)
VOREL (47)	Overlay relocation list (FWA/24-41, LWA/0-17)
VSIW (50)	TAF identification word (SSIW)
VSCR (51)	TAF receiving buffer (SSCR)
VSTAT1 (52)	Statistics area (0-17)
VRTLW (53)	Requested task list (0-17)
VNCT (55)	NAM communication table (FWA/24-41, number of entries/0-17)
VNON (56)	NETON status (zero if NAM is running)
VSND (57)	Application block number for NAM (0-12)

VCMM (60)	CMM FL 24/0, 18/EFL, 18/BFL
VLWP (65)	CMM Memory Management word
VHHA (104)	CMM "HHA"
VBCT (111)	FWA of Batch Communication Table (24-41) Length of Subcp area for Batch Concurrency request (0-17)
VNBCT (112)	Allowed number of Batch Concurrency requests (30-47) Maximum number of Batch Concurrency requests (0-17)
VLOCL2 (114)	FWA for loading AIP

FLOW OF A TRANSACTION THROUGH TAF

INITIAL TRANSACTION INPUT



NAM send connection request to TAF

ITASK gets called - sends "READY" back to terminal

Operator sends input to system

NAM receives input & sends it directly to CB within TAF

TAF builds CB system and user headers

Writes input to JOURO

TAF Schedules ITASK to process CB

ITASK issues a CALLTRN request

TAF searches Transaction Directory for Transaction unit name

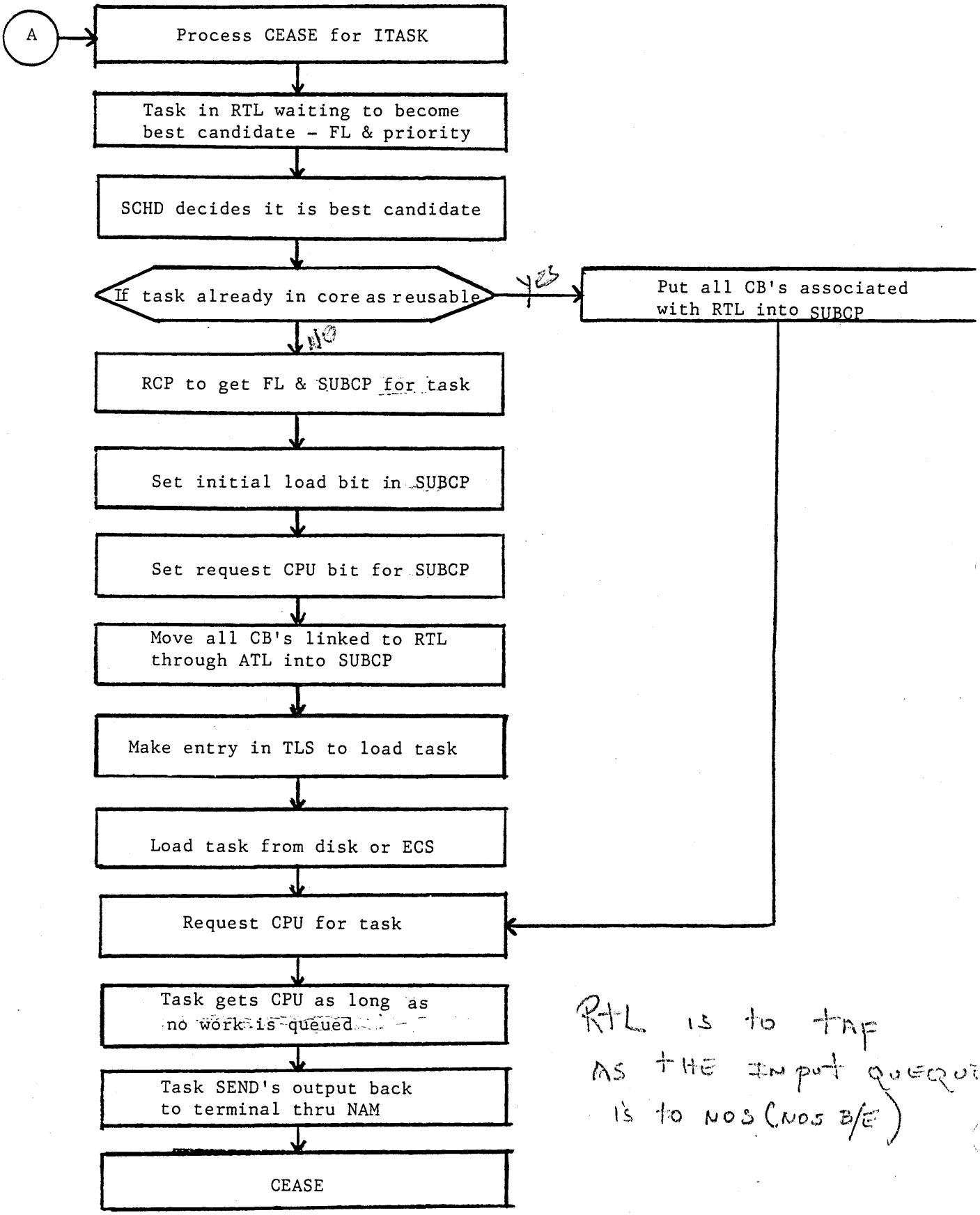
TAF updates CB System Header (CBTL/CBCS/CBRM)

TAF Queues Terminal Input Logging (QTW)

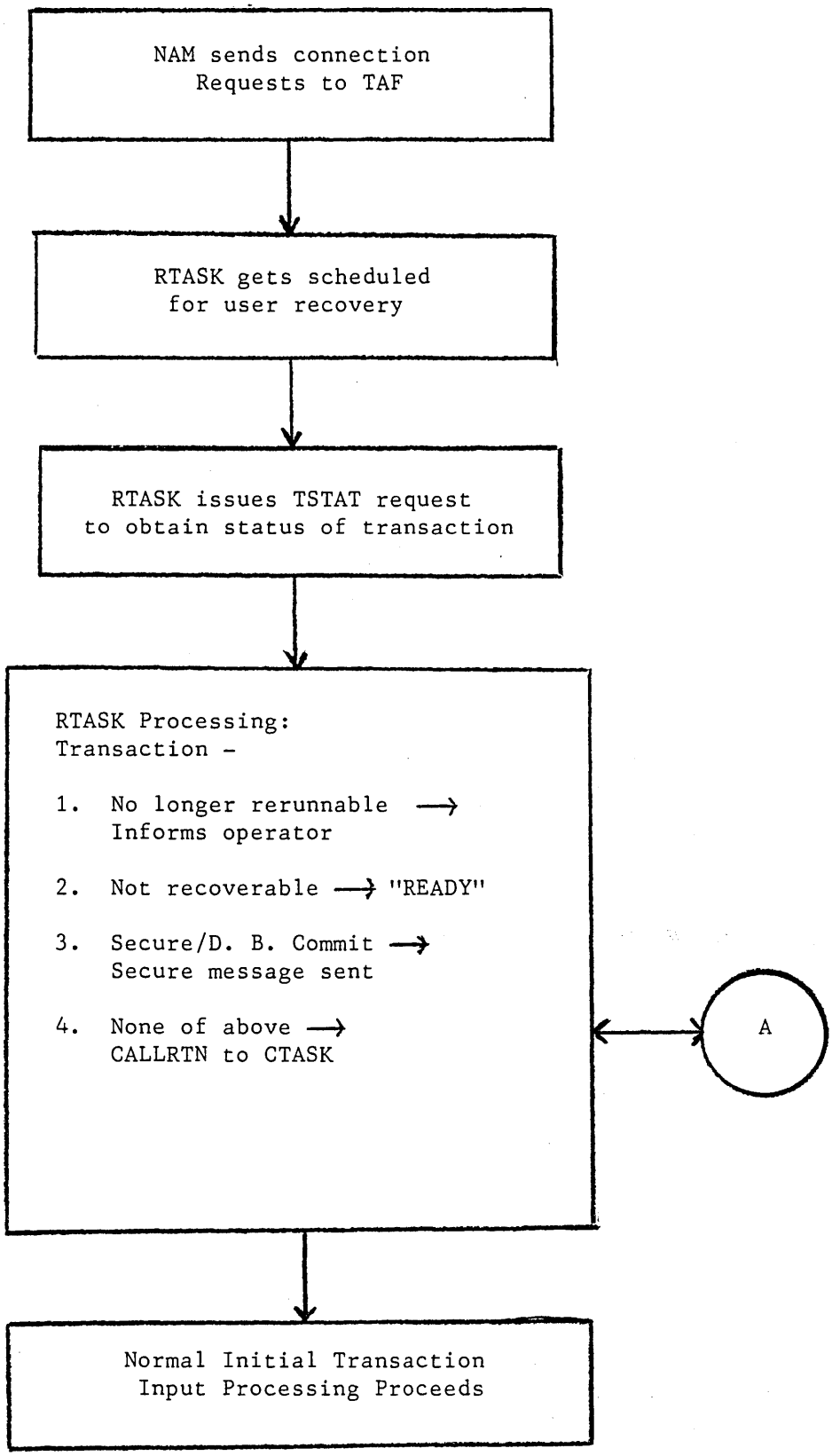
ETSQ - Enters First task into RTL

A

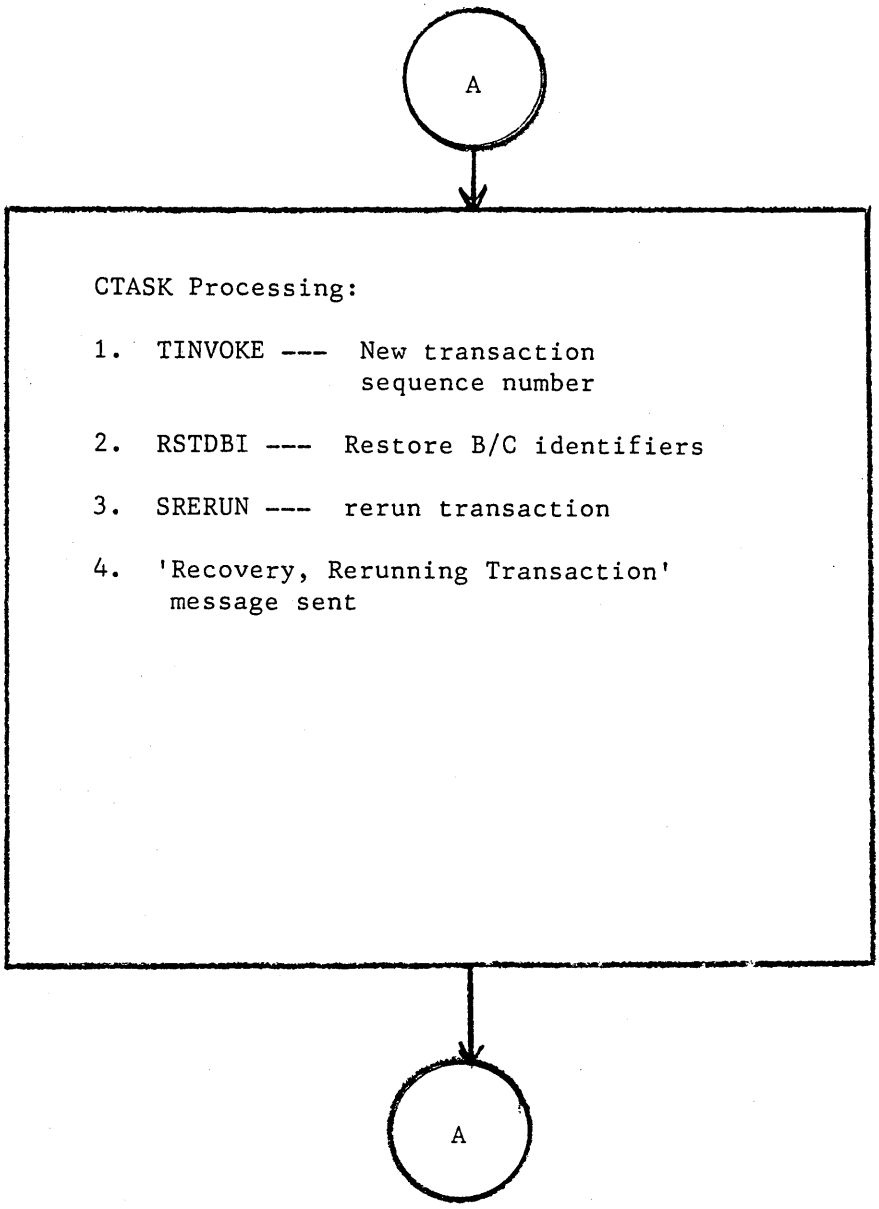
INITIAL TRANSACTION INPUT (cont)



Recovery Situation

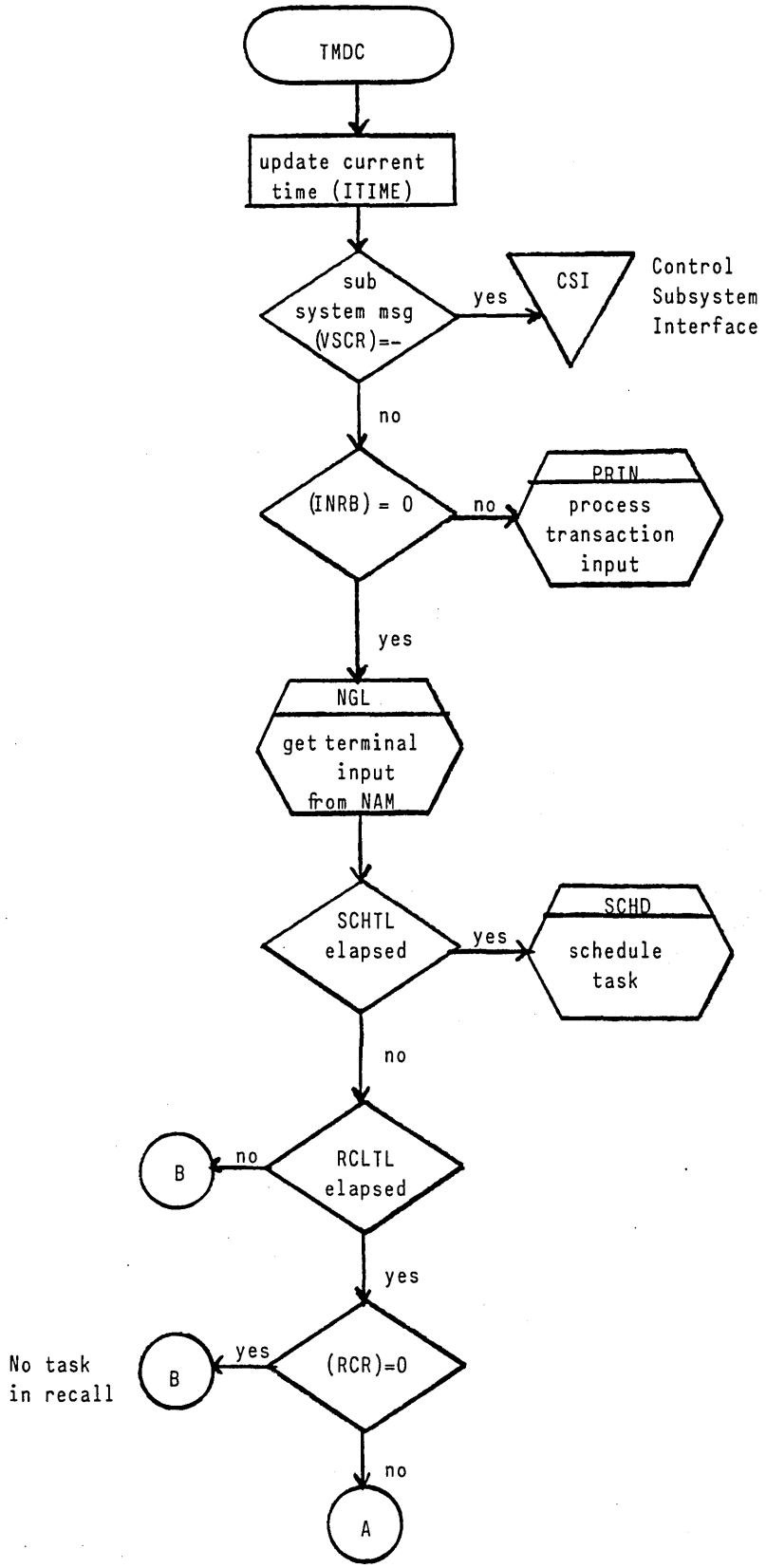


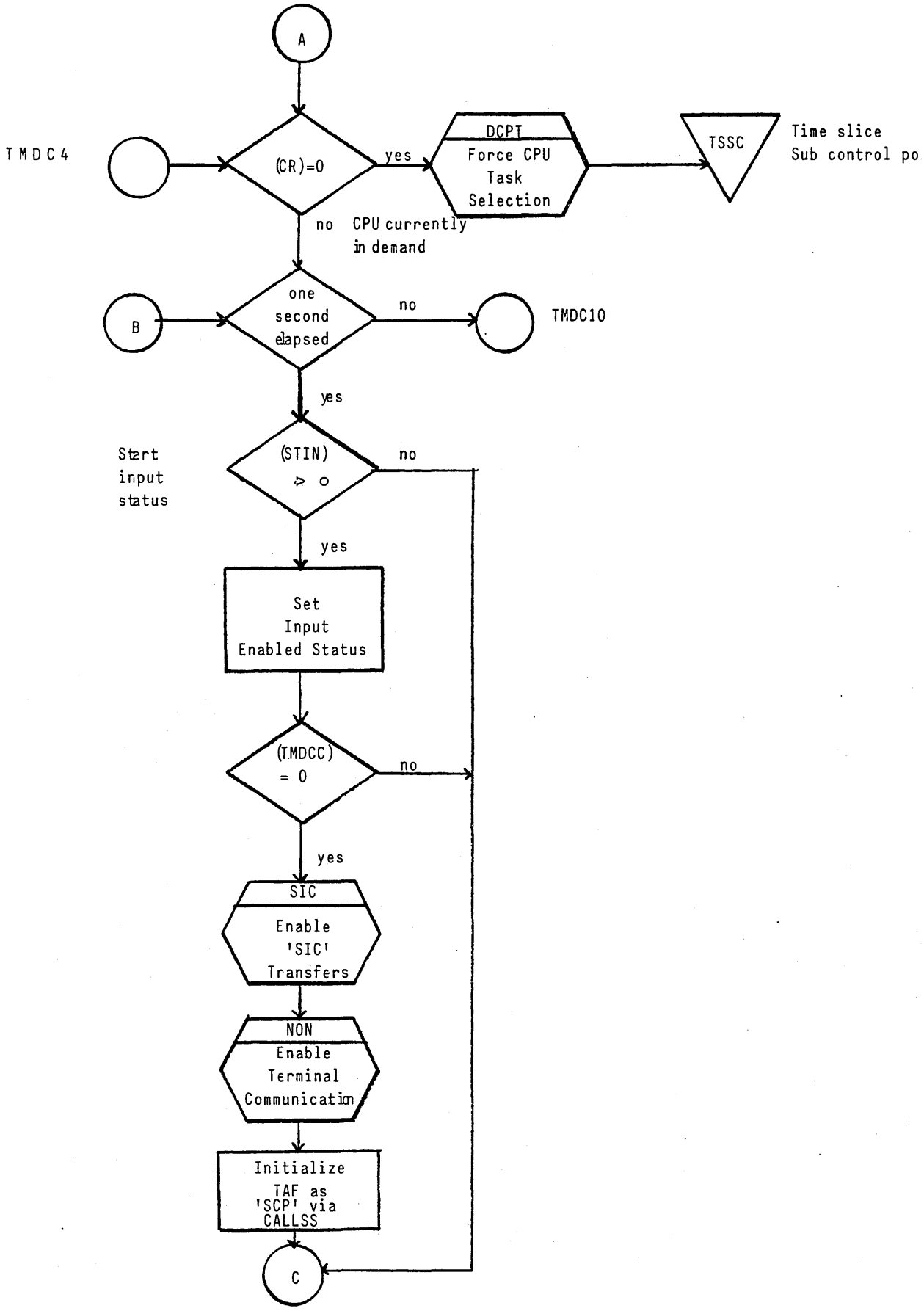
Recovery Situation (cont)

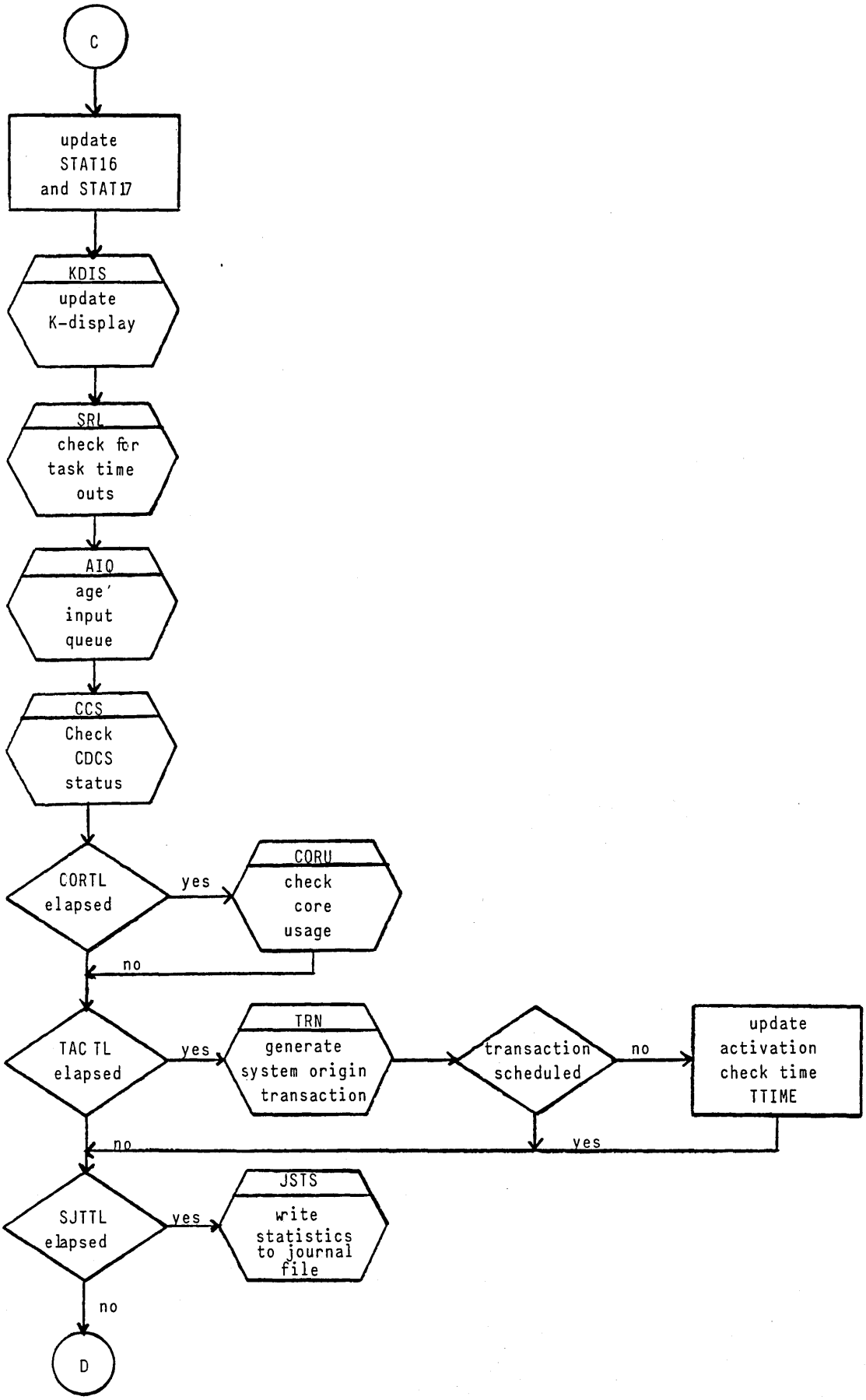


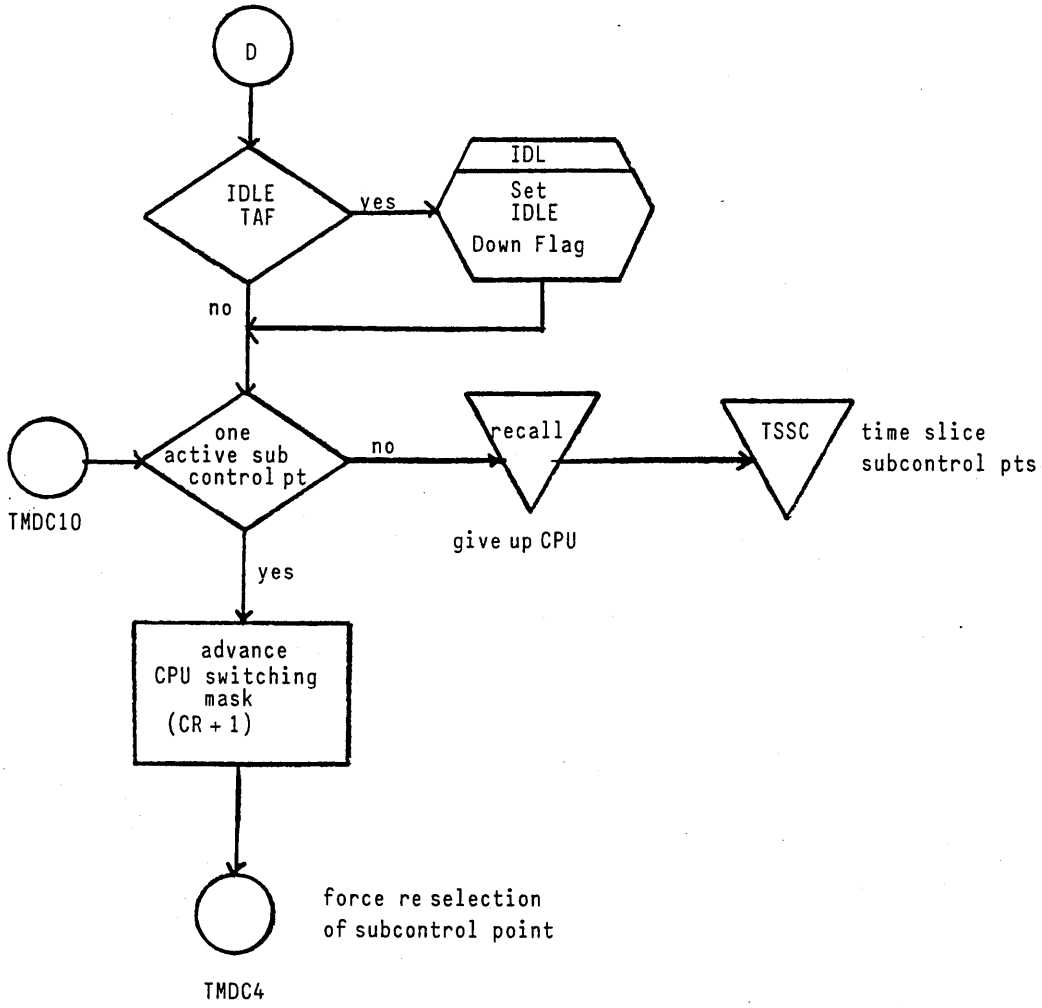
FLOW OF PROCESSING WITHIN TAF

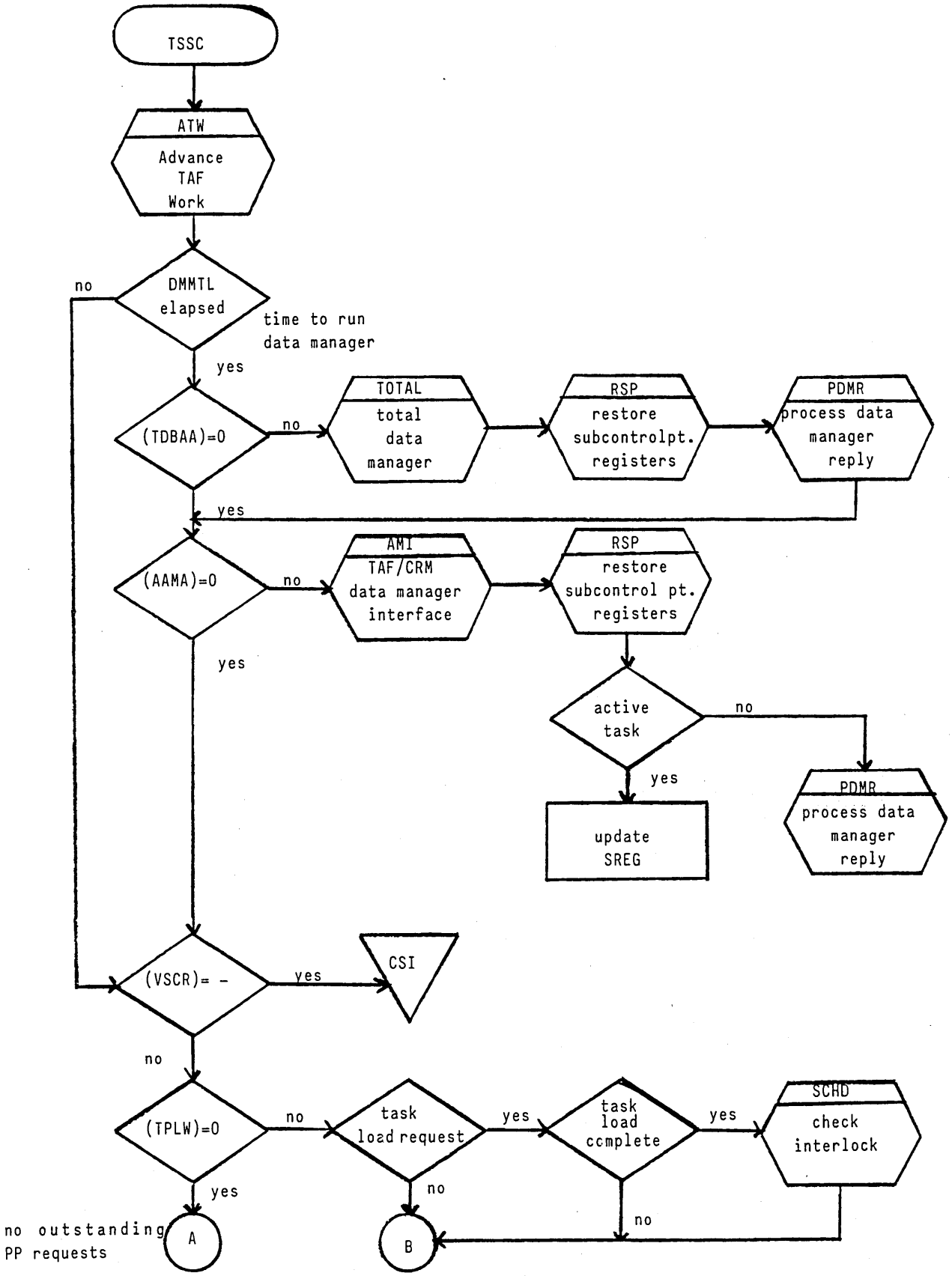
TMDC

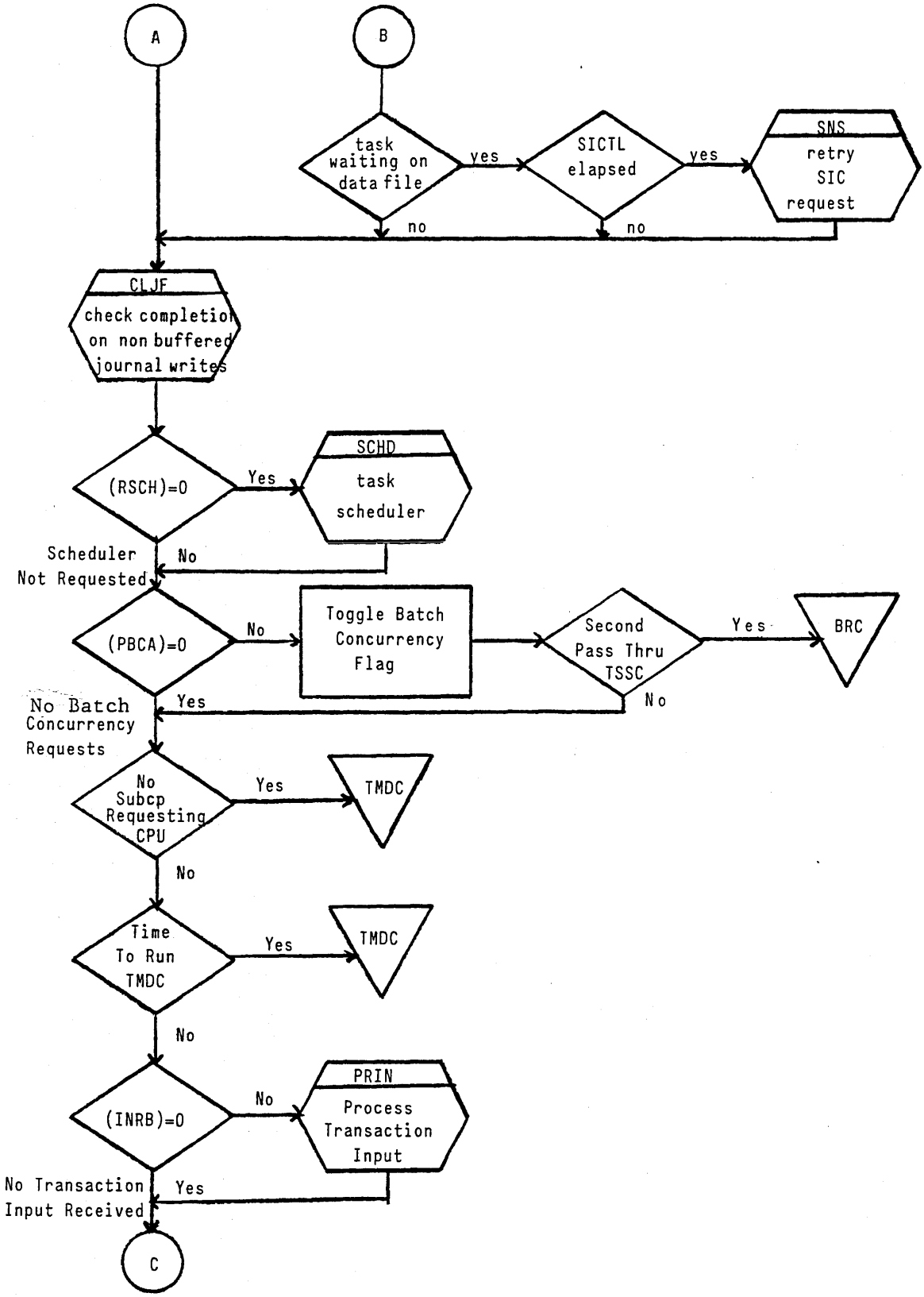

TMDC
update current time (ITIME)
sub system msg (VSCR)=-
yes
CSI
Control Subsystem Interface
no
(INRB) = 0
no
PRIN
process transaction input
yes
NGL
get terminal input from NAM
SCHTL elapsed
yes
SCHD
schedule task
no
RCLTL elapsed
no
B
yes
(RCR)=0
yes
B
No task in recall
no
A

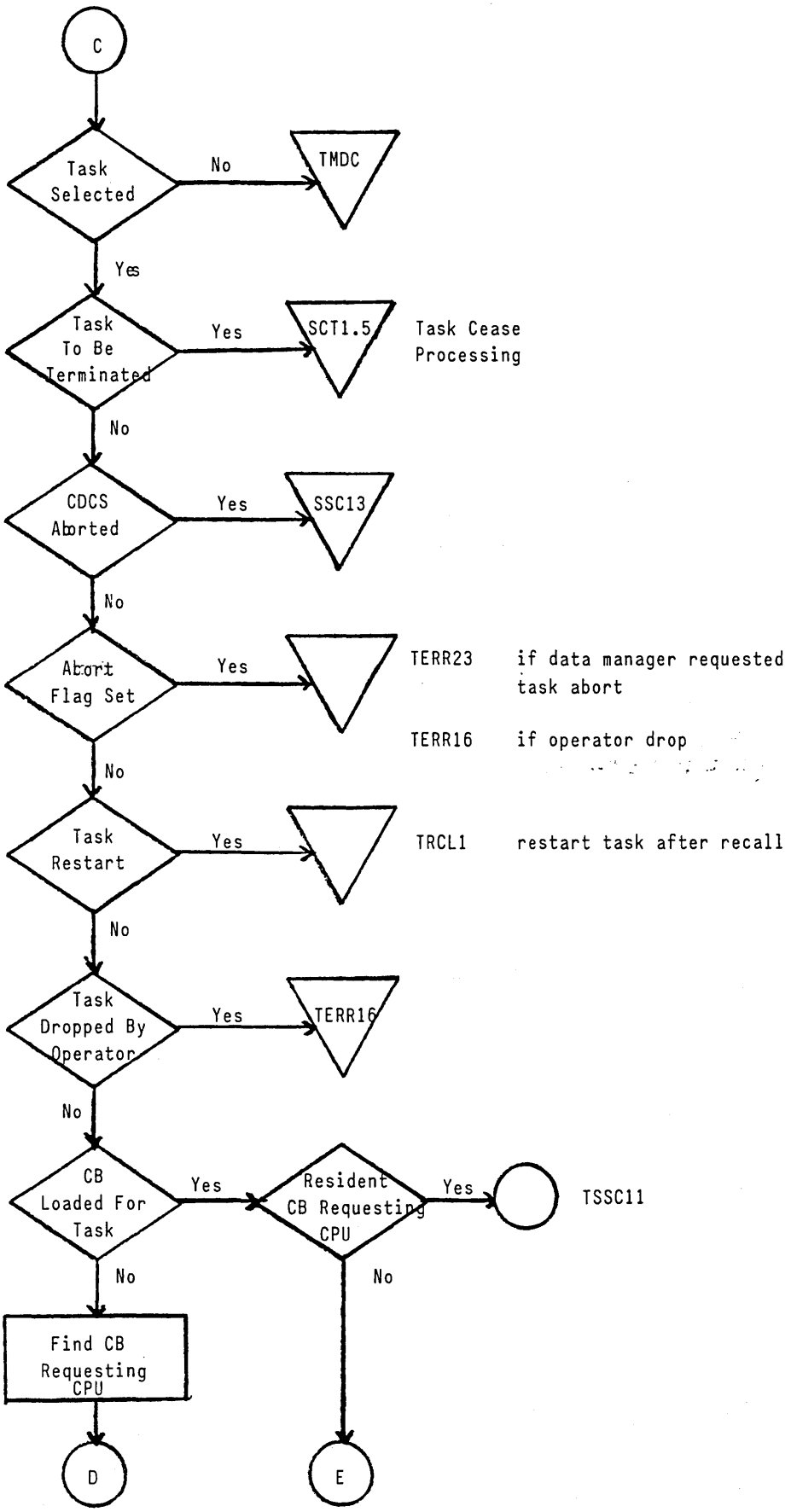

A
TMDC4
(CR)=0
yes
DCPT
Force CPU
Task
Selection
TSSC
Time slice
Sub control po
no
CPU currently
in demand
B
one
second
elapsed
no
TMDC10
yes
Start
input
status
(STIN)
> 0
no
yes
Set
Input
Enabled Status
(TMDCC)
= 0
no
yes
SIC
Enable
'SIC'
Transfers
NON
Enable
Terminal
Communication
Initialize
TAF as
'SCP' via
CALLSS
C

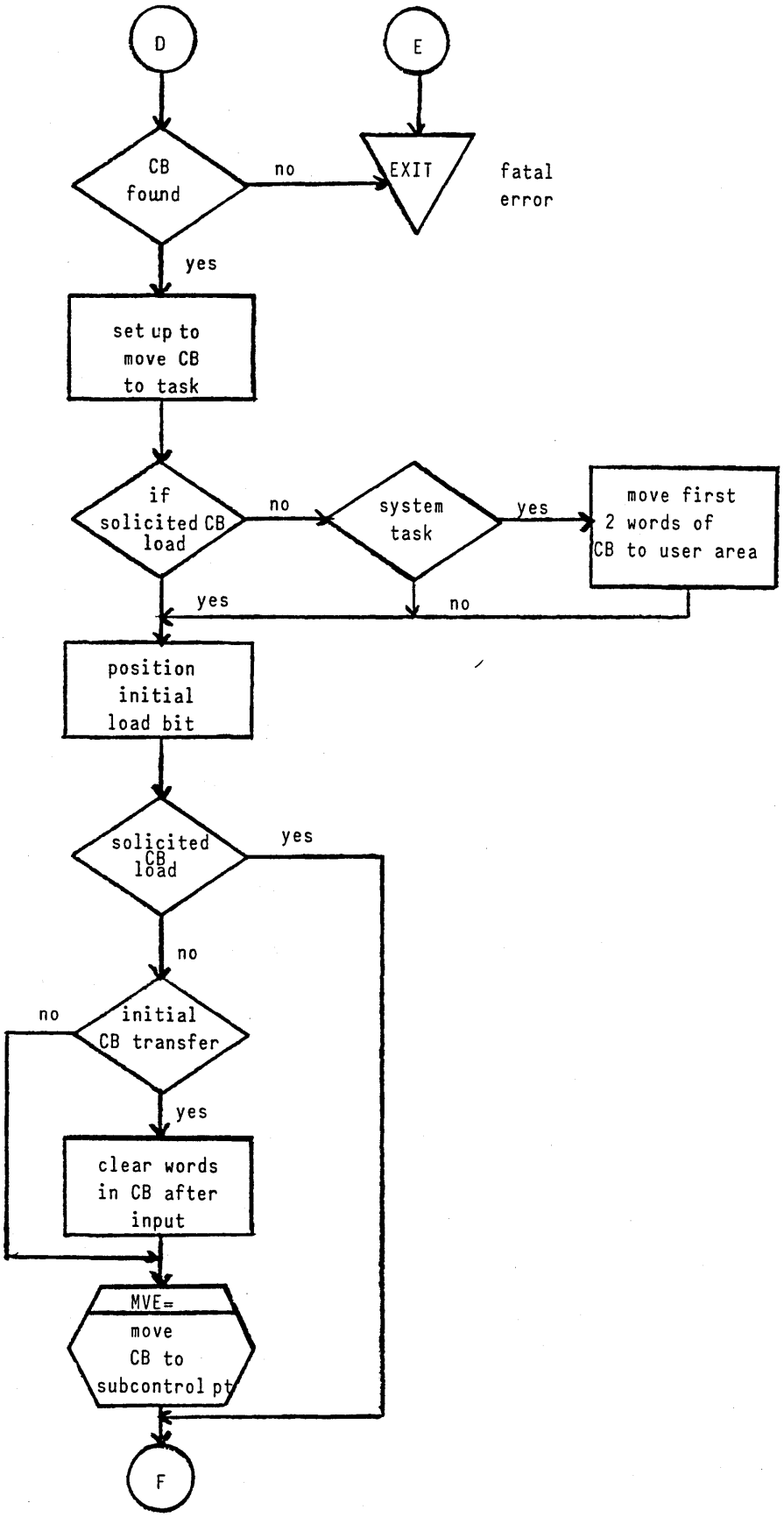

C
update STAT16 and STAT17
KDIS
update K-display
SRL
check for task time outs
AIQ
age input queue
CCS
Check CDCS status
CORTL elapsed
yes
CORU
check core usage
no
TAC TL elapsed
yes
TRN
generate system origin transaction
transaction scheduled
no
update activation check time TTIME
no
yes
SJTTL elapsed
yes
JSTS
write statistics to journal file
no
D

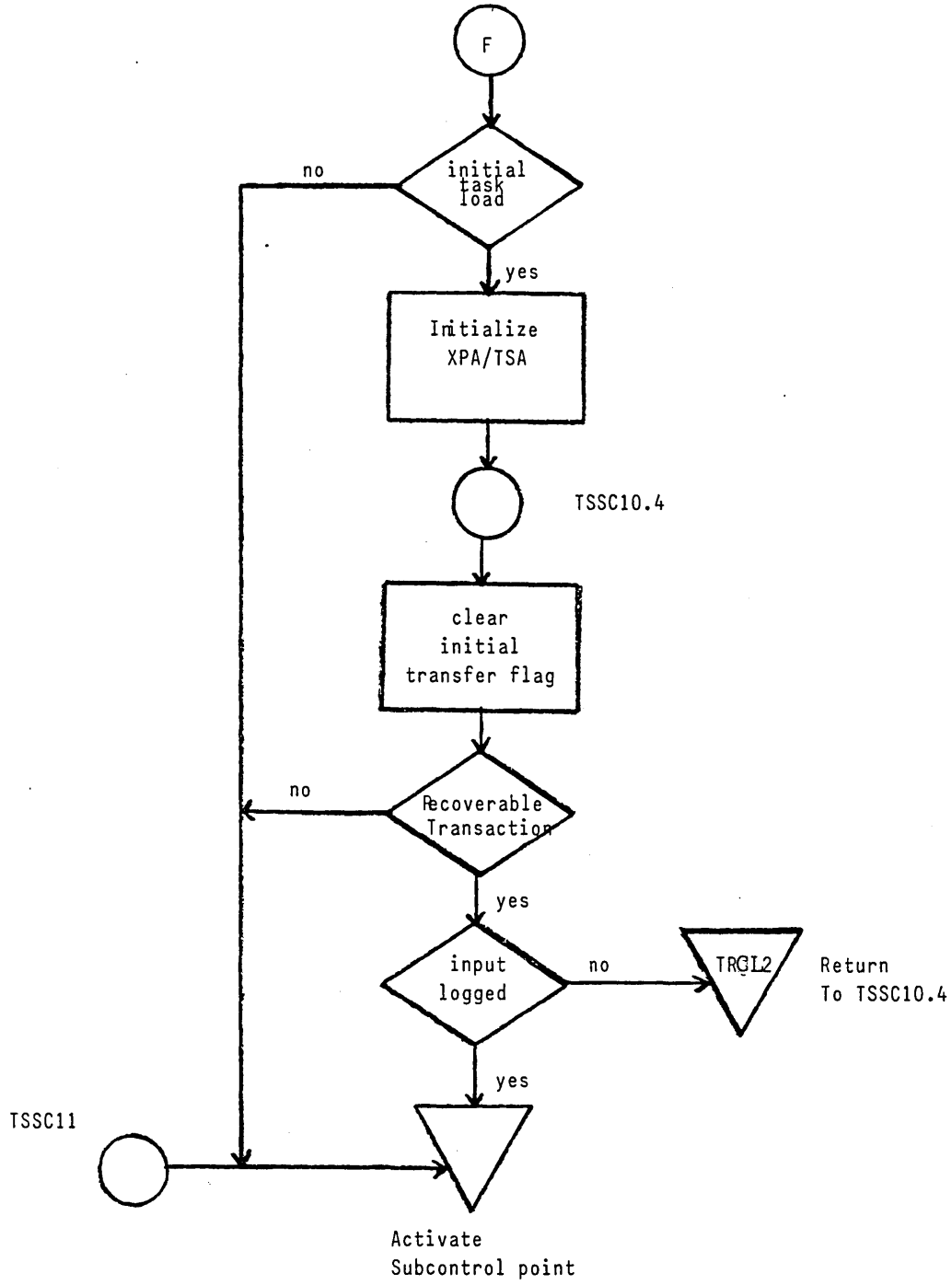

D
IDLE TAF
yes
IDL
Set IDLE Down Flag
no
one active sub control pt
no
recall
TSSC
time slice subcontrol pts
TMDC10
give up CPU
yes
advance CPU switching mask (CR + 1)
force re selection of subcontrol point
TMDC4


TSSC
ATW
Advance TAF Work
DMMTL elapsed
no
time to run data manager
yes
(TDBAA)=0
no
TOTAL
total data manager
RSP
restore subcontrolpt. registers
PDMR
process data manager reply
yes
(AAMA)=0
no
AMI
TAF/CRM data manager interface
RSP
restore subcontrol pt. registers
yes
active task
no
yes
update SREG
PDMR
process data manager reply
(VSCR)= -
yes
CSI
no
(TPLW)=0
no
task load request
yes
task load complete
yes
SCHD
check interlock
yes
no
no
no outstanding PP requests
A
B


A
B
task waiting on data file
yes
SICTL elapsed
yes
SNS
retry SIC request
no
no
CLJF
check completion on non buffered journal writes
(RSCH)=0
Yes
SCHD
task scheduler
Scheduler Not Requested
No
(PBCA)=0
No
Toggle Batch Concurrency Flag
Second Pass Thru TSSC
Yes
BRC
No
No Batch Concurrency Requests
Yes
No Subcp Requesting CPU
Yes
TMDC
No
Time To Run TMDC
Yes
TMDC
No
(INRB)=0
No
PRIN
Process Transaction Input
No Transaction Input Received
Yes
C


C
Task Selected
No
TMDC
Yes
Task To Be Terminated
Yes
SCT1.5
Task Cease Processing
No
CDCS Aborted
Yes
SSC13
No
Abort Flag Set
Yes
TERR23
if data manager requested task abort
TERR16
if operator drop
No
Task Restart
Yes
TRCL1
restart task after recall
No
Task Dropped By Operator
Yes
TERR16
No
CB Loaded For Task
Yes
Resident CB Requesting CPU
Yes
TSSC11
No
No
Find CB Requesting CPU
D
E


D
E
CB found
no
EXIT
fatal error
yes
set up to move CB to task
if solicited CB load
no
system task
yes
move first 2 words of CB to user area
yes
no
position initial load bit
solicited CB load
yes
no
initial CB transfer
no
yes
clear words in CB after input
MVE=
move CB to subcontrol pt
F


F
initial task load
no
yes
Initialize XPA/TSA
TSSC10.4
clear initial transfer flag
Recoverable Transaction
no
yes
input logged
no
TRGL2
Return To TSSC10.4
yes
TSSC11
Activate Subcontrol point

TAF WORK QUEUEING

- TAF Queues Work For:

  1. Initial Transaction Input Logging To CRF
  2. Communication Recovery Requests
  3. TAF/CRM Batch Concurrency Requests

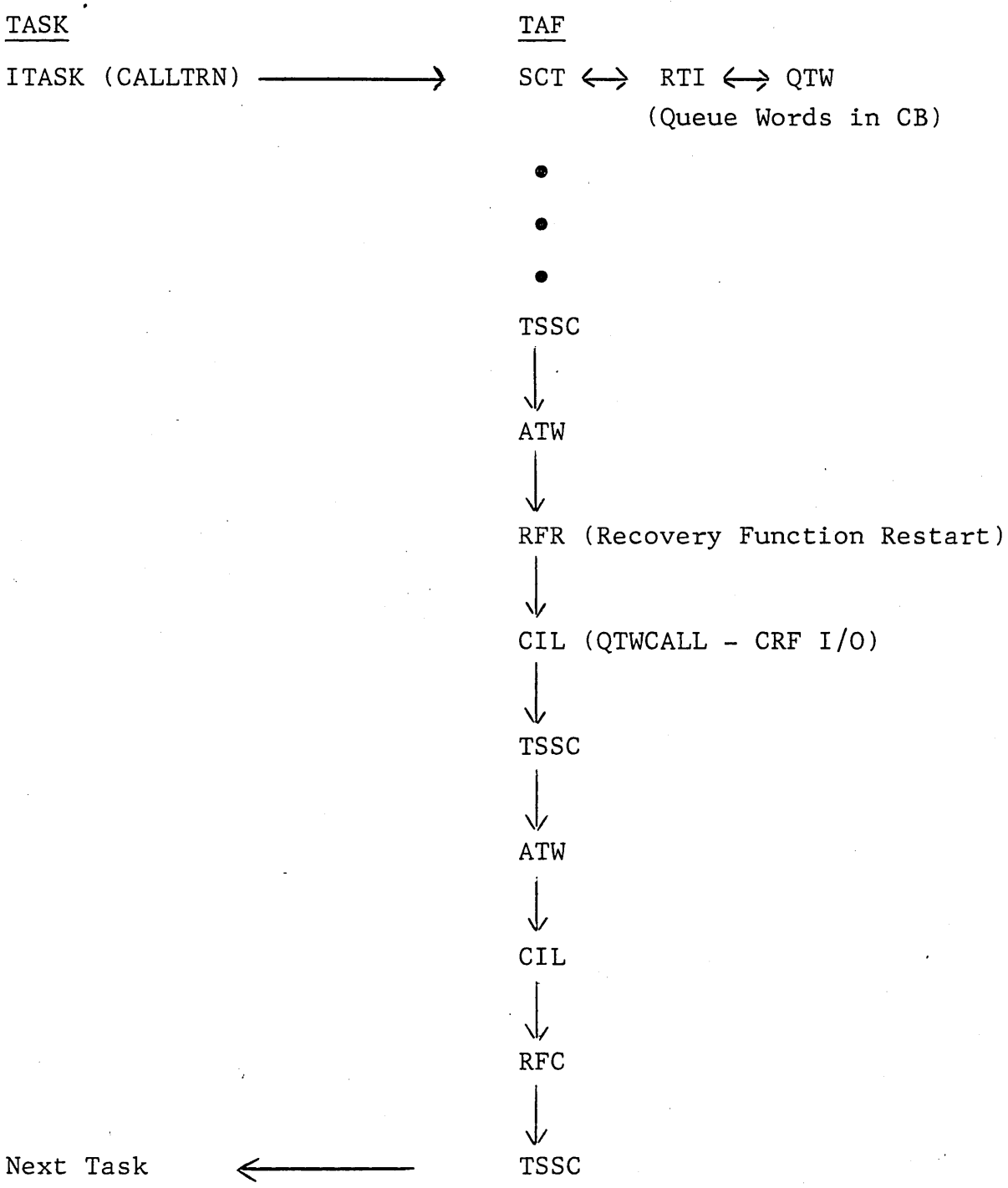
- Queued Work Entries Reside In:

  1. Communication Block
  2. Task System Area (RCL)
  3. Batch Communication Table
  4. TAF Storage Buffer (TSB)

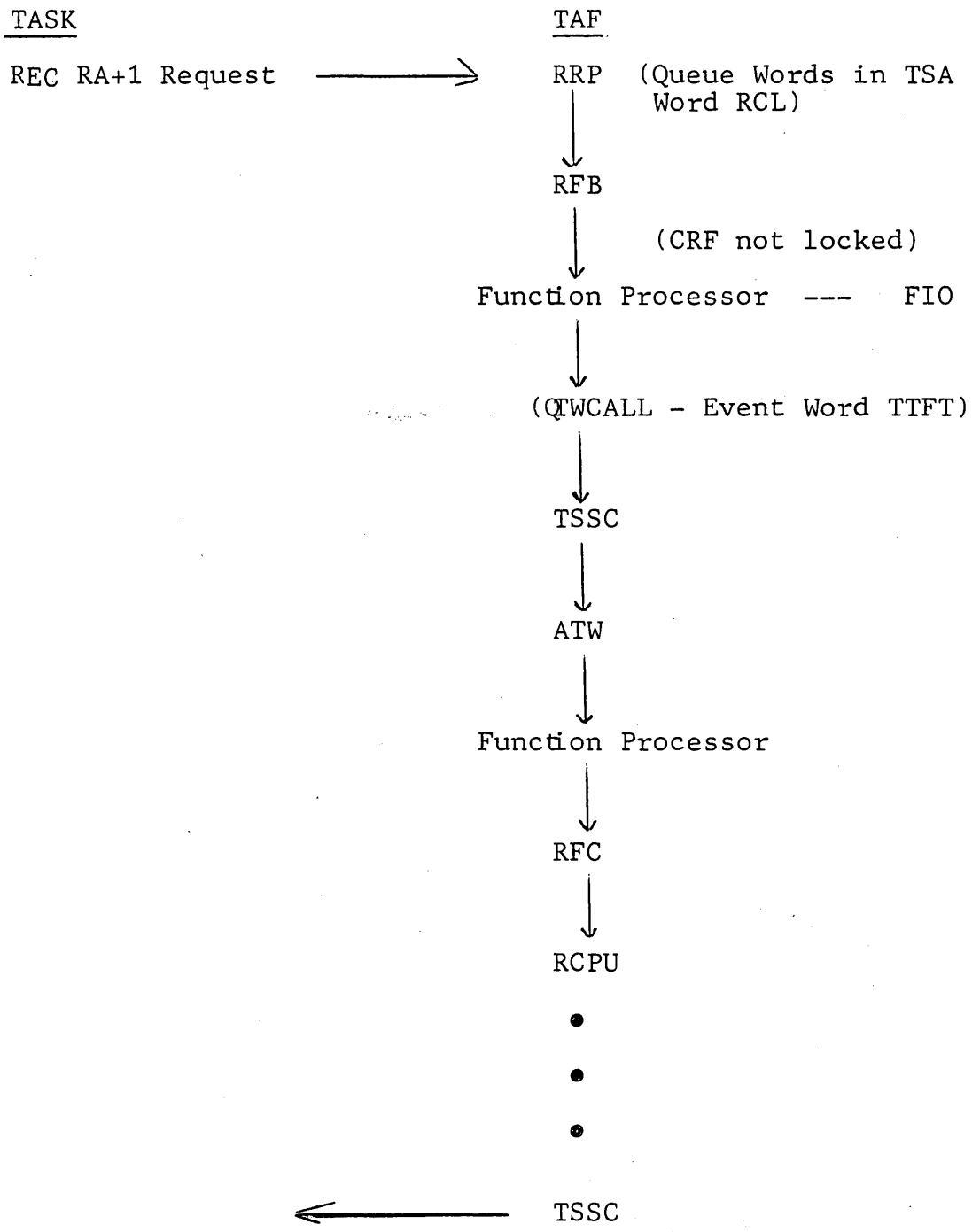
TAF WORK QUEUEING (con't)

- Event Words Reside In:
  1. TTRF (TTEVW/TTFTW)
  2. CB System Header (CBQ3)
  3. EVCB
  4. EVIT
  5. BCT (BCSFW)
  6. TSB entry + 2
- Word 'TAFQ' Points to Beginning of Circular Queue
- Routines
  1. QTW/ATW
  2. GTS/RTS
- QTWCALL Macro

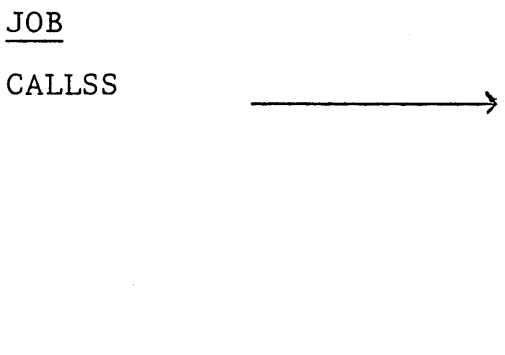
Initial Transaction Input Logging to CRF



COMMUNICATION RECOVERY REQUESTS



TAF/CRM Batch Concurrency Requests



Queueing occurs in:

1. SFR - if waiting for job swapin after SFCALL request.
2. VBA - issuing an internal TSTAT request for validation.
3. BAM - queueing TAF/CRM requests awaiting completion.
4. BRC - issuing an internal WSTAT request for DBCOMIT/JOB termination.

RA+1 REQUEST PROCESSING

SUBCONTROL POINT RETURN

- TASK ERROR
- TASK RA+1 REQUEST
- TASK TIME SLICE

RA+1 REQUESTS

- Task Scheduling - SCT

  0 - CEASE

  1 - NEWTRAN

  2 - CALLTSK W/CEASE

  3 - CALLTSK W/O CEASE

  4 - CALLRTN

  5 - WAITINP

  6 - WAIT

  7 - CHKON

  10 - CHKOFF

  11 - BWAITINP

  12 - CALLTRN

- Data Manager Requests

  AAM - TAF/CRM

  SSC - CDCS

  TOT - TOTAL

- Transaction Interface for Task Control - CTI

  0 - SEND (CTS)

  1 - JOURNL

  2 - TRANCHK

  3 - TARO

  4 - CMDUMP

  5 - DSDUMP

  6 - TSIM

  7 - KPOINT

  10 - KDIS

  11 - Reserved

  12 - SUBMT

  13 - ITL

  14 - IIO

  15 - LOGT (TLO)

  16 - LOADCB

  17 - RELSCB

  20 - SETCHT (CTA)

  21 - TERMDEF (CTD)

  22 - GETABH (CTH)

  23 - Task Request Argument Error (TERR28)

  24 - TPSTAT

  25 - BEGIN

  26 - TSTAT (TFP)

  27 - WSTAT (WFP)

RA+1 REQUESTS (cont)

- Communication Recovery - REC

  0 - CALLTRN (CIL)

  1 - RERUN (RUN)

  2 - RGET (GET)

  3 - RPUT (PUT)

  4 - RSECURE (RSE)

  5 - SECURE (SEC)

  6 - SRERUN (SRE)

  7 - TINVOKE (TIN)

- Task Memory Requests

  MEM - Functions 0/2

  RFL

RA+1 REQUESTS (cont)

- TAF Dayfile/Console Display Messages

  MSG - Functions 0/1

  TMS - MSG Function 0/System Task only

- Miscellaneous Task Requests

  ABT - Dump Tasks FL/End Task (SCT24)

  CPM - Function 27B/Get Job Origin

  DOO - COBOL Diagnostic Routines

  END - CEASE (SCT)

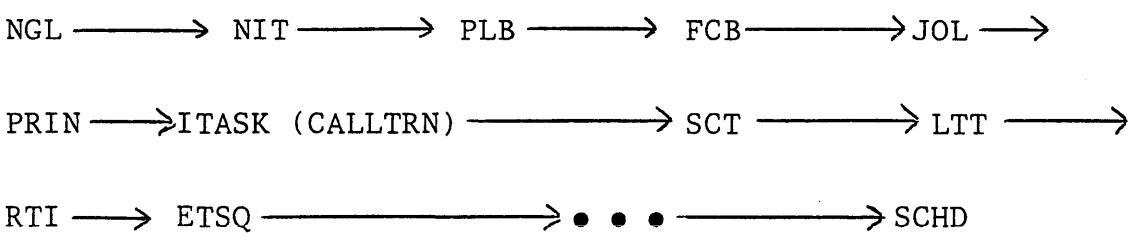
  TIM - System Time Functions 0-6

TASK SCHEDULING

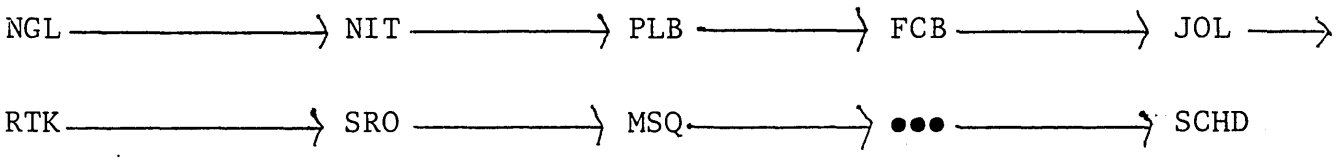
TAF SCHEDULING OF TASKS

- Scheduling Done When
  - Terminal Input Starts a Transaction.
  - Interactive Input (WAITINP) is Received.
  - WAIT Request Time Limit Expires.
  - TASK Calls Another Task.
  - TASK Ceases.
  - System Origin Transaction Generated
- Routine
  - SCHD

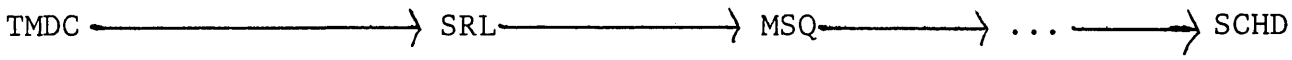
INITIAL TERMINAL INPUT



INTERACTIVE TERMINAL INPUT

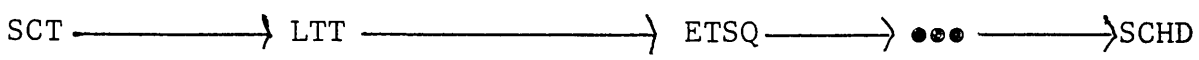


WAIT REQUEST TIME LIMIT

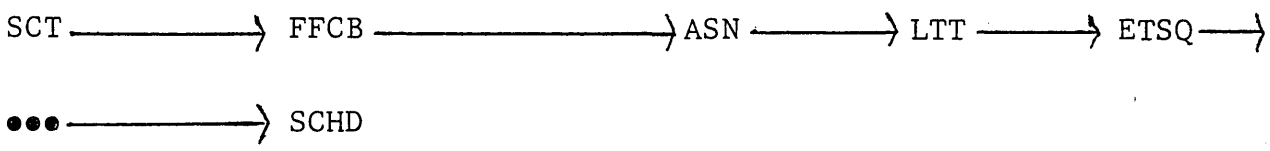


TASK CALLS ANOTHER TASK

- With Cease



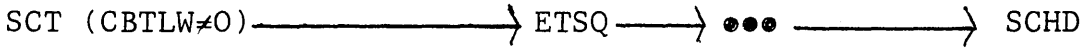
- Without Cease



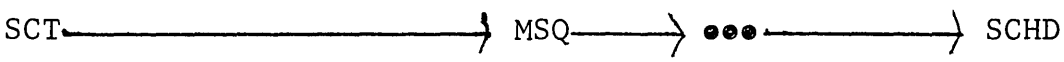
- With Return

TASK CEASE

- Additional Tasks to Schedule



- Task Was Initiated By CALLRTN



SYSTEM ORIGIN TRANSACTION

TMDC	CCS	RAN	ATK	PCI	CSM	IDL
CIACT)	(Rctask)	(CINAB)	(Sup/Msg)	(Rtask)	(Crmtask)	(CIIDL)

DSDM	PRE	PRE	SWITCH	MESSAGE	DIS
(Sysmsg)	(CIREC)	(Ctask)	(Kdis)	(Sysmsg)	(Crmtask)

SCHEDULER ROUTINE - SCHD

- RTL
- SCHDA
- RCP
- ROLT - Disk Address of Rolled Task
- RTL/ATL ——————→ Subcp Status Words
- (TL+5) ≠ 0 if (RO+5) ≠ 0
- (TPLW) = Task Library File Name or Rollout File FET Address
- RCPT - CR Word
- B3/B4 - Return Addresses
- Exit From SCHD
  - When TLS is full
  - NO FL for load
  - No Tasks in RTL

TASK ROLLOUT/ROLLIN PROCESSING

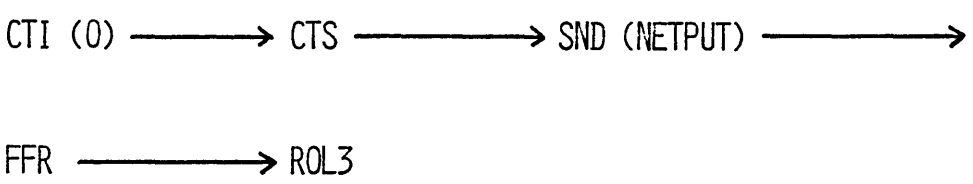
TASK ROLLOUT

- CAUSED BY:
  - WAITINP REQUEST
  - TERMINAL OUTPUT LIMIT EXCEEDED
  - CALLRTN REQUEST
  - WAIT REQUEST
  - TASK MEMORY REQUEST

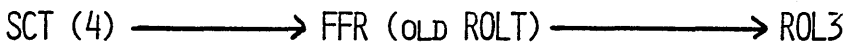
- ROUTINE
  - ROLL

WAITINP REQUEST

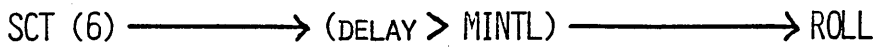
TERMINAL OUTPUT LIMIT EXCEEDED



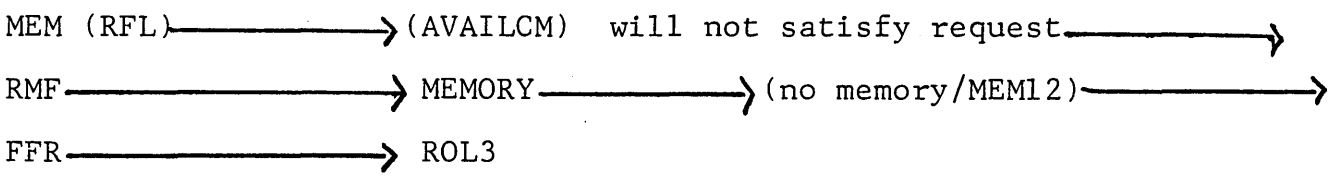
CALLRTN REQUEST



WAIT REQUEST



TASK MEMORY REQUEST



TASK ROLLOUT - ROLL

- FFR
- TRCL
  - ROLT Full
  - Wait for Delay Time
  - Rollout File Locked
  - Rollout File Full
  - During Rollout
- (X5) = Delay Before Rollout
- ROSC < 0 if Rollout Interrupted
  - TSSC
  - ROL15
- (RO+5) ≠ 0, When Rollout File Interlocked
- RRSC = ROLT reservation bit mask
- Reset CPU Priority to Original
- RSCC = Image of Subcp +1 Through Subcp +7
- Subcp Storage Move Lock Bit Set
- AVAILCM
- ESCP1

TAF MEMORY MANAGEMENT

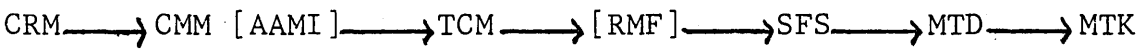
TAF MEMORY MANAGEMENT

- PROCESSES
  - Request subcontrol point (Subcp) Memory
  - CRM/CMM Memory Requests
  - Reduction of Field Length

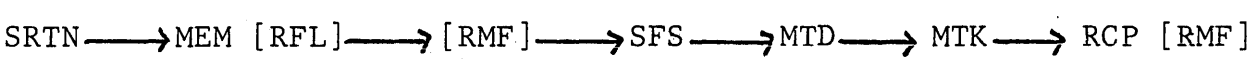
- ROUTINES
  - RMF - Request more field length
  - SFS - Search for free space
  - MTD - Move tasks down
  - MTK - Move task
  - RCP - Request Subcp memory
  - CORU - Core utilization check

REQUEST SUBCONTROL MEMORY

CRM-CMM MEMORY REQUESTS



TASK MEMORY REQUESTS



REDUCTION OF FIELD LENGTH

CORU called every CORTL seconds to:

- Release free space after last Subcp if FCMFL words or more [REDFL - maximum released]
- Evict releasable task with largest RA if last task released more than RRTTL milliseconds earlier
- Check for TAF idle ITRTL milliseconds to rollout part of its FL
- Check for idle down of TAF

CORU called to:

- Evict releaseable task if necessary to honor request for a subcontrol point table entry

DETECTION OF POTENTIALLY BLOCKED TASKS

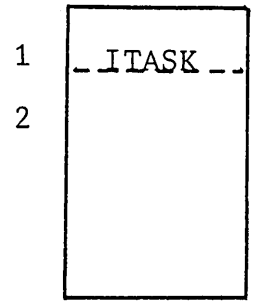
- A blocked task is one which is unable to obtain the central memory space needed for execution.

- Code in Common Deck - COMKDPB

- Checked for During
  - Initialization
  - Task Library Update (TT option)
  - K.MAXFL command processing

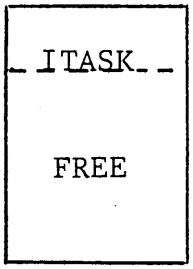
SCP MEMORY MANAGEMENT
Example 1

Sub-control Point Table
a. ITASK



VNACP 2, 3, etc.

Memory for
Sub-control Points

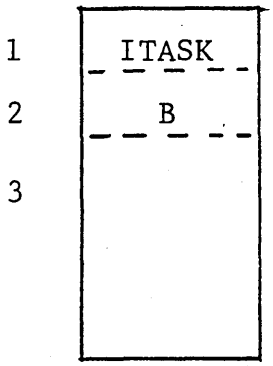


SCP MEMORY MANAGEMENT

Example 1 (cont)

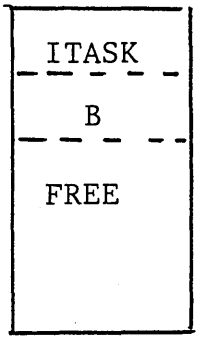
b. Scheduler requests memory for task B.

Sub-Control Point Table



VNACP 3, 4, etc.

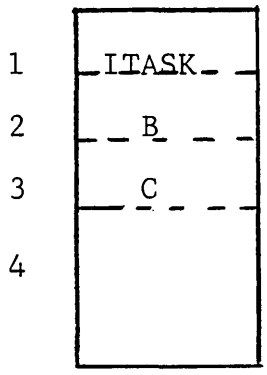
Memory for sub-control Points



Example 1 (cont)

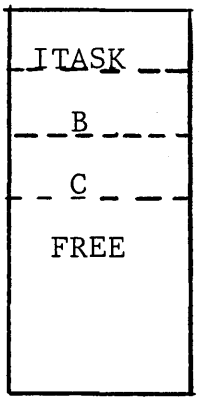
c. Scheduler requests memory for task c.

Sub-control Point Table



VNACP 4, 5 etc.

Memory for Sub-control Points

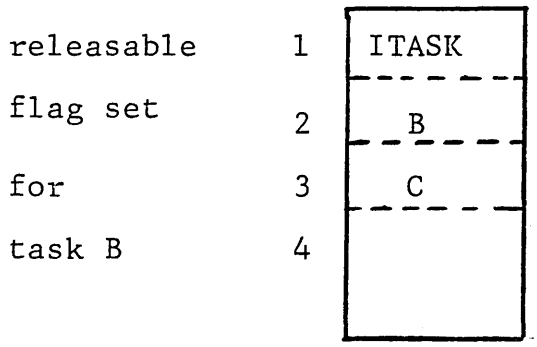


SCP MEMORY MANAGEMENT

Example 1 (cont)

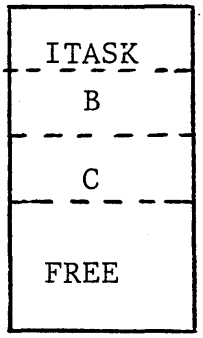
d. Task B ceases

Sub-control point table



VNACP 4 5 etc.

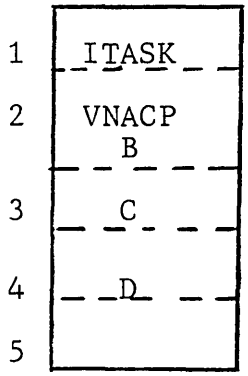
Memory for Sub-control Points



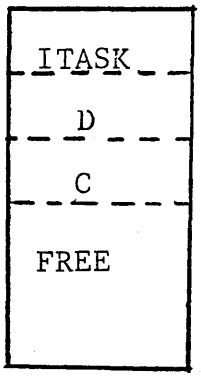
Example 1 (cont)

e. Scheduler requests memory for task D; task D fits in task B slot.

Sub-control point table



Memory for Sub-control Points

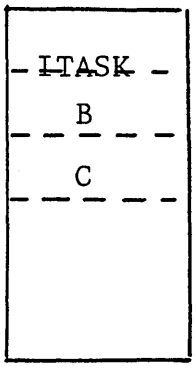


Sub-control point 2 is added to chain of free sub-control points pointed to by VNACP. First hole that fits task is chosen without storage move. VNACP = 2,5, etc.

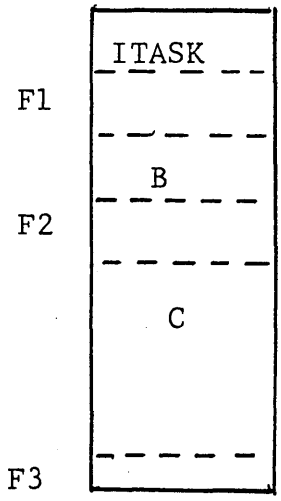
SCP MEMORY MANAGEMENT
Example 2

Assume following conditions:

1. Sub-control Point Table



Memory for sub-control points



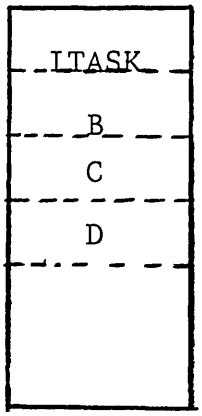
2. Scheduler requests memory for task D. Routine RCP will use hole(s) that require the least storage move.

3. Assume $D \leq F1+F2$
   $C > B$

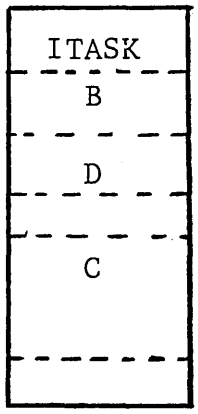
SCP MEMORY MANAGEMENT
Example 2 (cont)

Then B will be storage moved.

Sub-control Point Table



Memory for Sub-control Points



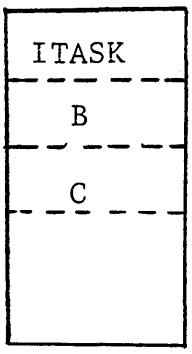
F4

F3

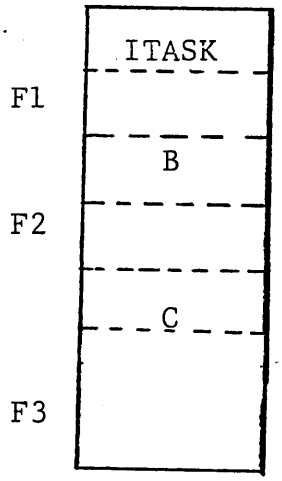
SCP MEMORY MANAGEMENT
Example 3

Assume following conditions:

1. Sub-control Point Table



Memory for sub-control points



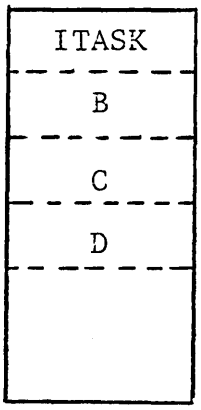
2. Scheduler requests memory for task D.

3. $D \leq F1+F2$
   $D \leq F2+F3$
   $C < B$

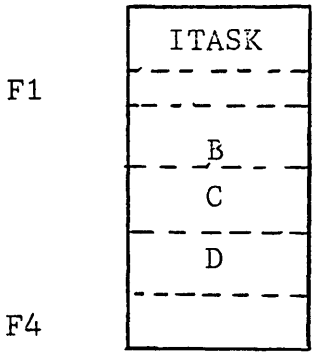
SCP MEMORY MANAGEMENT
Example 3 (cont)

Then C will be moved.

Sub-control Point Table



Memory for sub-control points

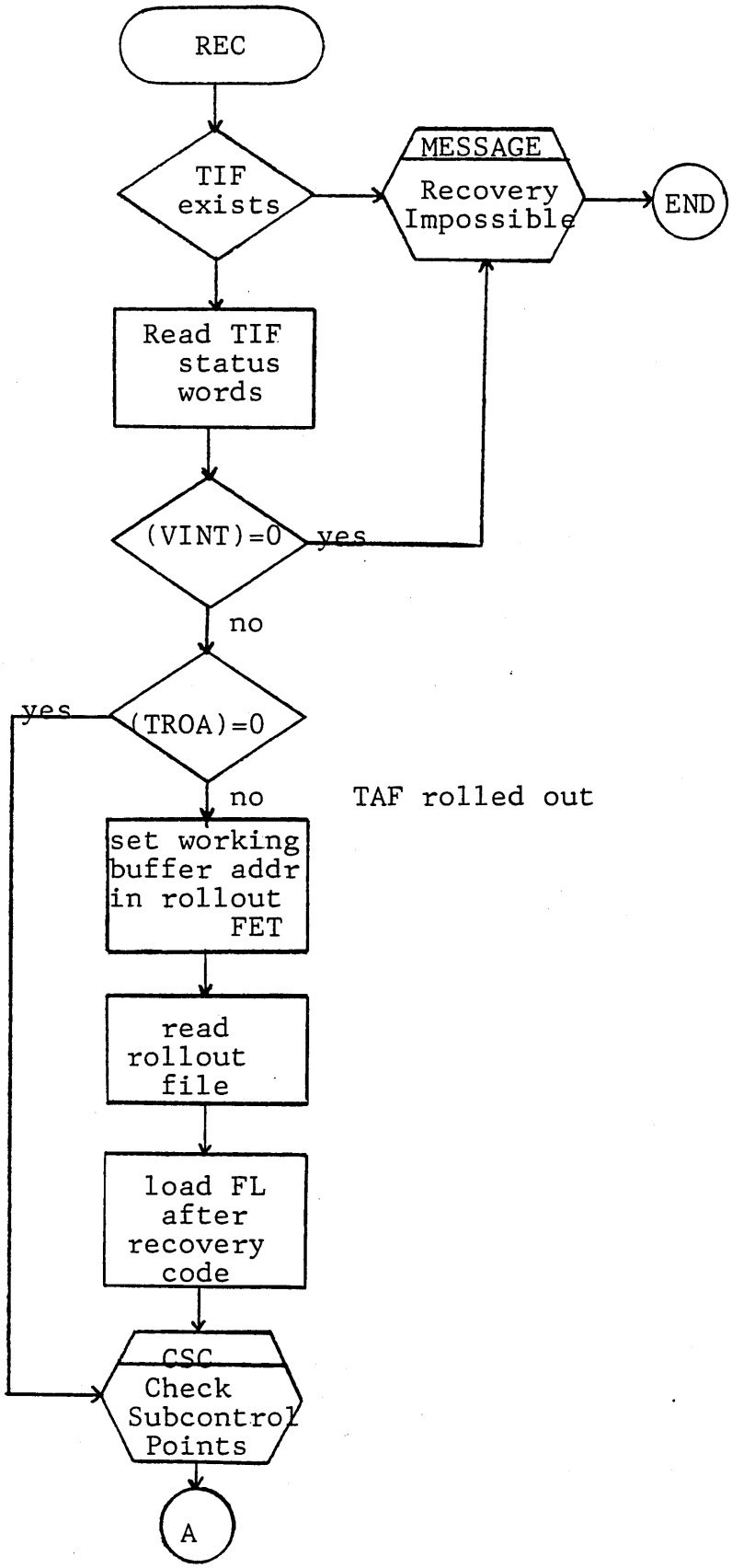


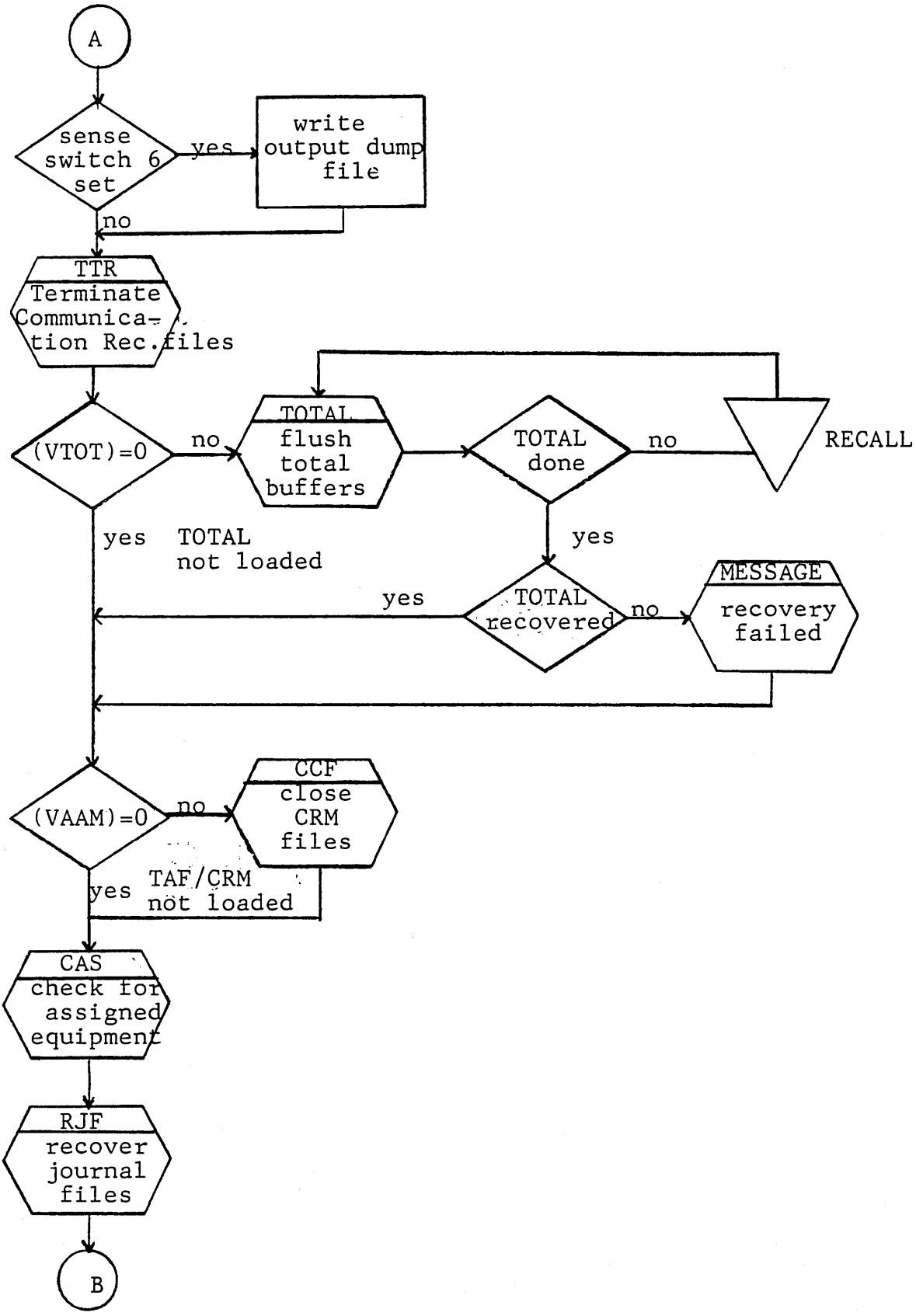
TAF TERMINATION

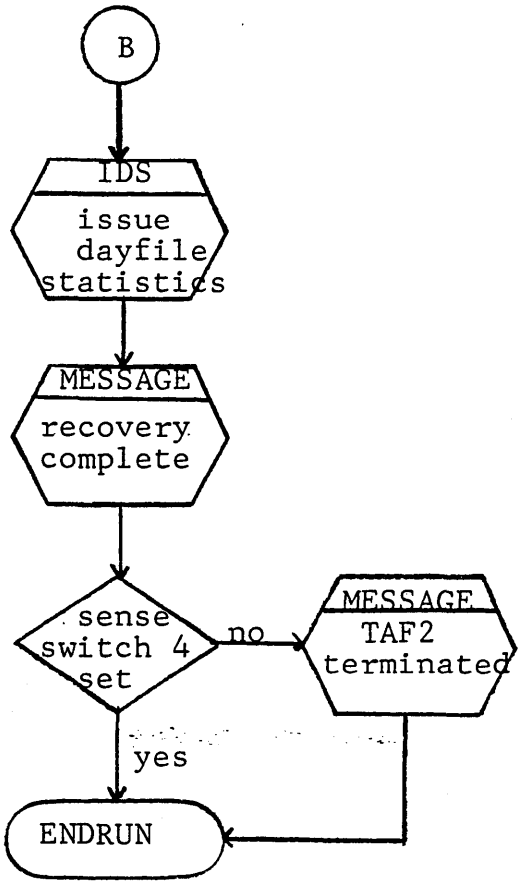
TAF TERMINATION

- Process
    1. Close Communication Recovery Files
    2. Call TOTAL (if used) to Flush Buffers
    3. Call AAMI (if used) to Close CRM and Recovery Files
    4. Flush Buffered Journal Files
    5. Issue Dayfile Statistics

- Routine ÷ REC

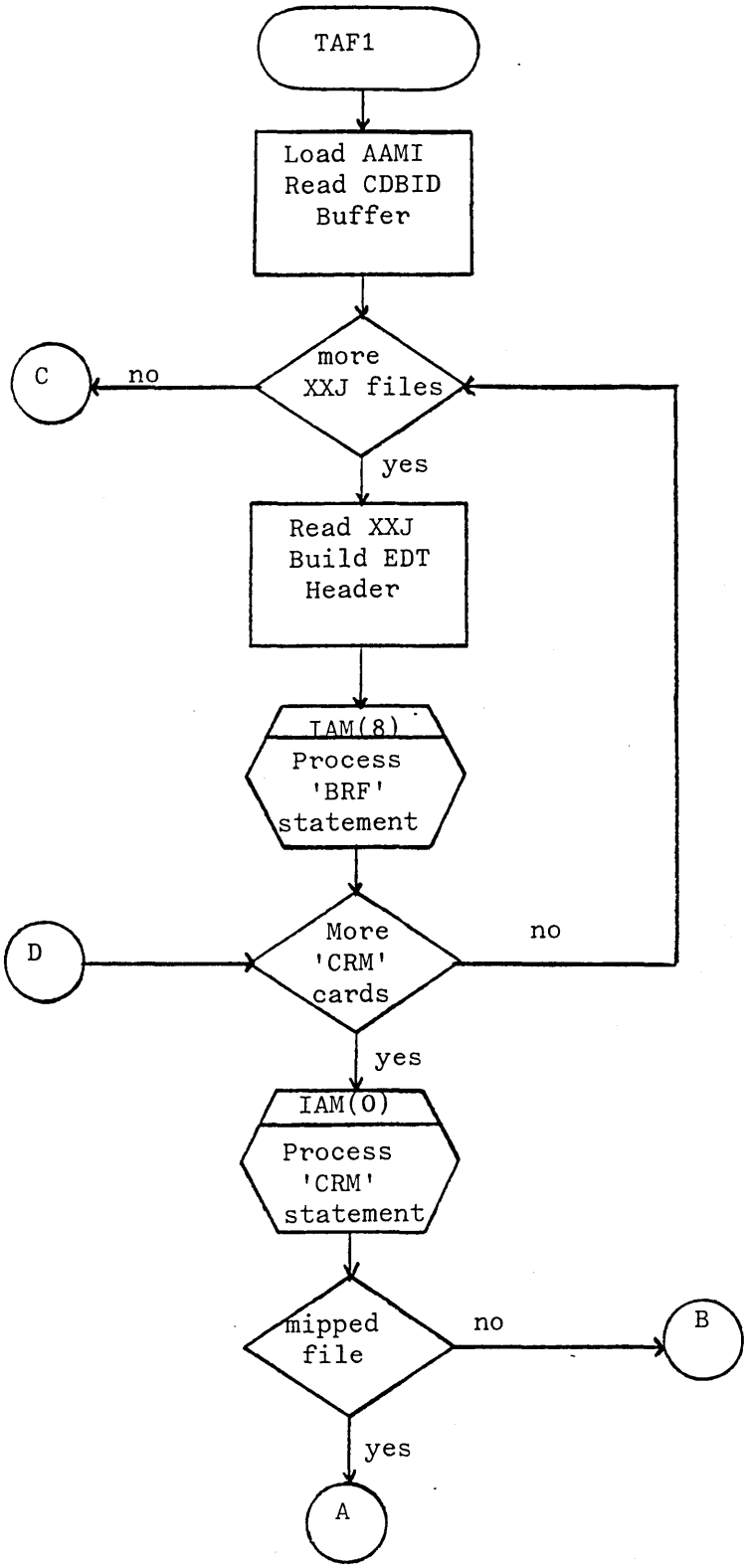

REC
TIF exists
MESSAGE
Recovery Impossible
END
Read TIF status words
(VINT)=0
yes
no
yes
(TROA)=0
no
TAF rolled out
set working buffer addr in rollout FET
read rollout file
load FL after recovery code
CSC
Check Subcontrol Points
A


A
sense switch 6 set
yes
write output dump file
no
TTR
Terminate Communica-tion Rec.files
(VTOT)=0
no
TOTAL
flush total buffers
TOTAL done
no
RECALL
yes
TOTAL not loaded
yes
TOTAL recovered
no
MESSAGE
recovery failed
yes
(VAAM)=0
no
CCF
close CRM files
yes
TAF/CRM not loaded
CAS
check for assigned equipment
RJF
recover journal files
B

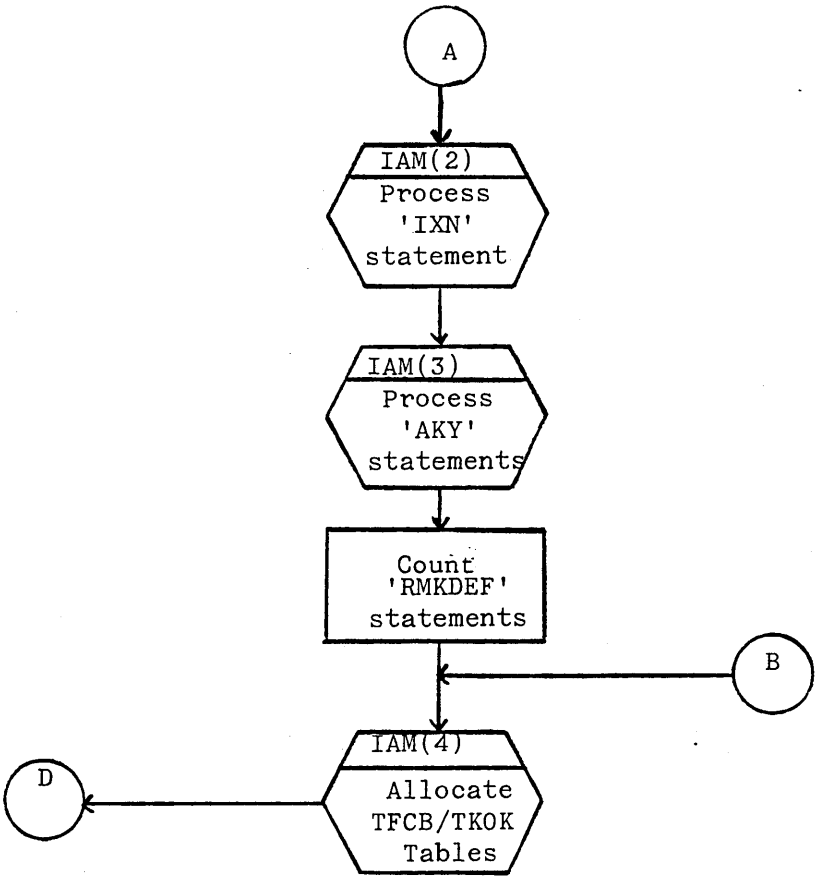

B
IDS
issue dayfile statistics
MESSAGE
recovery complete
sense switch 4 set
no
MESSAGE
TAF2 terminated
yes
ENDRUN

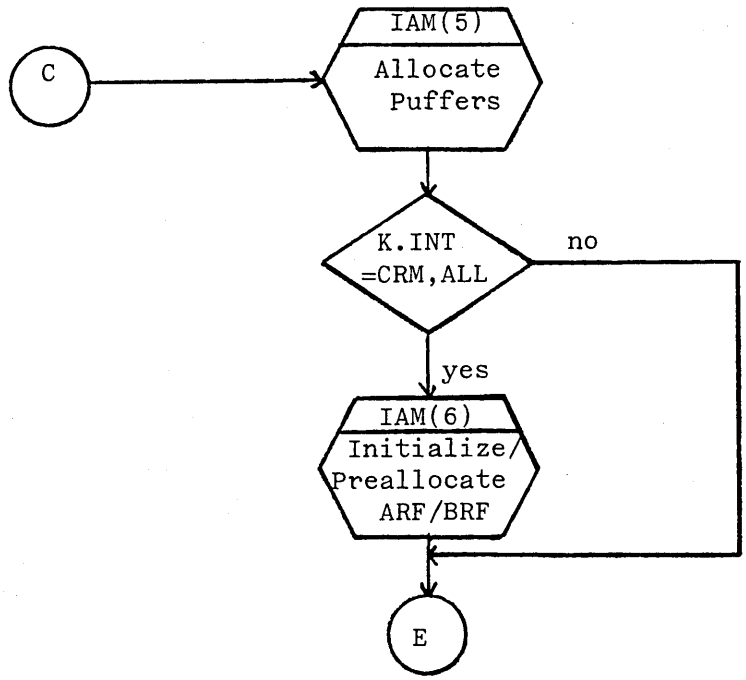
TAF/CRM

TAF/CRM Initialization

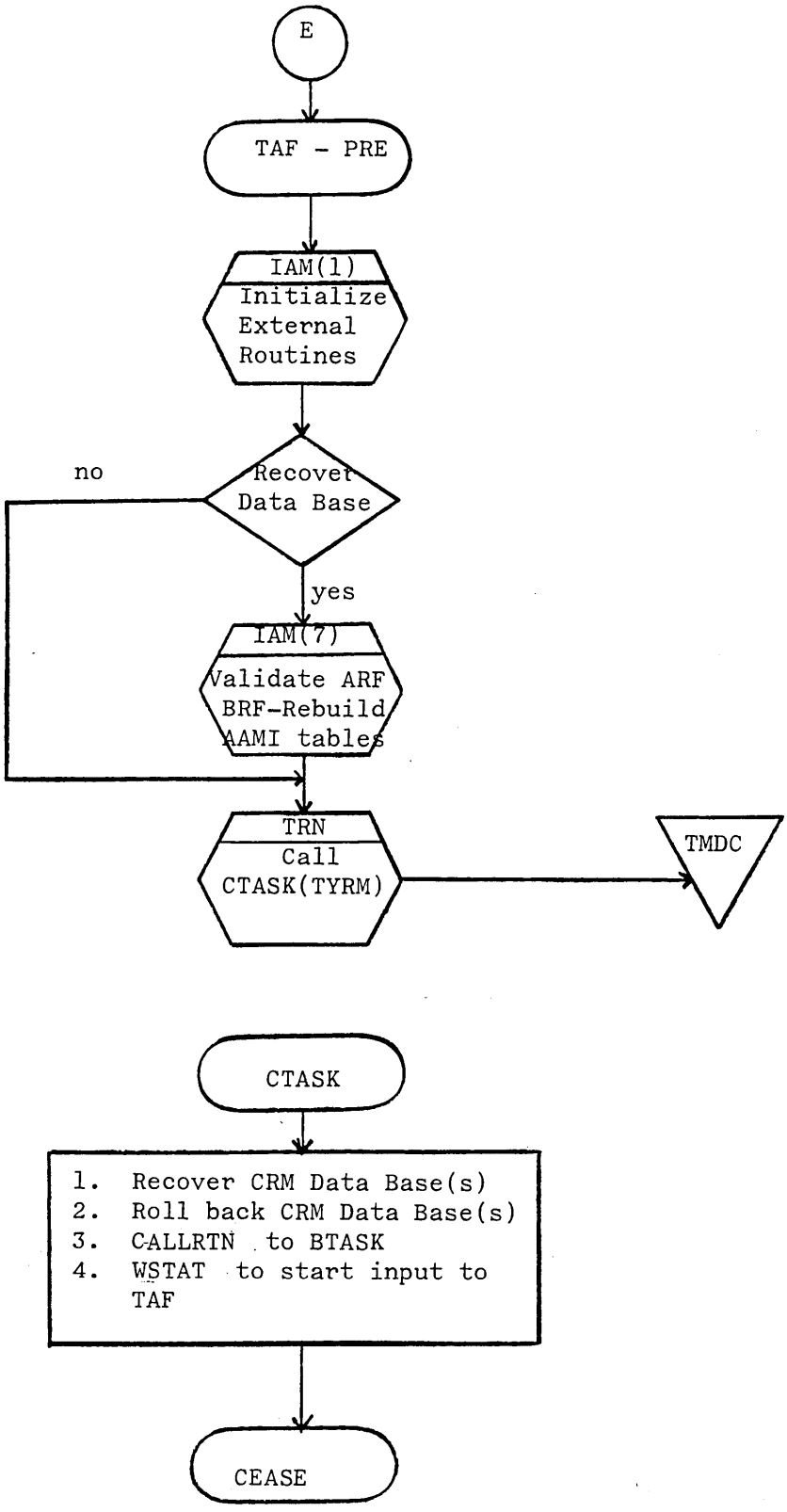


TAF/CRM Initialization
(continued)





TAF/CRM Initialization
(continued)

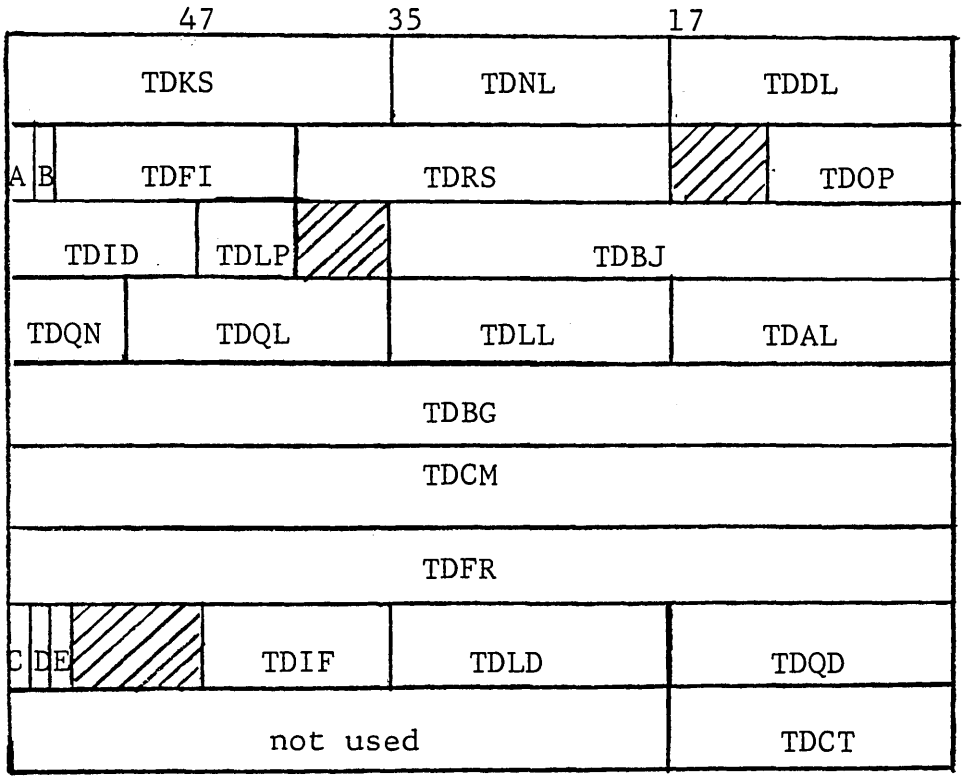


DATA BASE/RECOVERY FILE TABLE - TDRF

THIS TABLE CONTAINS CONTROLLING INFORMATION FOR DATA BASES AND LINKS TO THE RECOVERY FILES. STATISTICAL INFORMATION ABOUT THE DATA BASE AS WELL AS INTERFACE INFORMATION FOR THE BATCH RECOVERY UTILITY, DMREC, IS ALSO KEPT IN THIS TABLE. THERE WILL BE ONE TDRF CREATED FOR EACH BRF STATEMENT ENCOUNTERED ON THE XXJ FILES USED TO INSTALL TAF/CRM DATA BASES (ONE TDRF PER DATA BASE).

THE ADDRESS OF THE FIRST TDRF IS FOUND IN THE WORD REFERENCED BY TAG RDRT (LOW CORE OF AAMI). EACH TDRF POINTS TO THE NEXT TDRF FOR THE NEXT DATA BASE THAT IS ACTIVE WITH TAF/CRM.

Data Base/Recovery File Table - TDRF



DATA BASE/RECOVERY FILE TABLE - TDRF

```
**        TDRF - DATA BASE AND RECOVERY FILE TABLE.
*
*T  W1    24/ TDKS,18/ TDNL,18/ TDDL
*T, W2    1/A,1/B,16/ TDFI,24/ TDRS,6/,12/ TDOP
*T, W3    12/ TDID,6/ TDLP,6/,36/ TDBJ
*T, W4    6/ TDQN,18/ TDQL,18/ TDLL,18/ TDAL
*T, W5    60/ TDBG
*T, W6    60/ TDCM
*T, W7    60/ TDFR
*T, W8    1/C,1/D,1/E,9/,12/ TDIF,18/ TDLD,18/ TDQD
*T, W9    42/,18/ TDCT
*
*             TDKS - LARGEST KEY LENGTH IN CHARACTERS.
*             TDNL - FWA OF FIRST *TLNT* FOR DATA BASE FILE.
*             TDDL - FWA OF NEXT *TDRF* TABLE.
*          A = TDSD - DATA BASE IS DOWN IF *TDSD* .EQ. 1.
*          B = TDSI - DATA BASE IS IDLE IF *TDSI* .EQ. 1.
*             TDFI - NUMBER OF *FIT* *FWI-S* CHANGED VIA *DLX*.
*             TDRS - MAXIMUM RECORD SIZE IN CHAR-S FOR DATA BASE.
*             TDOP - CURRENT OPEN FILE COUNT.
*             TDID - DATA BASE IDENTIFIER.
*             TDLP - LAST CHARACTER OF LOCAL *ARF*.
*             TDBJ - OUTSTANDING BATCH JOB SEQUENCE NUMBERS,
*                    18/ TDJA,18/ TDJB.
*             TDJA - OUTSTANDING BATCH JOB SEQUENCE NUMBER 1.
*             TDJB - OUTSTANDING BATCH JOB SEQUENCE NUMBER 2.
*             TDQN - NUMBER OF *BRF-S* FOR DATA BASE.
*             TDQL - FWA OF FIRST *TBRF* FOR THE DATA BASE.
*             TDLL - FWA OF LAST *TLNT* FOR DATA BASE FILE.
*             TDAL - FWA OF *TARF* FOR DATA BASE.
*             TDBC - NUMBER OF *DBEGIN* REQUESTS PROCESSED FOR DB.
*             TDCM - NUMBER OF *DBCOMIT* REQUESTS PROCESSED FOR DB.
*             TDFR - NUMBER OF *DBFREE* REQUESTS PROCESSED FOR DB.
*          C = TDRQ - DATA BASE DOWN TO RECOVER *BRF* IF .EQ. 1.
*          D = TDRL - DATA BASE DOWN TO RECOVER *ARF* IF .EQ. 1.
*          E = TDOD - DATA BASE DOWN BY OPERATOR IF .EQ. 1.
*             TDIF - COUNT OF IDLED DATA BASE FILES.
*             TDLD - FWA OF *TLNT* DOWN FOR BATCH RECOVERY.
*             TDQD - FWA OF *TBRF* OF FIRST DOWN *BRF*.
*             TDCT - COUNT OF ACTIVE TRANSACTIONS USING RECOVERY.
```

DATA BASE/RECOVERY FILE TABLE - TDRF
(CONTINUED)

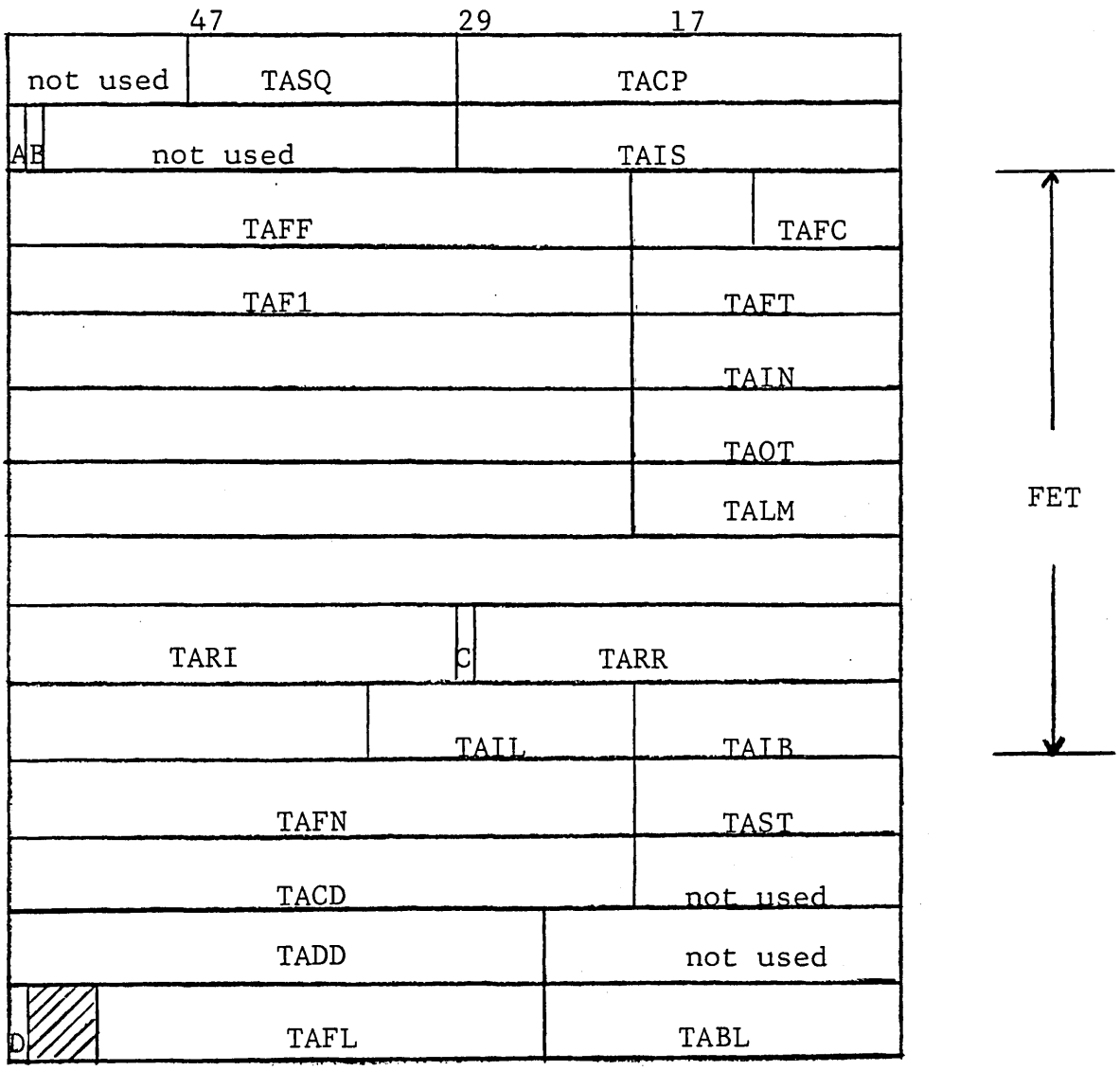
```
TDKS     FIELD  0,59,36     LARGEST KEY LENGTH IN CHARACTERS
TDNL     FIELD  0,35,18     FWA OF FIRST *TLNT* FOR DB FILE
TDDL     FIELD  0,17,0      FWA OF NEXT *TDRF* TABLE
TDSD     FIELD  1,59,59     DATA BASE DOWN IF *TDSD* .EQ. 1
TDSI     FIELD  1,58,58     DATA BASE IDLE IF *TDSI* .EQ. 1
TDFI     FIELD  1,57,42     NUMBER OF *FIT* *FWI-S* CHANGED VIA *DLX*
TDRS     FIELD  1,41,18     MAX RECORD SIZE IN CHAR-S FOR DATA BASE
TDOP     FIELD  1,11,0      CURRENT OPEN FILE COUNT
TDID     FIELD  2,59,48     DATA BASE IDENTIFIER
TDLP     FIELD  2,47,42     LAST CHARACTER OF LOCAL *ARF*
TDLB     FIELD  2,42,42     LAST BIT OF ACTIVE *ARF* NAME (0=01, 1=02)
TDBJ     FIELD  2,35,0      DB BATCH RECOVERY JOB SEQUENCE NUMBERS
TDJA     FIELD  2,35,18     BATCH JOB SEQUENCE NUMBER 1
TDJB     FIELD  2,17,0      BATCH JOB SEQUENCE NUMBER 2
TDQN     FIELD  3,59,54     NUMBER OF *BRF-S* FOR DATA BASE
TDQL     FIELD  3,53,36     FWA OF FIRST *TBRF* FOR DATA BASE
TDLL     FIELD  3,35,18     FWA OF LAST *TLNT* FOR DATA BASE FILE
TDAL     FIELD  3,17,0      FWA OF *TARF* FOR DATA BASE
TDBG     FIELD  4,59,0      NUMBER OF *DBEGIN* REQUESTS PROCESSED
TDCM     FIELD  5,59,0      NUMBER OF *DBCOMIT* REQUESTS PROCESSED
TDFR     FIELD  6,59,0      NUMBER OF *DBFREE* REQUESTS PROCESSED
TDRQ     FIELD  7,59,59     DATA BASE DOWN TO RECOVER *BRF*
TDRL     FIELD  7,58,58     DATA BASE DOWN TO RECOVER *ARF*
TDOD     FIELD  7,57,57     DATA BASE DOWN BY OPERATOR
TDIF     FIELD  7,47,36     COUNT OF IDLED DATA BASE FILES
TDLD     FIELD  7,35,18     FWA OF *TLNT* DOWN FOR RECOVERY
TDQD     FIELD  7,17,0      FWA OF *TBRF* OF FIRST DOWN *BRF*
TDCT     FIELD  8,17,0      COUNT OF TRANSACTIONS USING RECOVERY
TDRFE    EQU    9           LENGTH OF *TDRF* ENTRY
```

AFTER IMAGE RECOVERY FILE TABLE - TARF

THIS TABLE CONTAINS CONTROLLING INFORMATION FOR AFTER IMAGE RECOVERY FILE (ARF) PROCESSING. THE FET AS WELL AS AN IMAGE OF THE ARF HEADER IS PART OF THIS TABLE. THERE IS ONE TARF FOR EACH DATA BASE.

THIS FIRST WORD ADDRESS OF THIS TABLE FOR A PARTICULAR DATA BASE IS CONTAINED IN THE TDRF FOR THAT DATA BASE IN FIELD TDAL.

After Image Recovery File Table - TARF



AFTER IMAGE RECOVERY FILE TABLE - TARF

```
**        TARF - AFTER IMAGE RECOVERY FILE TABLE.
*
*T  W1    12/,18/ TASQ,30/ TACP
*T, W2    1/A,1/B,28/,30/ TAIS
*T, W3    42/ TAFF,8/0,10/ TAFC
*T, W4    42/ TAF1,18/ TAFT
*T, W5    42/0,18/ TAIN
*T, W6    42/0,18/ TAOT
*T, W7    42/0,18/ TALM
*T, W8    60/0
*T, W9    30/ TARI,1/C,29/ TARR
*T, W10   24/0,18/ TAIL,18/ TAIB
*T, W11   42/ TAFN,18/ TAST
*T, W12   36/ TACD,24/
*T, W13   36/ TADD,24/
*T, W14   1/D,5/,30/ TAFL,24/ TABL
*
*                 TASQ - FWA OF *TSEQ* ENTRY RESERVING *TARF*.
*                 TACP - UNUSED PRU COUNT.
*             A = TADN - *ARF* DOWN IF *TADN* .EQ. 1.
*             B = TAFB - *ARF* BUFFER FLUSHED IF *TAFB* .EQ. 1.
*                 TAIS - MAXIMUM AFTER IMAGE RECORD SIZE IN WORDS.
* FET+0           TAFF - LOCAL FILE ANME, FIRST WORD OF *ARF* FET.
*                 TALP - LAST CHARACTER OF *ARF* NAME.
*                 TALB - LAST BIT OF *ARF* NAME.
*                 TAFC - REQUEST/RETURN CODE.
* FET+1           TAF1 - 42 BITS OF FET+1 = 12/,1/R,1/,1/,1/E,20/,6/L.
*                        R - SET TO 1 FOR RANDOM PROCESSING.
*                        E - SET TO 1 FOR USER PROCESSING OF ERRORS.
*                        L - SET TO 3 FOR 8 WORD FET.
*                 TAFT - FIRST.
* FET+2           TAIN - IN.
* FET+3           TAOT - OUT.
* FET+4           TALM - LIMIT.
* FET+6           TARI - CURRENT RANDOM INDEX.
*             C = TARW - WRITE IN-PLACE IF *TARW* .EQ. 1.
*                 TARR - RELATIVE SECTOR ADDRESS FOR RANDOM I/O.
* FET+7           TAIL - INDEX LENGTH.
*                 TAIB - FWA OF INDEX BUFFER.
* ARF HEADER+0    TAFN - AFTER IMAGE RECOVERY FILE NAME.
*                 TAST - AFTER IMAGE RECOVERY FILE STATUS.
* ARF HEADER+1    TACD - AFTER IMAGE RECOVERY FILE CREATION
*                        DATE AND TIME.
* ARF HEADER+2    TADD - DATE ANE TIME *ARF* DUMPED.
* ARF HEADER+3    TAFL - LENGTH OF *ARF* IN PRU-S (*CRMARFN*).
*                 TABL - MAXIMUM BLOCK LENGTH.
*             D = TAD1 - FIRST *ARF* DUMP IF *TAD1* .EQ. 1
```

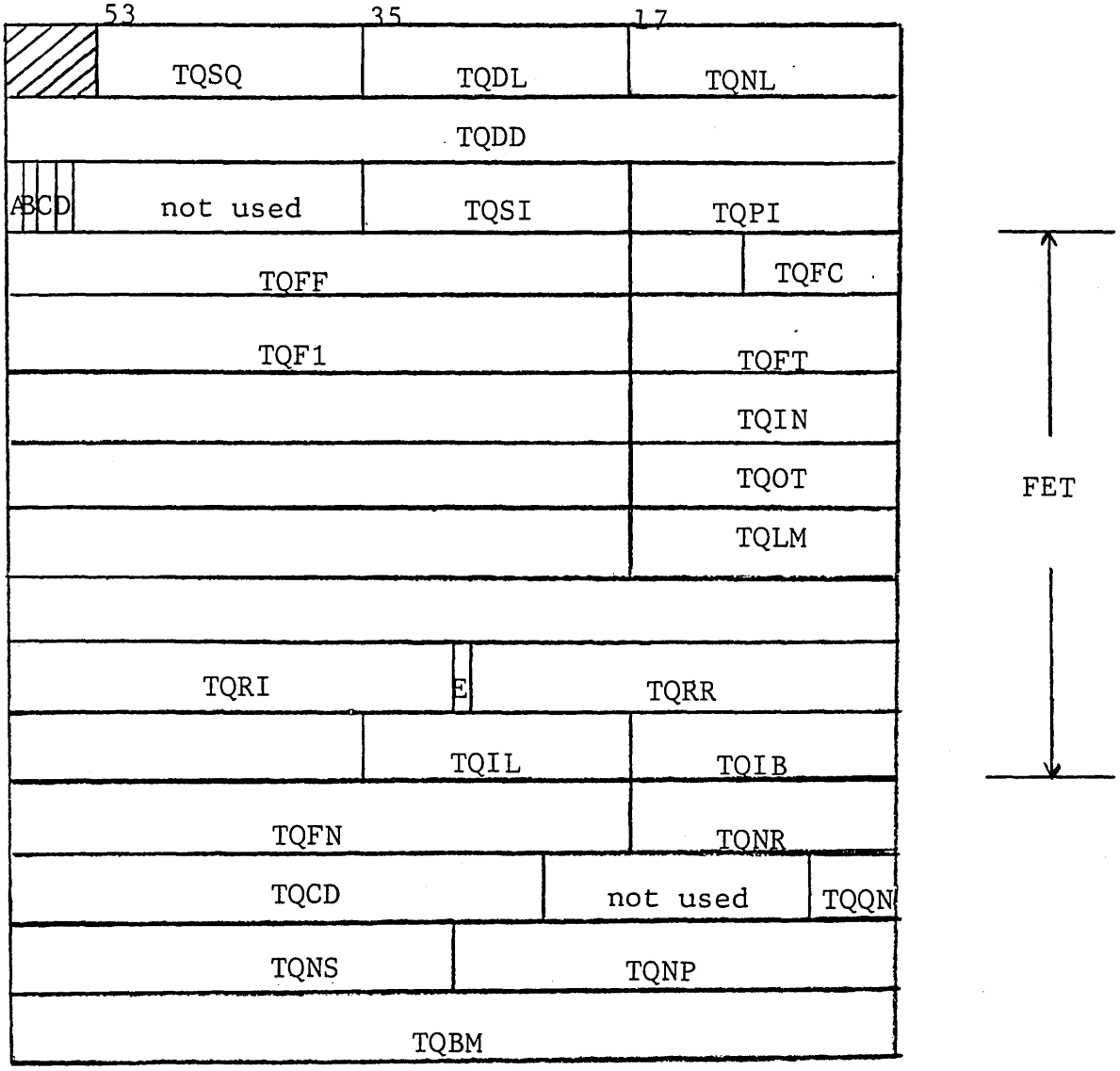
```
TASQ    FIELD  0,47,30     FWA OF *TSEQ* ENTRY RESERVING *TARF*
TACP    FIELD  0,29,0      UNUSED PRU COUNT
TADN    FIELD  1,59,59     *ARF* DOWN IF *TADN* .EQ. 1
TAFB    FIELD  1,58,58     *ARF* BUFFER FLUSHED IF *TAFB* .EQ. 1
TAIS    FIELD  1,29,0      MAXIMUM AFTER IMAGE RECORD SIZE IN WORDS
TAFF    FIELD  2,59,18     LFN, FIRST WORD OF *ARF* RANDOM FET
TALP    FIELD  2,23,18     LAST CHARACTER OF *ARF* NAME
TALB    FIELD  2,18,18     LAST BIT OF *ARF* NAME
TAFC    FIELD  2,9,0       REQUEST/RETURN CODE
TAF1    FIELD  3,59,18     FET+1 UPPER 42 BITS
TAFT    FIELD  3,17,0      FIRST
TAIN    FIELD  4,17,0      IN
TAOT    FIELD  5,17,0      OUT
TALM    FIELD  6,17,0      LIMIT
TARI    FIELD  8,59,30     CURRENT RANDOM INDEX
TARW    FIELD  8,29,29     WRITE IN-PLACE IF *TARW* .EQ. 1
TARR    FIELD  8,28,0      RELATIVE SECTOR ADDRESS FOR RANDOM WRITE
TAIL    FIELD  9,35,18     INDEX LENGTH
TAIB    FIELD  9,17,0      FWA OF INDEX BUFFER
TAFN    FIELD  10,59,18    AFTER IMAGE RECOVERY FILE NAME
TAST    FIELD  10,17,0     AFTER IMAGE RECOVERY FILE STATUS
TACD    FIELD  11,59,24    *ARF* CREATION DATE AND TIME, PACKED
TADD    FIELD  12,59,24    DATE AND TIME *ARF* DUMPED, PACKED
TAFL    FIELD  13,53,24    LENGTH OF *ARF* IN PRU-S (CRMARFN)
TABL    FIELD  13,23,0     MAXIMUM BLOCK LENGTH
TAD1    FIELD  13,59,59    FIRST *ARF* DUMP IF *TAD1* .EQ. 1
TARFE   EQU    14          LENGTH OF *TARF* ENTRY
TAHDL   EQU    4           *ARF* HEADER LENGTH
TARHL   EQU    6           *ARF* RECORD HEADER LENGTH
```

BEFORE IMAGE RECOVERY FILE TABLE - TBRF

THIS TABLE CONTAINS CONTROLLING INFORMATION FOR BEFORE IMAGE RECOVERY FILE (BRF) PROCESSING. THE FET AS WELL AS AN IMAGE OF THE BRF HEADER IS PART OF THIS TABLE. THERE IS ONE TBRF FOR EACH BEFORE IMAGE RECOVERY FILE DEFINED FOR TAF/CRM.

THE FIRST WORD ADDRESS OF THE FIRST TBRF FOR A PARTICULAR DATA BASE IS CONTAINED IN THE TDRF FOR THAT DATA BASE IN FIELD TDQL.

Before Image Recovery File Table - TBRF



```
**        TBRF - BEFORE IMAGE RECOVERY FILE TABLE.
*
*T  W1     6/,18/ TQSQ,18/ TQDL,18/ TQNL
*T, W2     60/ TQDD
*T, W3     1/A,1/B,1/C,1/D,20/,18/ TQSI,18/ TQPI
*T, W4     42/ TQFF,8/0,10/ TQFC
*T, W5     42/ TQF1,18/ TQFT
*T, W6     42/0,18/ TQIN
*T, W7     42/0,18/ TQOT
*T, W8     42/0,18/ TQLM
*T, W9     60/0
*T, W10    30/ TQRI,1/E,29/ TQRR
*T, W11    24/0,18/ TQIL,18/ TQIB
*T, W12    42/ TQFN,18/ TQNR
*T, W13    36/ TQCD,18/,6/ TQQN
*T, W14    30/ TQNS,30/ TQNP
*T, W15    60/ TQBM
*T, W15+N  60/ TQBM
*
*                 TQSQ - FWA OF *TSEQ* ENTRY RESERVING *TBRF*.
*                 TQDL - FWA OF DATA BASE RECOVERY FILE TABLE *TDRF*.
*                 TQNL - FWA OF NEXT *TBRF*.
*                 TQDD - DATE/TIME *BRF* WENT DOWN (PACKED).
*            A = TQST - BEFORE IMAGE RECOVERY FILE DOWN.
*                       IF *TQST* .EQ. 1.
*            B = TQEA - EXCLUSIVE ACCESS TO *BRF* IF *TQEA* .EQ. 1.
*            C = TQBI - BEFORE IMAGE WRITE PENDING IF *TQBI* .EQ. 1.
*            D = TQDI - *BRF* DOWN STAMP WRITTEN IF *TQDT* .EQ. 1.
*                 TQSI - FWA OF *TSEQ* OF TASK WRITING BEFORE IMAGE.
*                 TQPI - NUMBER OF PRU-S PER BEFORE IMAGE RECORD.
* FET+0           TQFF - LOCAL FILE NAME, FIRST WORD OF *BRF* FET.
*                 TQFC - REQUEST/RETURN CODE.
* FET+1           TQF1 - 42 BITS OF FET+1 = 12/,1/R,1/,1/,1/E,20/,6/L.
*                        R - SET TO 1 FOR RANDOM PROCESSING.
*                        E - SET TO 1 FOR USER PROCESSING OF ERRORS.
*                        L - SET TO 3 FOR 8 WORD FET.
*                 TQFT - FIRST.
* FET+2           TQIN - IN.
* FET+3           TQOT - OUT.
* FET+4           TQLM - LIMIT.
* FET+6           TQRI - CURRENT RANDOM INDEX.
*            E = TQRW - WRITE IN-PLACE IF *TQRW* .EQ. 1.
*                 TQRR - RELATIVE SECTOR ADDRESS FOR RANDOM I/O.
* FET+7           TQIL - INDEX LENGTH.
*                 TQIB - FWA OF INDEX BUFFER.
* BRF HEADER+0    TQFN - BEFORE IMAGE RECOVERY FILE NAME.
*                 TQNR - NUMBER OF RECORDS PER SEGMENT (*CRMUPM*).
* BRF HEADER+1    TQCD - *BRF* CREATION DATE AND TIME.
*                 TQQN - NUMBER OF *BRF-S* ASSIGNED DATE BASE.
* BRF HEADER+2    TQNS - NUMBER OF SEGMENTS ON BRF (*CMDM*).
*                 TQNP - NUMBER OF PRU-S PER SEGMENT.
*                 TQBM - SEGMENT ALLOCATION BIT MAP - 1 OR 2 WORDS LONG
*                        DEPENDING ON *CMDM* PARAMETER.
```

BEFORE IMAGE RECOVERY FILE TABLE - TBRF
(CONTINUED)

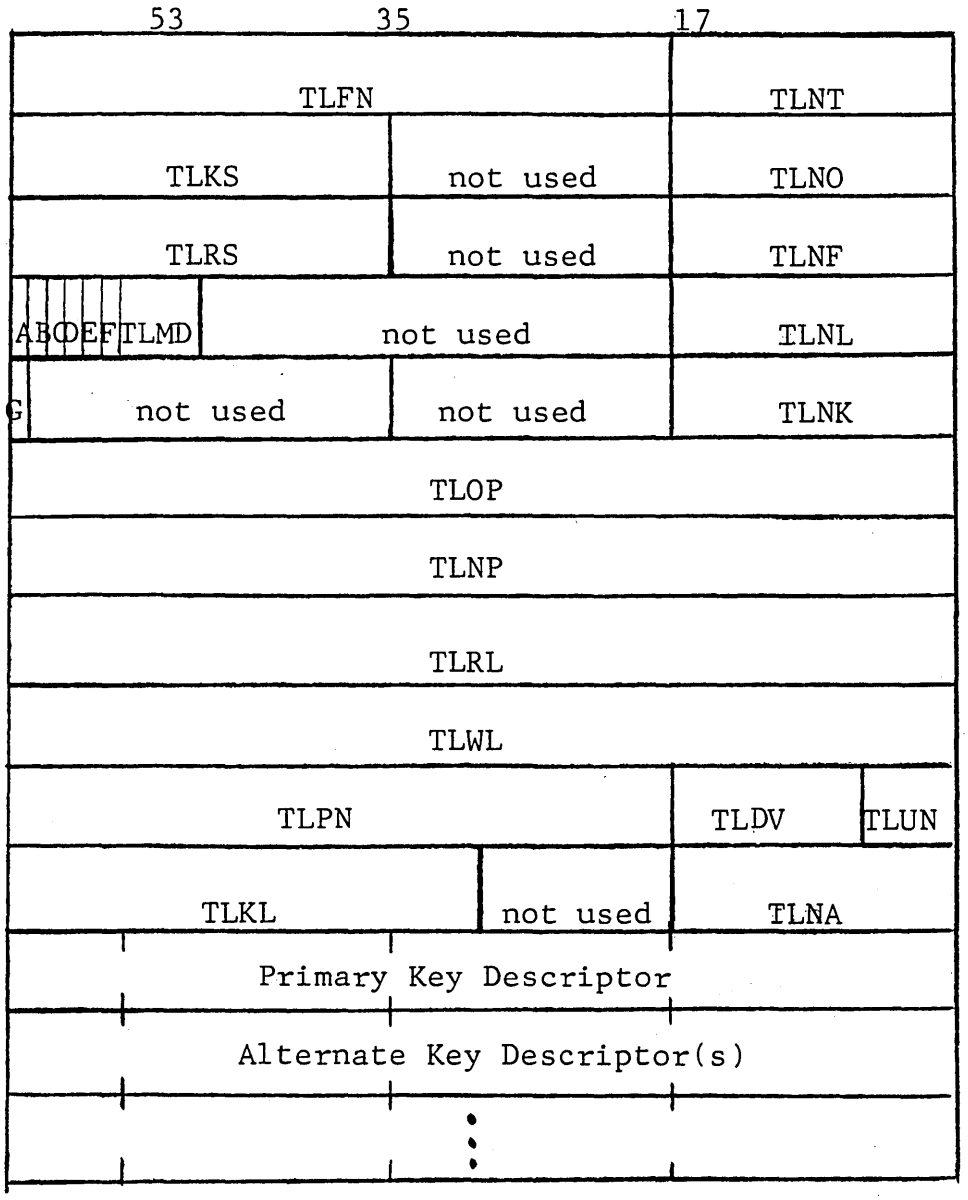
```
TQSQ      FIELD  0,53,36      FWA OF *TSEQ* ENTRY RESERVING *TQRF*
TQDL      FIELD  0,35,18      FWA OF *TDRF* TABLE
TQNL      FIELD  0,17,0       FWA OF NEXT *TBRF*
TQDD      FIELD  1,59,0       DATE/TIME *BRF* WENT DOWN (PACKED)
TQST      FIELD  2,59,59      *BRF* DOWN IF *TQST* .EQ. 1
TQEA      FIELD  2,58,58      TASK HAS EXCLUSIVE ACCESS IF *TQEA* .EQ. 1
TQBI      FIELD  2,57,57      BEFORE IMAGE PENDING IF *TQBI* .EQ. 1
TQDI      FIELD  2,56,56      *BRF* DOWN STAMP WRITTEN ON *ARF* IF SET
TQSI      FIELD  2,35,18      FWA OF *TSEQ* OF TASK WRITING BEFORE IMAGE
TQPI      FIELD  2,17,0       PRU-S PER BEFORE IMAGE RECORD
TQFF      FIELD  3,59,18      LFN, FIRST WORD OF *BRF* RANDOM FET
TQFC      FIELD  3,9,0        REQUEST/RETURN CODE
TQF1      FIELD  4,59,18      FET+1 UPPER 42 BITS
TQFT      FIELD  4,17,0       FIRST
TQIN      FIELD  5,17,0       IN
TQOT      FIELD  6,17,0       OUT
TQLM      FIELD  7,17,0       LIMIT
TQRI      FIELD  9,59,30      CURRENT RANDOM INDEX
TQRW      FIELD  9,29,29      WRITE IN-PLACE IF *TQRW* .EQ. 1
TQRR      FIELD  9,28,0       RELATIVE SECTOR ADDRESS FOR RANDOM WRITE
TQIL      FIELD  10,35,18     INDEX LENGTH
TQIB      FIELD  10,17,0      FWA OF INDEX BUFFER
TQFN      FIELD  11,59,18     BEFORE IMAGE RECOVERY FILE NAME
TQNR      FIELD  11,17,0      NUMBER OF RECORDS PER SEGMENT (*CRMUPM*)
TQCD      FIELD  12,59,24     *BRF* CREATION DATE AND TIME, PACKED
TQQN      FIELD  12,5,0       NUMBER OF *BRF-S* ASSIGNED DATA BASE
TQNS      FIELD  13,59,30     NUMBER OF SEGMENTS (*CMDM*)
TQNP      FIELD  13,29,0      NUMBER OF PRU-S PER SEGMENT
TQBM      FIELD  14,59,0      SEGMENT ALLOCATION BIT MAP FIRST WORD

.TQBML    EQU    CMDM+59      NUMBER OF BITS REQUIRED FOR BIT MAP
.TQRFE    EQU    .TQBML/60    NUMBER OF WORDS REQUIRED FOR *TQBM*
TQRFE     EQU    TQBMW+.TQRFE   LENGTH OF *TBRF* ENTRY
TQHDL     EQU    3            *BRF* HEADER LENGTH
TQRHL     EQU    7            *BRF* RECORD HEADER LENGTH
```

LOGICAL NAME TABLE - TLNT

THIS TABLE CONTAINS CONTROLLING AND STATISTICAL INFORMATION FOR A DATA BASE FILE. IT CONTAINS ALL THE ALTERNATE KEY DESCRIPTORS FOR A MULTIPLE INDEX FILE. THERE IS ONE OF THESE TABLES CREATED FOR EVERY FILE IN EVERY DATA BASE.

LOW CORE POINTER WORD VAMB (BITS 24-41) CONTAINS THE FIRST WORD ADDRESS OF THE FIRST TLNT FOR TAF/CRM. POINTERS TO THE FIRST AND LAST TLNT FOR THE DATA BASE ARE KEPT IN TDRF FIELDS TDNL AND TDLL, RESPECTIVELY.

Logical Name Table - TLNT



```
**          TLNT - LOGICAL NAME TABLE.
*
*T  W1      42/ TLFN,18/ TLNT
*T, W2      24/ TLKS,18/0,18/ TLNO
*T, W3      24/ TLRS,18/0,18/ TLNF
*T, W4      1/A,1/B,1/C,1/D,1/E,1/F,6/ TLMD,30/,18/ TLNL
*T, W5      1/G,23/,18/0,18/ TLNK
*T, W6      60/ TLOP
*T, W7      60/ TLNP
*T, W8      60/ TLRL
*T, W9      60/ TLWL
*T, W10     42/ TLPN,12/ TLDV,6/ TLUN
*T, W11     30/ TLKL,12/,18/ TLNA
*T, W12     60/ TLKW(0)
*T, W12+N   60/ TLKW(N)
*
*                   TLFN - LOGICAL FILE NAME.
*                   TLNT - FWA OF NEXT LOGICAL NAME TABLE ENTRY.
*                   TLKS - KEY SIZE IN CHARACTERS.
*                   TLNO - FWA OF NEXT OPEN FILE LINK.
*                   TLRS - RECORD SIZE IN CHARACTERS.
*                   TLNF - FWA OF NEXT FREE FILE LINK.
*              A =  TLFD - FILE DOWN IF *TLFD* .EQ. 1.
*              B =  TLFI - FILE IDLE IF *TLFI* .EQ. 1.
*              C =  TLFL - FILE LOCKED IF *TLFL* .EQ. 1.
*              D =  TLRF - FILE RECOVERABLE IF *TLRF* .EQ. 1.
*              E =  TLBR - FILE DOWN FOR BATCH RECOVERY IF *TLBR* .EQ. 1.
*              F =  TLFE - FILE IDLE/DOWN FOR FATAL *CRM* ERROR IF .EQ. 1.
*                   TLMD - FILE ATTACH MODE.
*                   TLNL - FWA OF NEXT USED LOCK LINK.
*              G =  TLIC - FILE INCONSISTENT.
*                   TLNK - FWA OF NEXT FREE LOCK LINK.
*                   TLOP - TOTAL NUMBER OF OPENS ON FILE.
*                   TLNP - NUMBER OF OPEN REJECTS BECAUSE OF NO AVAILABLE
*                          FILE CONTROL ENTRIES.
*                   TLRL - TOTAL NUMBER OF LOCKS ON FILE.
*                   TLWL - NUMBER OF TIMES LOCK COULD NOT BE GRANTED
*                          BECAUSE OF NO AVAILABLE LOCK ENTRIES.
*                   TLPN - PACKNAME OF DEVICE WHERE FILE RESIDES.
*                   TLDV - DEVICE TYPE ON WHICH FILE RESIDES.
*                   TLUN - DEVICE UNIT NUMBER (OCTAL).
*                   TLKL - PRIMARY KEY LENGTH.
*                   TLNA - NUMBER OF ALTERNATE KEYS.
*                         (ONLY IF TLNA .NE. 0)
*                   TLKW(0) - PRIMARY KEY DESCRIPTOR.
*                   TLKW(N) - KEY DESCRIPTOR FOR ALTERNATE KEY N.
*
*                   KEY DESCRIPTOR FORMAT.
*
*T                  6/0,18/ KA,18/ BKP,18/ KL
*
*                   KA    - RELATIVE KEY ADDRESS.
*                   BKP   - BEGINNING KEY POSITION.
*                   KL    - KEY LENGTH.
```

LOGICAL NAME TABLE - TLNT
(CONTINUED)

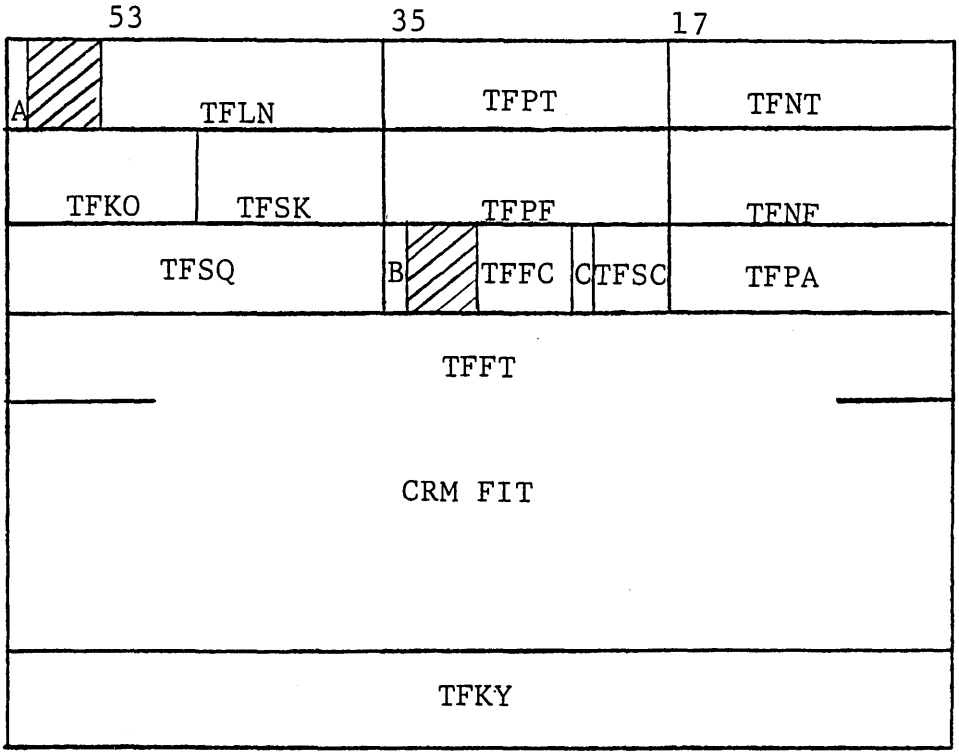
```
TLFN     FIELD  0,59,18     LOGICAL FILE NAME
TLNT     FIELD  0,17,0      FWA OF NEXT *TLNT* ENTRY
TLKS     FIELD  1,59,36     KEY SIZE IN CHARACTERS
TLNO     FIELD  1,17,0      FWA OF NEXT OPEN FILE LINK
TLRS     FIELD  2,59,36     RECORD SIZE IN CHARACTERS
TLNF     FIELD  2,17,0      FWA OF NEXT FREE FILE CONTROL LINK
TLFD     FIELD  3,59,59     FILE DOWN IF *TLFD* .EQ. 1
TLFI     FIELD  3,58,58     FILE IDLE IF *TLFI* .EQ. 1
TLFL     FIELD  3,57,57     FILE LOCKED IF *TLFL* .EQ. 1
TLRF     FIELD  3,56,56     FILE RECOVERABLE IF *TLRF* .EQ. 1
TLBR     FIELD  3,55,55     FILE DOWN FOR BAT. REC. IF *TLBR* .EQ. 1
TLFE     FIELD  3,54,54     FILE IDLE/DOWN FOR FATAL *CRM* ERROR
TLMD     FIELD  3,53,48     FILE ATTACH MODE
TLNL     FIELD  3,17,0      FWA OF NEXT USED LOCK LINK
TLIC     FIELD  4,59,59     FILE INCONSISTENT
TLNK     FIELD  4,17,0      FWA OF NEXT FREE LOCK LINK
TLOP     FIELD  5,59,0      TOTAL NUMBER OF OPENS FOR FILE
TLNP     FIELD  6,59,0      NUMBER OF OPEN REJECTS
TLRL     FIELD  7,59,0      TOTAL NUMBER OF LOCKS FOR FILE
TLWL     FIELD  8,59,0      NUMBER OF LOCK REJECTS FOR FILE
TLPN     FIELD  9,59,18     PACKNAME OF DEVICE WHERE FILE RESIDES
TLDV     FIELD  9,17,6      DEVICE TYPE ON WHICH FILE RESIDES
TLUN     FIELD  9,5,0       DEVICE UNIT NUMBER (OCTAL)
TLKL     FIELD  10,59,30    LENGTH OF PRIMARY KEY IN CHARACTERS
TLNA     FIELD  10,17,0     NUMBER OF ALTERNATE KEYS
TLKW     FIELD  11,59,0     KEY DESCRIPTOR

TLNTE    EQU    12          LENGTH OF LOGICAL NAME ENTRY
```

FILE CONTROL TABLE - TFCB

THIS TABLE HOLDS THE CURRENT TAF/CRM REQUEST AND THE FIT FOR THE DATA BASE FILE. THE NUMBER OF ENTRIES IN THIS TABLE FOR ONE DATA BASE FILE IS EQUAL TO THE 'NUMBER OF USERS' SPECIFIED ON THE CRM CARD IN THE XXJ FILE. THERE IS ONE ENTRY RESERVED FOR EACH TRANSACTION ACTIVELY USING THE FILE.

THE ADDRESS OF THE FIRST TFCB FOR A PARTICULAR FILE CAN BE FOUND IN THE TLNT FOR THAT FILE, FIELD TLNF. ALSO, IN THE TLNT, FIELD TLNO CONTAINS THE ADDRESS OF THE FIRST TFCB RESERVED FOR THE FILE.

File Control Table - TFCB



FILE CONTROL TABLE - TFCB

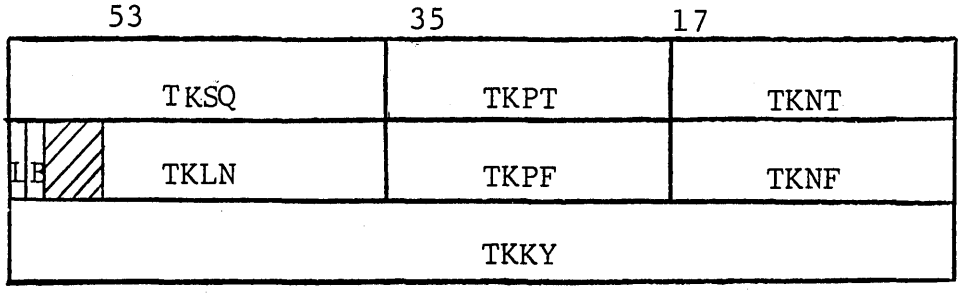
```
**        TFCB - FILE CONTROL TABLE.
*
*T  W1    1/A,5/,18/ TFLN,18/ TFPT,18/ TFNT
*T, W2    12/ TFKO,12/ TFSK,18/ TFPF,18/ TFNF
*T, W3    24/ TFSQ,1/B,5/,6/ TFFC,1/C,5/ TFSC,18/ TFPA
*T, W4    60/ TFFT
*T, W39   60/ TFKY
*
*           A = TFFI - *FIT* *FWI* CHANGED BY *DLX* IF *TFFI* .EQ. 1.
*               TFLN - FWA OF LOGICAL NAME TABLE.
*               TFPT - FWA OF LAST FILE CONTROL LINK FOR
*                      TRANSACTION.
*               TFNT - FWA OF NEXT FILE CONTROL LINK FOR
*                      TRANSACTION.
*               TFKO - KEY ORDINAL FOR CURRENT KEY.
*               TFSK - CURRENT SEEK COUNT.
*               TFPF - FWA OF LAST FILE CONTROL LINK FOR FILE.
*               TFNF - FWA OF NEXT FILE CONTROL LINK FOR FILE.
*           B = TFBF - INTERNAL DBFREE PROCESSING IF *TFBF* .EQ. 1.
*               TFSQ - TRANSACTION SEQUENCE NUMBER.
*               TFFC - FUNCTION CODE OF REQUEST.
*           C = TFRC - RECALL SELECTED IF *TFRC* .EQ. 1.
*               TFSC - SUB-CONTROL POINT NUMBER OF REQUEST.
*               TFPA - ADDRESS OF REQUEST PARAMETERS.
*               TFFT - FILE INFORMATION TABLE, *FIT*.
*                      WORDS 4 - 38 CONTAIN THE *FIT*.
*               TFKY - KEY OF DESIRED RECORD.  THE LENGTH OF THIS
*                      AREA DEPENDS ON KEY SIZE OF FILE.


 TFFI     FIELD  0,59,59     *FIT* *FWI* CHANGED BY *DLX* IF .EQ. 1
 TFLN     FIELD  0,53,36     FWA OF LOGICAL NAME ENTRY
 TFPT     FIELD  0,35,18     FWA OF LAST *TFCB* LINK FOR TRANSACTION
 TFNT     FIELD  0,17,0      FWA OF NEXT *TFCB* LINK FOR TRANSACTION
 TFKO     FIELD  1,59,48     CURRENT ALTERNATE KEY ORDINAL
 TFSK     FIELD  1,47,36     CURRENT SEEK COUNT
 TFPF     FIELD  1,35,18     FWA OF LAST FILE CONTROL LINK
 TFNF     FIELD  1,17,0      FWA OF NEXT FILE CONTROL LINK
 TFRQ     FIELD  2,59,0      *TAF CRM* REQUEST
 TFSQ     FIELD  2,59,36     TRANSACTION SEQUENCE NUMBER
 TFBF     FIELD  2,35,35     INTERNAL DBFREE IF *TFBF* .EQ. 1
 TFFC     FIELD  2,29,24     FUNCTION CODE
 TFRC     FIELD  2,23,23     IF *TFRC* .EQ. 1, USE RECALL
 TFSC     FIELD  2,22,18     SUB-CONTROL POINT NUMBER
 TFPA     FIELD  2,17,0      FWA OF REQUEST PARAMETERS
 TFFT     FIELD  3,59,0      FIRST WORD OF *FIT*
 TFKY     FIELD  38,59,0     FIRST WORD OF KEY
```

LOCK TABLE - TKOK

THIS TABLE CONTAINS THE CONTROLLING INFORMATION THROUGH WHICH FILE AND RECORD LOCKS ON DATA BASE FILES ARE GRANTED AND RELEASED. THE NUMBER OF ENTRIES IN THIS TABLE FOR ONE DATA BASE FILE IS EQUAL TO THE 'NUMBER OF LOCKS' SPECIFIED ON THE CRM CARD IN THE XXJ FILE. THERE IS ONE ENTRY RESERVED FOR EACH LOCK HELD BY THE TRANSACTION.

THE ADDRESS OF THE FIRST TKOK FOR A PARTICULAR FILE CAN BE FOUND IN THE TLNT FOR THAT FILE, FIELD TLNK. ALSO, IN THE TLNT, FIELD TLNL CONTAINS THE ADDRESS OF THE FIRST TKOK RESERVED FOR THE FILE.

Lock Table - TKOK



LOCK TABLE - TKOK

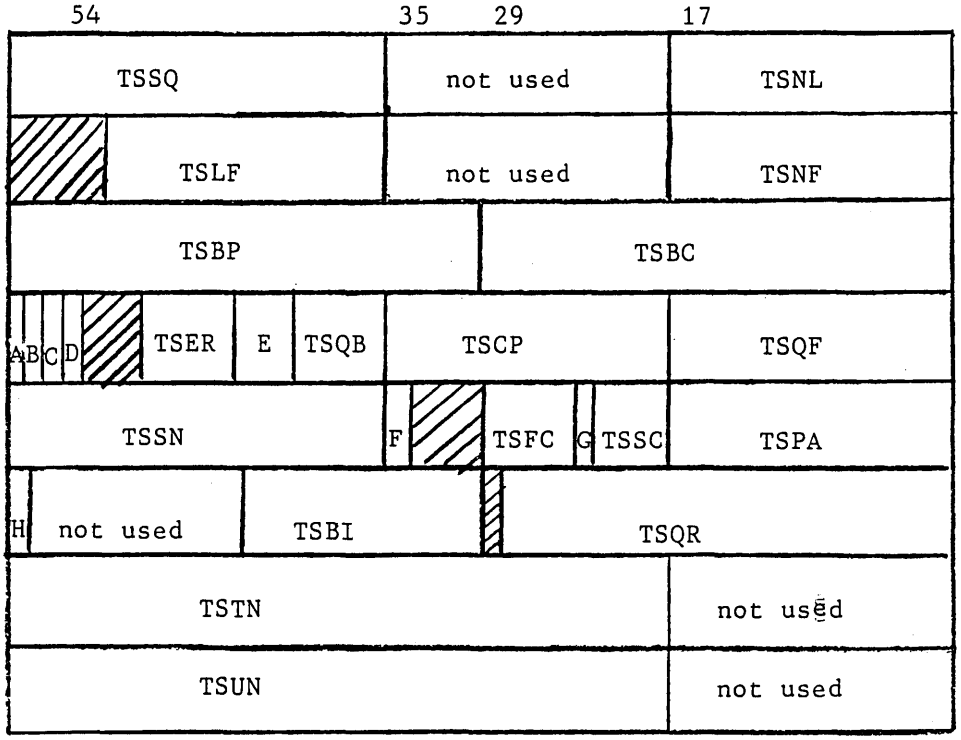
```
**          TKOK - LOCK TABLE.
*
*T  W1      24/ TKSQ,18/ TKPT,18/ TKNT
*T, W2      1/L,1/B,4/,18/ TKLN,18/ TKPF,18/ TKNF
*T, W3      60/ TKKY
*                   TKSQ - TRANSACTION SEQUENCE NUMBER RESERVING ENTRY.
*                   TKPT - FWA OF LAST LOCK LINK FOR TRANSACTION.
*                   TKNT - FWA OF NEXT LOCK LINK FOR TRANSACTION.
*                      L - IF FILE IS LOCKED, *L* .EQ. 1.
*              B = TKQR - BEFORE IMAGE WRITTEN TO DISK IF
*                         *TKQR* .EQ. 1 .
*                   TKLN - FWA OF LOGICAL NAME ENTRY.
*                   TKPF - FWA OF LAST LOCK LINK FOR FILE.
*                   TKNF - FWA OF NEXT LOCK LINK FOR FILE.
*                   TKKY - KEY OF LOCKED RECORD.  THE LENGTH OF THIS FIELD
*                          DEPENDS ON KEY SIZE OF FILE.

 TKSQ     FIELD  0,59,36     TRANSACTION SEQUENCE FOR LOCK
 TKPT     FIELD  0,35,18     FWA OF LAST TRANSACTION LOCK LINK
 TKNT     FIELD  0,17,0      FWA OF NEXT TRANSACTION LOCK LINK
 TKFK     FIELD  1,59,59     IF FILE LOCK, *TKFK* .EQ. 1
 TKQR     FIELD  1,58,58     *BI* WRITTEN IF *TKQR* .EQ. 1
 TKLN     FIELD  1,53,36     FWA OF LOGICAL NAME TABLE ENTRY
 TKPF     FIELD  1,35,18     FWA OF LAST LOCK LINK FOR FILE
 TKNF     FIELD  1,17,0      FWA OF NEXT LOCK LINK FOR FILE
 TKKY     FIELD  2,59,0      KEY OF LOCKED RECORD
```

TRANSACTION SEQUENCE TABLE - TSEQ

THIS TABLE TRACKS AND CONTROLS ACCESS TO THE TAF/CRM DATA MANAGER. THROUGH THIS TABLE RESOURCE INFORMATION NECESSARY FOR A TRANSACTION'S USE OF TAF/CRM IS KEPT TRACK OF. THE LENGTH OF THE TSEQ IS ESTABLISHED BY INTALLATION PARAMETERS CMDM AND RMDM.

THE TSEQ IMMEDIATELY FOLLOWS THE TAF/CRM INPUT AND OUTPUT QUEUES. IT IS REFERENCED IN LOW CORE OF AAMI AT TAG TSEQ.

Transaction Sequence Table - TSEQ



TRANSACTION SEQUENCE TABLE - TSEQ

```
**          TSEQ - TRANSACTION SEQUENCE TABLE.
*
*T  W1      24/ TSSQ,18/0,18/ TSNL
*T, W2      6/,18/ TSLF,18/0,18/ TSNF
*T, W3      30/ TSBP,30/ TSBC
*T, W4      1/A,1/B,1/C,1/D,5/,6/ TSER,3/ E,6/ TSQB,18/ TSCP,18/ TSQF
*T, W5      24/ TSSN,1/F,5/,6/ TSFC,1/G,5/ TSSC,18/ TSPA
*T, W6      1/H,20/,9/ TSBI,1/,29/ TSQR
*T, W7      42/ TSTN,18/
*T, W8      42/ TSUN,18/
*
*                   TSSQ - TRANSACTION SEQUENCE NUMBER.
*                   TSNL - FWA OF NEXT LOCK LINK FOR TRANSACTION.
*                   TSLF - FWA OF AFTER IMAGE RECOVERY FILE TABLE
*                          ASSIGNED TASK.
*                   TSNF - FWA OF NEXT FILE CONTROL LINK
*                          FOR TRANSACTION.
*                   TSBP - BEGIN ID OF PREVIOUS BEGIN/COMMIT SEQUENCE.
*                   TSBC - BEGIN ID OF CURRENT BEGIN/COMMIT SEQUENCE.
*              A = TSBR - BEGIN PROCESSED FOR TASK IF *TSBR* .EQ. 1.
*              B = TSBW - BEGIN IMAGE WRITE PENDING IF *TSBW* .EQ. 1.
*              C = TSRC - TASK RECOVERED FROM *BRF* IF *TSRC* .EQ. 1.
*              D = TSAI - AFTER IMAGE WRITE PENDING IF *TSAI* .EQ. 1.
*                  TSER - ERROR CODE WHICH CAUSED FREEING TO OCCUR.
*              E = TSQW - ASSIGNED *BRF* SEGMENT BIT MAP WORD NUMBER.
*                  TSQB - ASSIGNED *BRF* SEGMENT BIT MAP WORD BIT NUMBER.
*                  TSCP - DATA BASE LEVEL REQUEST CONTINUATION PROCESSOR.
*                  TSQF - FWA OF BEFORE IMAGE RECOVERY FILE TABLE
*                         ASSIGNED TASK.
*                  TSRQ - INITIAL *TAF CRM* REQUEST IF IMPLICIT
*                         COMMIT/FREE PROCESSING.
*                  TSSN - TRANSACTION SEQUENCE NUMBER.
*              F = TSBF - INTERNAL *DBFREE* IF *TSBF* .EQ. 1.
*                  TSFC - FUNCTION CODE.
*              G = TSRR - USE RECALL IF *TSRR* .EQ. 1.
*                  TSSC - SUB-CONTROL POINT NUMBER.
*                  TSPA - FWA OF REQUEST PARAMETERS.
*              H = TSRF - SECOND ATTEMPT TO ROLLBACK UPDATE IF .EQ. 1.
*                  TSBI - NUMBER OF BEFORE IMAGES ON *BRF*.
*                  TSQR - MS RANDOM ADDR. TO WRITE NEXT *BI* RECORD.
*                  TSTN - TASK NAME.
*                  TSUN - USER NAME.
```

TRANSACTION SEQUENCE TABLE - TSEQ
(CONTINUED)

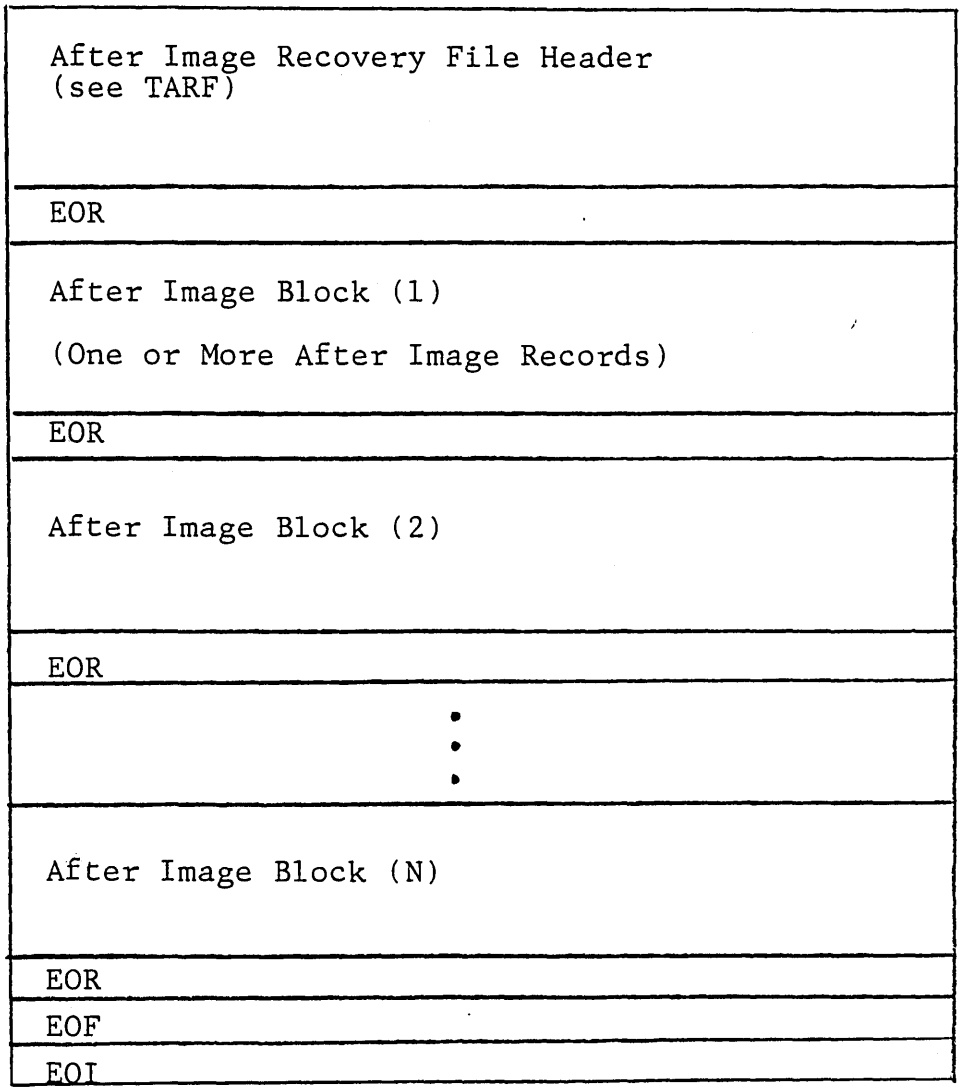
```
TSSQ     FIELD  0,59,36     TRANSACTION SEQUENCE NUMBER
TSNL     FIELD  0,17,0      FWA OF NEXT LOCK LINK
TSLF     FIELD  1,53,36     FWA OF ASSIGNED *ARF* TABLE
TSNF     FIELD  1,17,0      FWA OF NEXT FILE CONTROL LINK
TSBP     FIELD  2,59,30     PREVIOUS BEGIN ID
TSBC     FIELD  2,29,0      CURRENT BEGIN ID
TSBR     FIELD  3,59,59     IF *TSBR* .EQ. 1 BEGIN WAS PROCESSED
TSBW     FIELD  3,58,58     IF *TSBW* .EQ. 1 BEGIN IMAGE WRITE PENDING
TSRC     FIELD  3,57,57     TASK RECOVERED FROM *BRF* IF *TSRC* .EQ. 1
TSAI     FIELD  3,56,56     AFTER IMAGE WRITE PENDING FLAG
TSER     FIELD  3,50,45     ERROR CODE WHICH CAUSED FREEING TO OCCUR
TSQW     FIELD  3,44,42     *BRF* SEGMENT BIT MAP WORD NUMBER
TSQB     FIELD  3,41,36     *BRF* SEGMENT BIT MAP WORD BIT NUMBER
TSMP     FIELD  3,44,36     BIT MAP BIT POINTER (*TSQW* + *TSQB*)
TSCP     FIELD  3,35,18     DB REQUEST CONTINUATION PROCESSOR ADDR.
TSQF     FIELD  3,17,0      FWA OF ASSIGNED *BRF* TABLE
TSRQ     FIELD  4,59,0      INITIAL *TAF CRM* REQUEST
TSSN     FIELD  4,59,36     TRANSACTION SEQUENCE NUMBER (DBLR)
TSBF     FIELD  4,35,35     INTERNAL *DBFREE* IF *TSBF* .EQ. 1 (DBLR)
TSFC     FIELD  4,29,24     FUNCTION CODE (DBLR)
TSRR     FIELD  4,23,23     USE RECALL IF *TSRR* .EQ. 1 (DBLR)
TSSC     FIELD  4,22,18     SUB-CONTROL POINT NUMBER (DBLR)
TSPA     FIELD  4,17,0      FWA OF PARAMETERS (DBLR)
TSRF     FIELD  5,59,59     SECOND ATTEMPT TO ROLLBACK UPDATE FLAG
TSBI     FIELD  5,38,30     NUMBER OF BEFORE IMAGES ON *BRF* SEGMENT
TSQR     FIELD  5,28,0      MS RANDOM ADDR. TO WRITE NEXT *BI* RECORD
TSTN     FIELD  6,59,18     TASK NAME
TSUN     FIELD  7,59,18     USER NAME

TSEQE    EQU    8           LENGTH OF TRANSACTION SEQUENCE ENTRY
```

AFTER IMAGE RECOVERY FILE

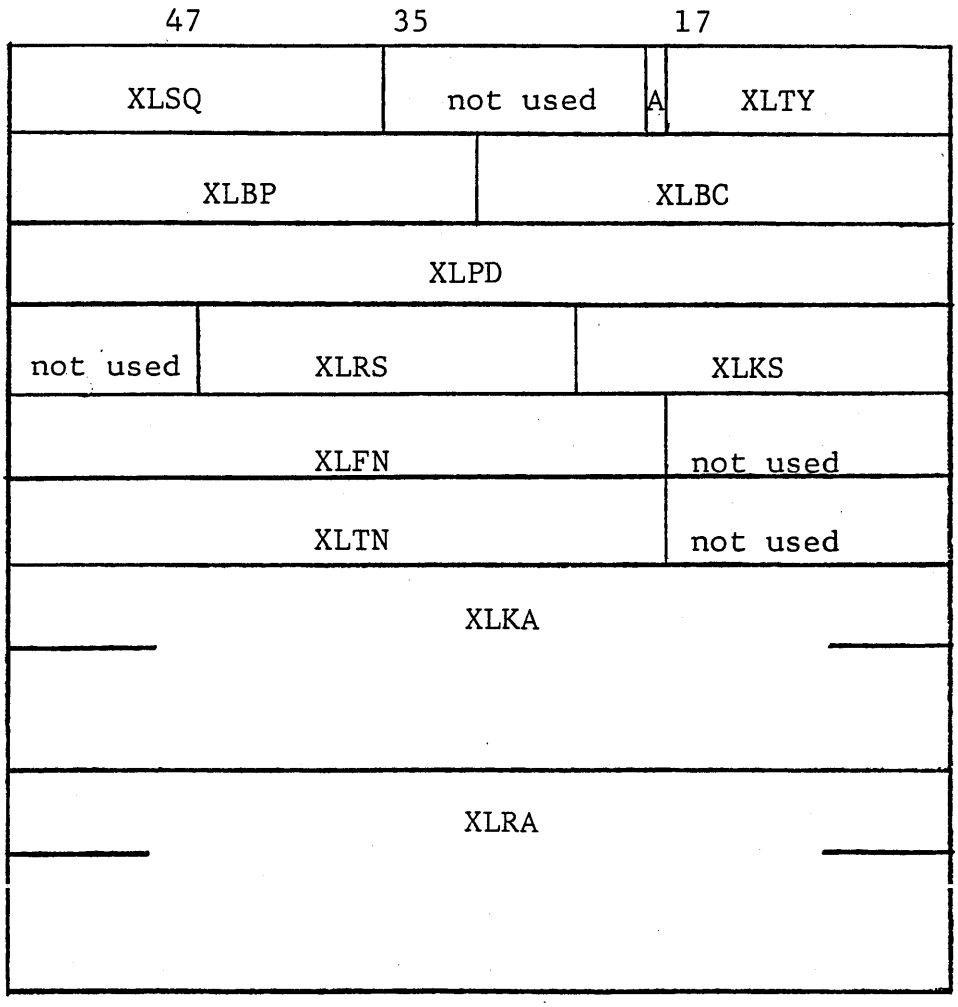
THE AFTER IMAGE RECOVERY FILE (ARF) CONTAINS IMAGES OF THE RECORDS THAT HAVE BEEN CHANGED BY A TRANSACTION UPDATING RECOVERABLE FILES. THE ARF CONTAINS AFTER IMAGE BLOCKS SEPARATED BY END OF RECORD MARKS. EACH BLOCK MAY CONTAIN ONE OR MORE AFTER IMAGE RECORDS. THE LAST BLOCK IS FOLLOWED BY AN END OF RECORD AND AN END OF FILE MARK. THE FIRST BLOCK ON THE ARF IS THE FILE HEADER. THE REMAINING BLOCKS CONTAIN AFTER IMAGE DATA.

THE AFTER IMAGE RECORDS ARE WRITTEN BY TAF/CRM AND READ AND DUMPED TO TAPE BY DMREC.

After Image Recovery File
File Structure



After Image Recovery File
Record Description



AFTER IMAGE RECOVERY FILE RECORD DESCRIPTION

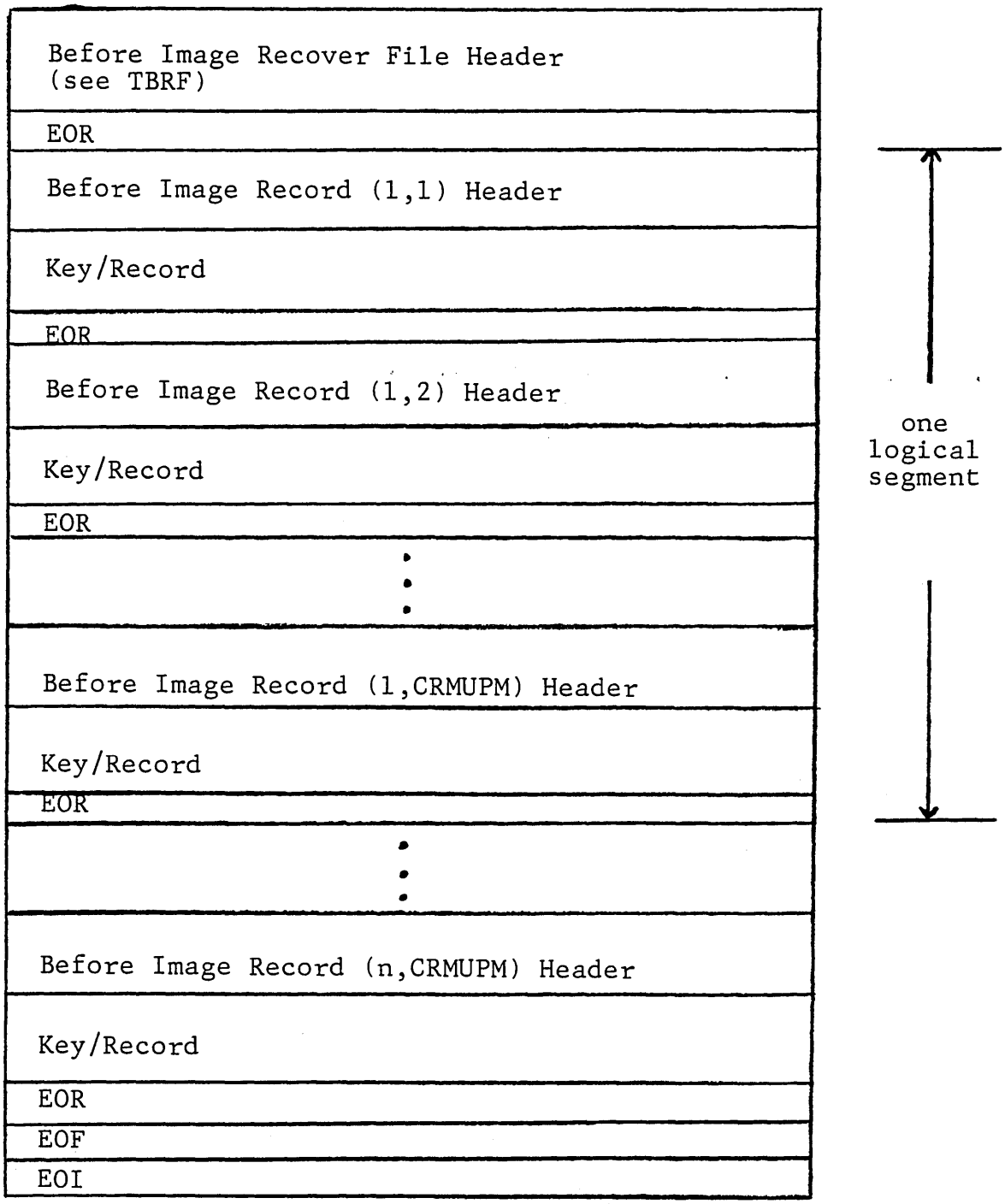
```
**        XARF - AFTER IMAGE RECOVERY FILE RECORD DESCRIPTION.
*
*T  W1    24/ XLSQ,17/,1/A,18/ XLTY
*T, W2    30/ XLBP,30/ XLBC
*T, W3    60/ XLPD
*T, W4    12/,24/ XLRS,24/ XLKS
*T, W5    42/ XLFN,18/
*T, W6    42/ XLTN,18/
*T, W7    N/XLKA  (N=((KS+9)/10 IN WORDS)
*T, W7+N  M/XLRA  (M=((RS+9)/10) IN WORDS)
*
* AI HEADER+0     XLSQ - TRANSACTION SEQUENCE NUMBER.
*              A = XLBW - BEGIN INDICATOR.
*                         1 IF FIRST AFTER IMAGE AFTER *DBEGIN* REQUEST.
*                         0 IF NOT FIRST AFTER IMAGE AFTER *DBEGIN* REQ.
*                 XLTY - AFTER IMAGE TYPE CODE.
*                         0    IF *DBCOMIT* REQUEST.
*                         *XLQD* IF *BRF* DOWN STAMP.
*                         *TRDE* IF DELETE REQUEST.
*                         *TRRW* IF REWRITE REQUEST.
*                         *TRWR* IF WRITE REQUEST.
*                         *TRDF* IF *DBFREE* REQUEST.
*                         *DMCC* IF DATA MANAGER *CEASE* REQUEST.
* AI HEADER+1     XLBP - PREVIOUS BEGIN IDENTIFIER.
*                 XLBC - CURRENT BEGIN IDENTIFIER.
* AI HEADER+2     XLPD - PACKED DATE/TIME AFTER IMAGE WRITTEN.
* AI HEADER+3     XLRS - RECORD AREA SIZE IN CHARACTERS.
*                 XLKS - KEY AREA SIZE IN CHARACTERS.
* AI HEADER+4     XLFN - LOGICAL FILE NAME OF MODIFIED FILE, OR
*                         NAME OF DOWN *BRF* IF *XLTY* .EQ. *XLQD*.
* AI HEADER+5     XLTN - TASK NAME.
*                 XLKA - FIRST WORD OF KEY-AREA.
*                 XLRA - FIRST WORD OF RECORD AREA.

 XLSQ     FIELD  0,59,36     TRANSACTION SEQUENCE NUMBER
 XLBW     FIELD  0,18,18     BEGIN INDICATOR
 XLTY     FIELD  0,17,0      AFTER IMAGE TYPE CODE
 XLBP     FIELD  1,59,30     PREVIOUS BEGIN IDENTIFIER
 XLBC     FIELD  1,29,0      CURRENT BEGIN IDENTIFIER
 XLPD     FIELD  2,59,0      PACKED DATE/TIME AFTER IMAGE WRITTEN
 XLRS     FIELD  3,47,24     RECORD SIZE IN CHARACTERS
 XLKS     FIELD  3,23,0      KEY SIZE IN CHARACTERS
 XLFN     FIELD  4,59,18     FILE NAME
 XLTN     FIELD  5,59,18     TASK NAME
 XLKA     FIELD  6,59,0      FIRST WORD OF KEY-AREA
```

BEFORE IMAGE RECOVERY FILE

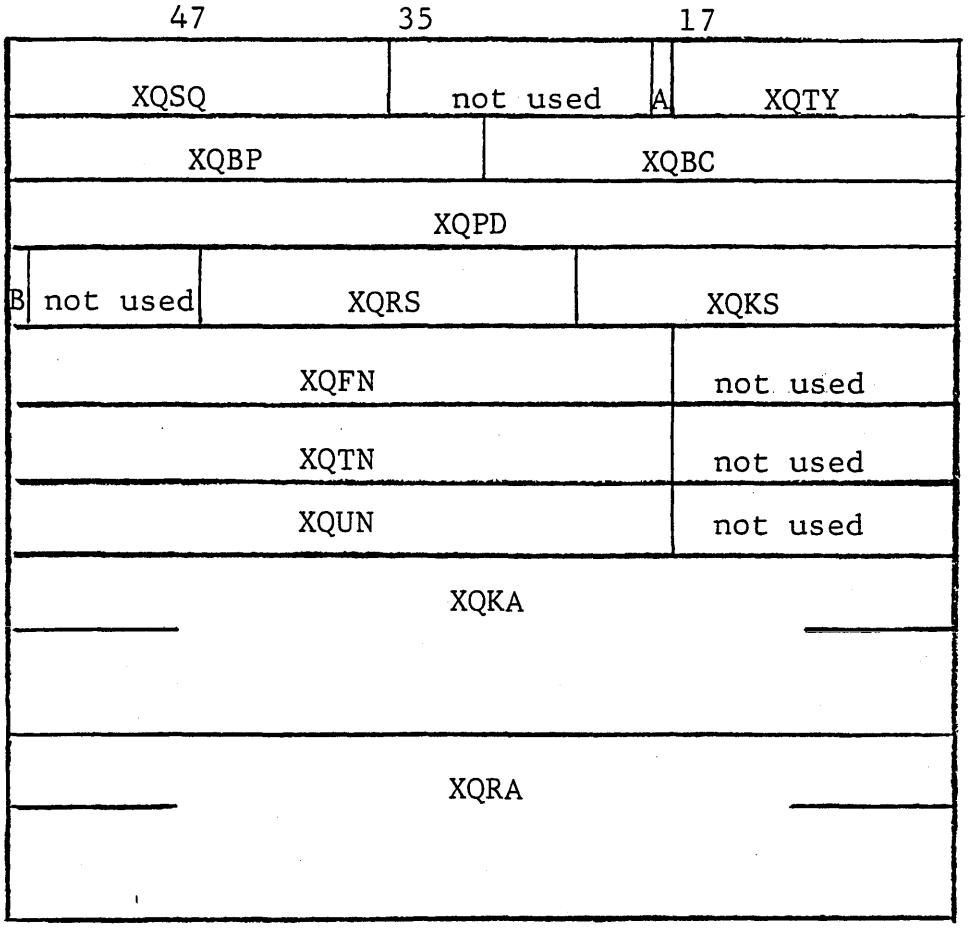
THE BEFORE IMAGE RECOVERY FILE (BRF) CONTAINS IMAGES OF THE RECORDS BEFORE THEY ARE CHANGED BY A TRANSACTION. BEFORE IMAGES ARE ONLY KEPT FOR RECOVERABLE FILES. THE BRF CONTAINS BEFORE IMAGE RECORDS SEPARATED BY END OF RECORD MARKS. THE FIRST RECORD ON THE BRF IS THE FILE HEADER. THE REMAINING RECORDS CONTAIN BEFORE IMAGE DATA. BEFORE IMAGE RECORDS ARE LOGICALLY GROUPED INTO (CMDM*RMDM) SEGMENTS. THE NUMBER OF BEFORE IMAG RECORDS IN EACH SEGMENT IS CONTROLLED BY THE INSTALLATION PARAMETER, CRMUPM.

THE BEFORE IMAGE RECORDS ARE WRITTEN/READ TO/FROM THE BRF BY TAF/CRM.

Before Image Recovery File
File Structure



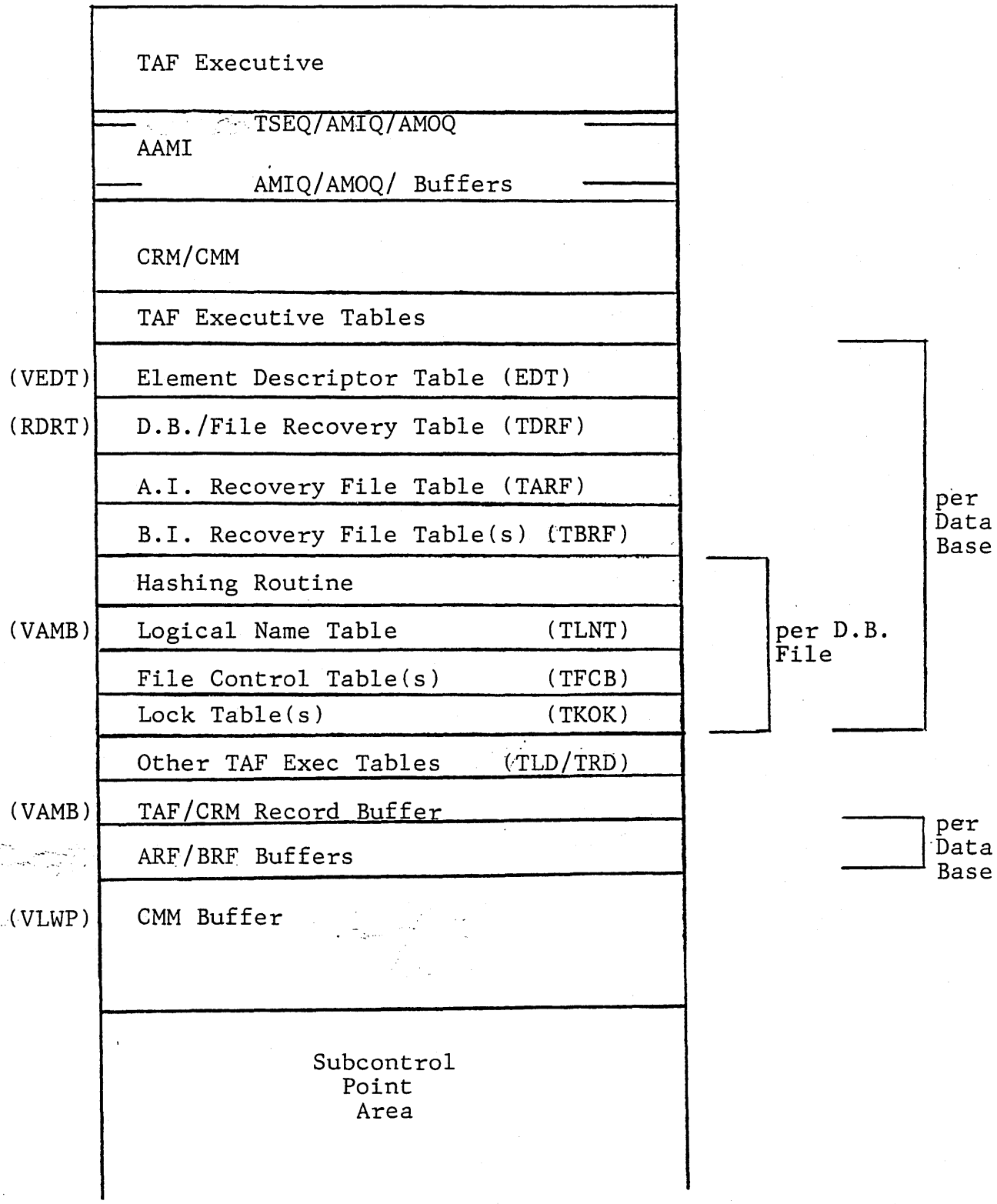
Before Image Recovery File
Record Description



BEFORE IMAGE RECOVERY FILE RECORD DESCRIPTION

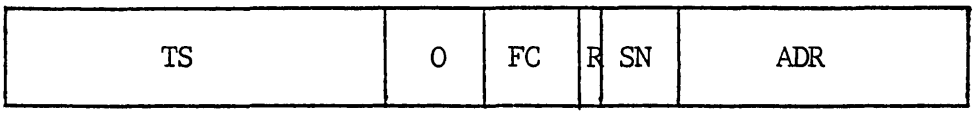
```
**        XBRF - BEFORE IMAGE RECOVERY FILE RECORD DESCRIPTION.
*
*T  W1     24/ XQSQ,17/,1/A,18/ XQTY
*T, W2     30/ XQBP,30/ XQBC
*T, W3     60/ XQPD
*T, W4     1/B,11/,24/ XQRS,24/ XQKS
*T, W5     42/ XQFN,18/
*T, W6     42/ XQTN,18/
*T, W7     42/ XQUN,18/
*T, W8     N/  XQKA (N=((KS+9)/10) IN WORDS)
*T, W8+N   M/  XQRA (M=((RS+9)/10) IN WORDS)
*
* BI HEADER+0     XQSQ - TRANSACTION SEQUENCE NUMBER.
*             A = XQBR - BEGIN INDICATOR.
*                        1 = *DBEGIN* OUTSTANDING.
*                        0 = *DBFREE* OR *DBCOMIT* REQUEST PROCESSED.
*                 XQTY - TYPE OF BEFORE IMAGE RECORD.
*                        ZERO   = *CEASE* STAMP.
*                        *TRDE* = DELETE REQUEST.
*                        *TRRW* = REWRITE REQUEST.
*                        *TRWR* = WRITE REQUEST.
*                        *TRDC* = DBCOMIT REQUEST.
*                        *TRDF* = DBFREE REQUEST.
* BI HEADER+1     XQBP - PREVIOUS BEGIN IDENTIFIER.
*                 XQBC - CURRENT BEGIN IDENTIFIER.
*                        PREVIOUS AND CURRENT BEGIN IDENTIFIERS ARE
*                        REQUIRED ON THE FIRST BEFORE IMAGE IN A *BRF*
*                        SEGMENT.
* BI HEADER+2     XQPD - PACKED DATE/TIME WHEN THIS *BI* WAS WRITTEN.
* BI HEADER+3 B=  XQFL - FILE/RECORD LOCK INDICATOR.
*                        1 = IF FILE LOCKED BY TRANSACTION.
*                        0 = IF RECORD LOCKED BY TRANSACTION.
*                 XQRS - RECORD SIZE IN CHARACTERS.
*                        THIS IS NOT THE SIZE OF THIS *BI* RECORD.
*                 XQKS - KEY SIZE IN CHARACTERS.
* BI HEADER+4     XQFN - FILE NAME OF MODIFIED FILE.
* BI HEADER+5     XQTN - TASK NAME.
* BI HEADER+6     XQUN - USER NAME.
*                 XQKA - KEY AREA (CONTAINS USER RECORD KEY).
*                 XQRA - RECORD AREA (CONTAINS USER DATA RECORD).
*                        THE SIZE OF THIS RECORD IS IN *XQRS* FIELD.

 XQSQ      FIELD  0,59,36     TRANSACTION SEQUENCE NUMBER
 XQBR      FIELD  0,18,18     BEGIN INDICATOR
 XQTY      FIELD  0,17,0      BEFORE IMAGE TYPE CODE
 XQBP      FIELD  1,59,30     PREVIOUS BEGIN IDENTIFIER
 XQBC      FIELD  1,29,0      CURRENT BEGIN IDENTIFIER
 XQPD      FIELD  2,59,0      PACKED DATE/TIME BEFORE IMAGE WRITTEN
 XQFL      FIELD  3,59,59     FILE LOCK INDICATOR
 XQRS      FIELD  3,47,24     RECORD SIZE IN CHARACTERS
 XQKS      FIELD  3,23,0      KEY SIZE IN CHARACTERS
 XQFN      FIELD  4,59,18     FILE NAME
 XQTN      FIELD  5,59,18     TASK NAME
 XQUN      FIELD  6,59,18     USER NAME
 XQKA      FIELD  7,59,0      FIRST WORD OF KEY AREA
```

TAF/CRM Memory Map



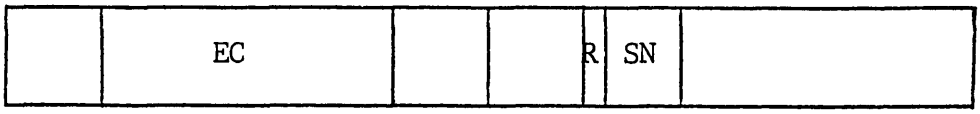
TAF/CRM INPUT/OUTPUT QUEUE ENTRIES

AMIQ



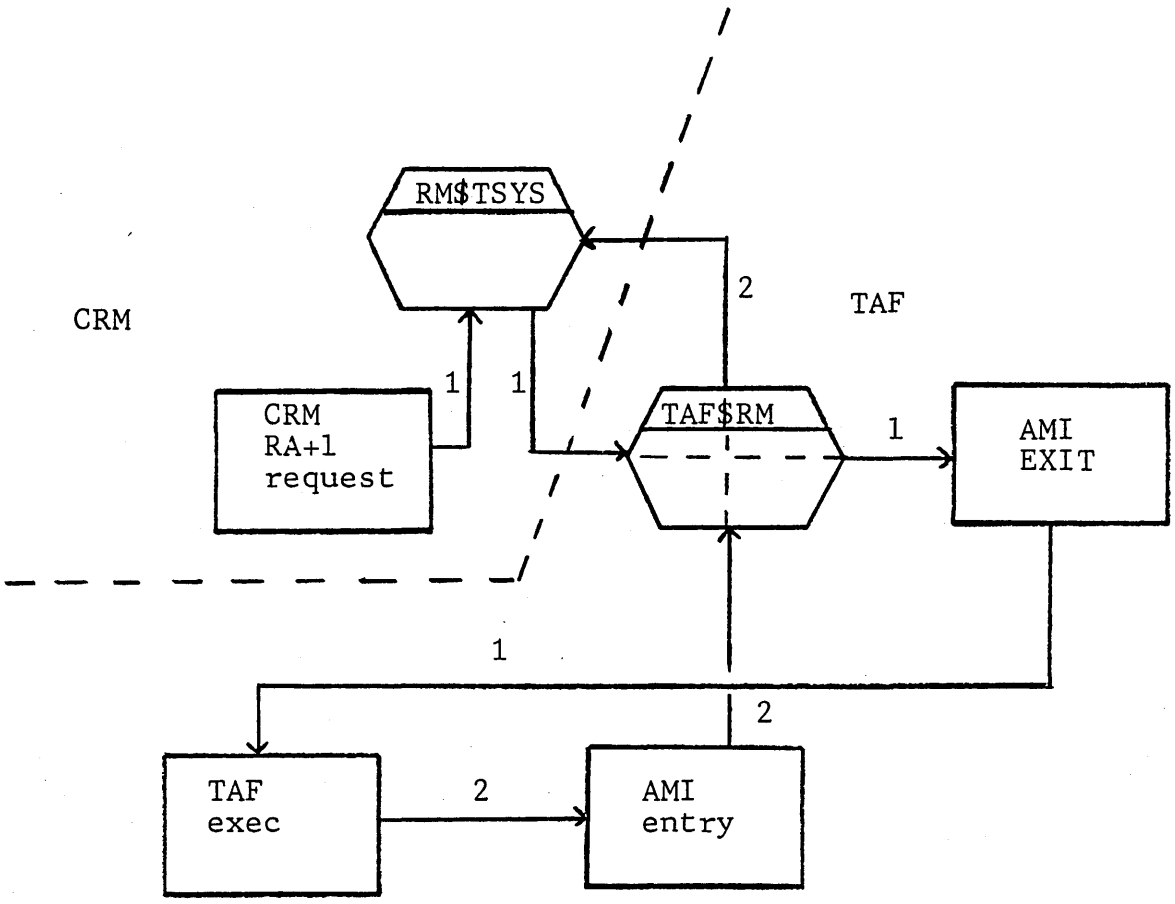
TS - Transaction Sequence Number
FC - Request Function Code
R - Recall Flag (always set)
SN - Sub-control Point Number
ADR - FWA of Request Parameters

AMOQ



EC - Error Code
R - Recall Flag
SN - Sub-control Point Number

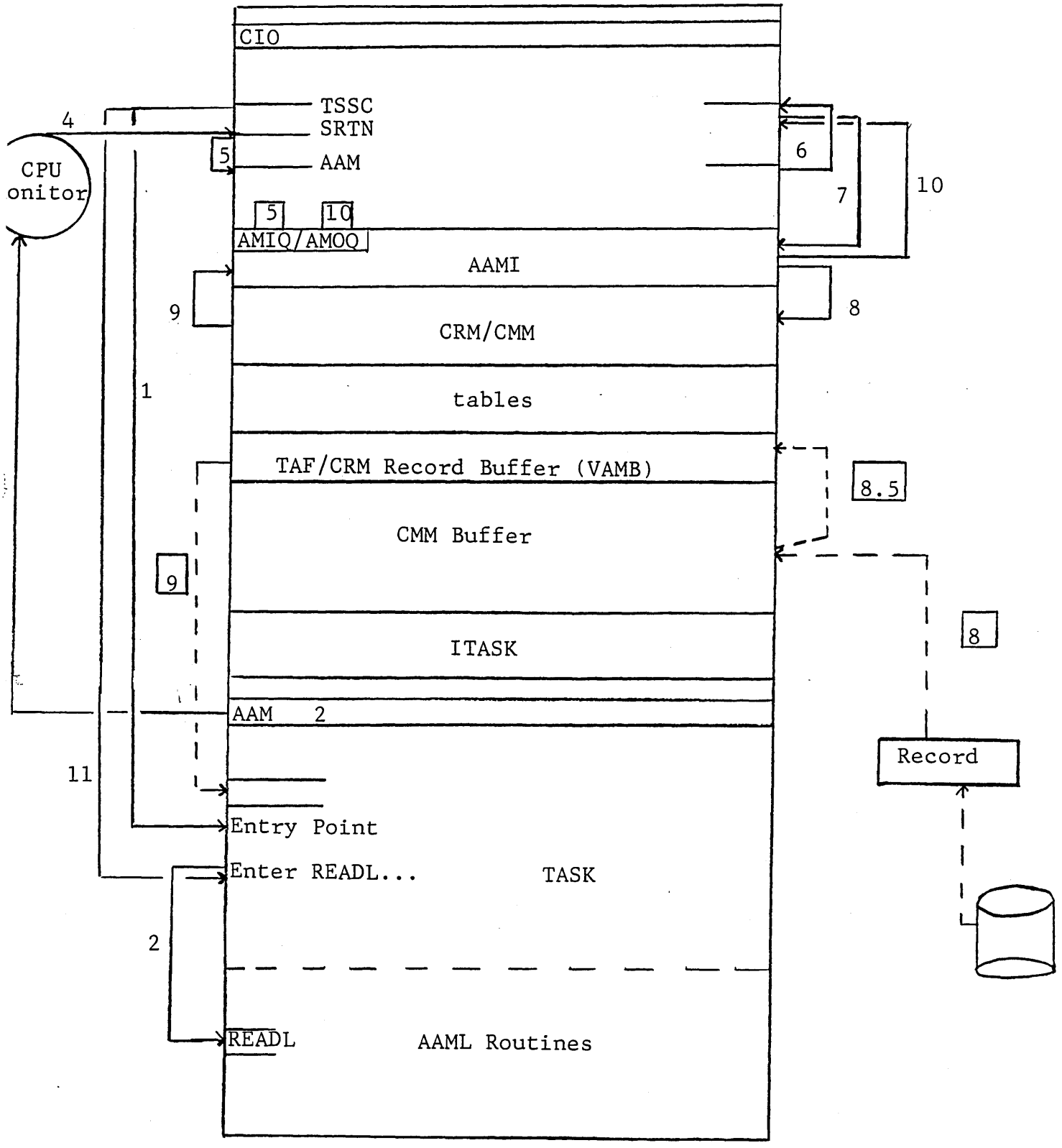
CRM RA+1 Processing



- RM$TSYS is substituted for normal CRM RA request processer. RM$SYS= at AAMI load time.

- TAF gets control during CRM requests with recall so TAF may do other work while request is being processed.

TAF/CRM Processing



FILE OPEN REQUEST

- Initial Open Process (FOP)

  1. Format free TFCB and link TFCB
  2. Validate file against CRM card (IOP)
  3. Issue OPENM, upon error go to 7
  4. Update TLNT with key descriptors and rewind file
  5. Open all FIT's
  6. Go to 11
  7. FIT not open or FNF set, return error code and go to 11
  8. Not CRM error or not recoverable, close file and go to 11
  9. File inconsistent, idle/close file and go to 11
  10. Non-fatal eror, clear FSNCL in FSTT and proceed
  11. Complete request (CRQ)

- Subsequent Open Requests (FOP)

  1. Format free TFCB and link TFCB
  2. Restore primary key description in FIT
  3. REWIND file
  4. Complete request (CRQ)

FILE READ REQUESTS
READ/READL/READN/READNL/READM

- READ/READL/READM
  1. Read begin routines, RDB/RLB/RMB
  2. Set key ordinal and extract key from task
  3. Lock record for primary key (READL only)
     If error, prepare for "Freeing" ⟶ CRQ
  4. Issue SEEK
  5. Read complete routines, RDC/RLC/RMC
  6. Issue START for READM request
  7. Get (GETN - READN) Record
  8. Lock record for alternate key (READL)
     If error, prepare for "Freeing" ⟶ CRQ
  9. Move Key/Record to task
  10. Return lock status (READ/READM)
  11. Complete request (CRQ)

- READN/READNL (RNB)
  1. Zero seek count
  2. RNC/ROC
  3. FETCH,FP/GETNR/FETCH,FP
  4. Lock record (READNL)
  5. Move key/record to task
  6. Complete request (CRQ)

RECOVERY REQUESTS
DBEGIN/DBCOMIT/DBFREE/DBSTAT

- DBEGIN (DBP)
  - Assigns and initializes TBRF/TARF
- DBCOMIT (DBC)
  1. Release all locks, if no BI's
  2. Flush all recoverable files
  3. Write commit stamp on ARF/flush ARF buffer
  4. Write commit stamp on BRF
  5. Release all locks ⟶ CRQ

- DBFREE (DBF/FRE)
  1. Release all locks if no BI's
  2. Read BRF/apply BI's to file/ARF
  3. Flush D.B. file
  4. Write free stamp on BRF/ARF
  5. If D.M. cease or TRMREC, release BRF assignment, all record/file locks, all files, and clear TSEQ entry
  6. Complete request ⟶ CRQ
- DBSTAT (DBS)
  - Gets current/previous id from TSEQ

FILE UPDATE REQUESTS
WRITE/REWRITE/DELETE

- Routines WRB/WRC/WDC
  1. Extract key from task/lock key
  2. Issue SEEK
  3. To WRC (WDC) when seek count exhausted
  4. Move record from task to buffer (WRD), write/rewrite only
  5. GET BI record (Rewrite/Delete)
  6. Move BI record (empty record or write) to BRF buffer
  7. PUT/REPLACE/DELETE record
  8. Write BI record to BRF (REWRITER)
  9. Write AI record to ARF Buffer
  10. Lock record and move key to task if AK file
  11. Complete request ⟶ CRQ

- ARF Buffer Flushed When:
  1. - full
  2. - DBFREE/DBCOMIT
  3. - BRF Down
  4. - Data Base Down

FILE CLOSE REQUEST

- Routine FCL

- Steps
  1. DBEGIN outstanding, return with error
  2. Release all locks for this transaction on this file.
  3. Delete file (TFCB) from transaction/open chain.
  4. Insert file (TFCB) into free chain
  5. Complete request ⟶ CRQ

CRM DEFERRED LOGGING EXIT

- For all recoverable files

- To ensure that the BI write completes before the record is physically updated on the data base file.

- Routine DLX
  1. Return to CRM if MIP Block or first call.
  2. Set FWI in FIT if multiple blocks involved.
  3. Return to CRM if FSTT block
  4. Log BI and initiate BRF I/O, if BI write is pending.
  5. Exit to TAF executive if I/O not complete.

  Subsequent entry to AMI will cause check on I/O completion. If complete, return to CRM. If not, back to TAF executive.

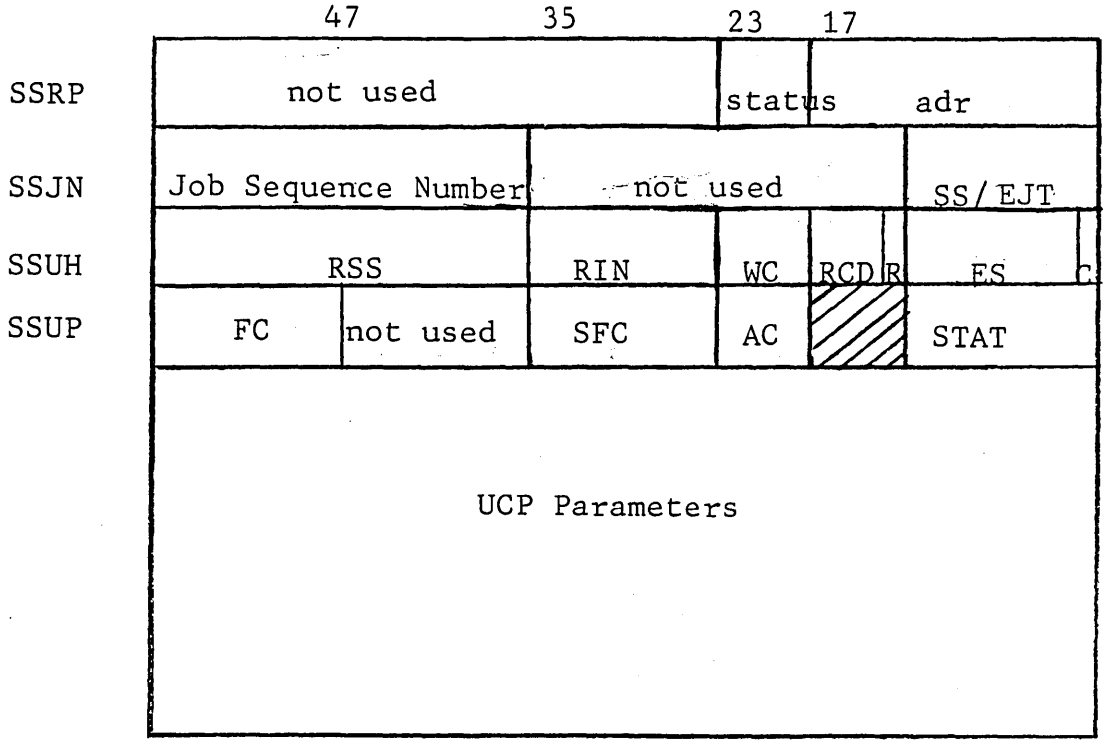
BATCH CONCURRENCY

BATCH JOBS MAY COMMUNICATE WITH THE TAF/CRM DATA MANAGER VIA SCP/UCP FACILI REQUESTS. THE USER SUBMITTING THE BATCH JOB MUST BE VALIDATED TO USE THE SCP FACILITY REQUESTS (CUCP IN VALIDUS ENTRY) AND , ALSO, BE A VALID USER OF TAF (ENTRY IN THE TST).

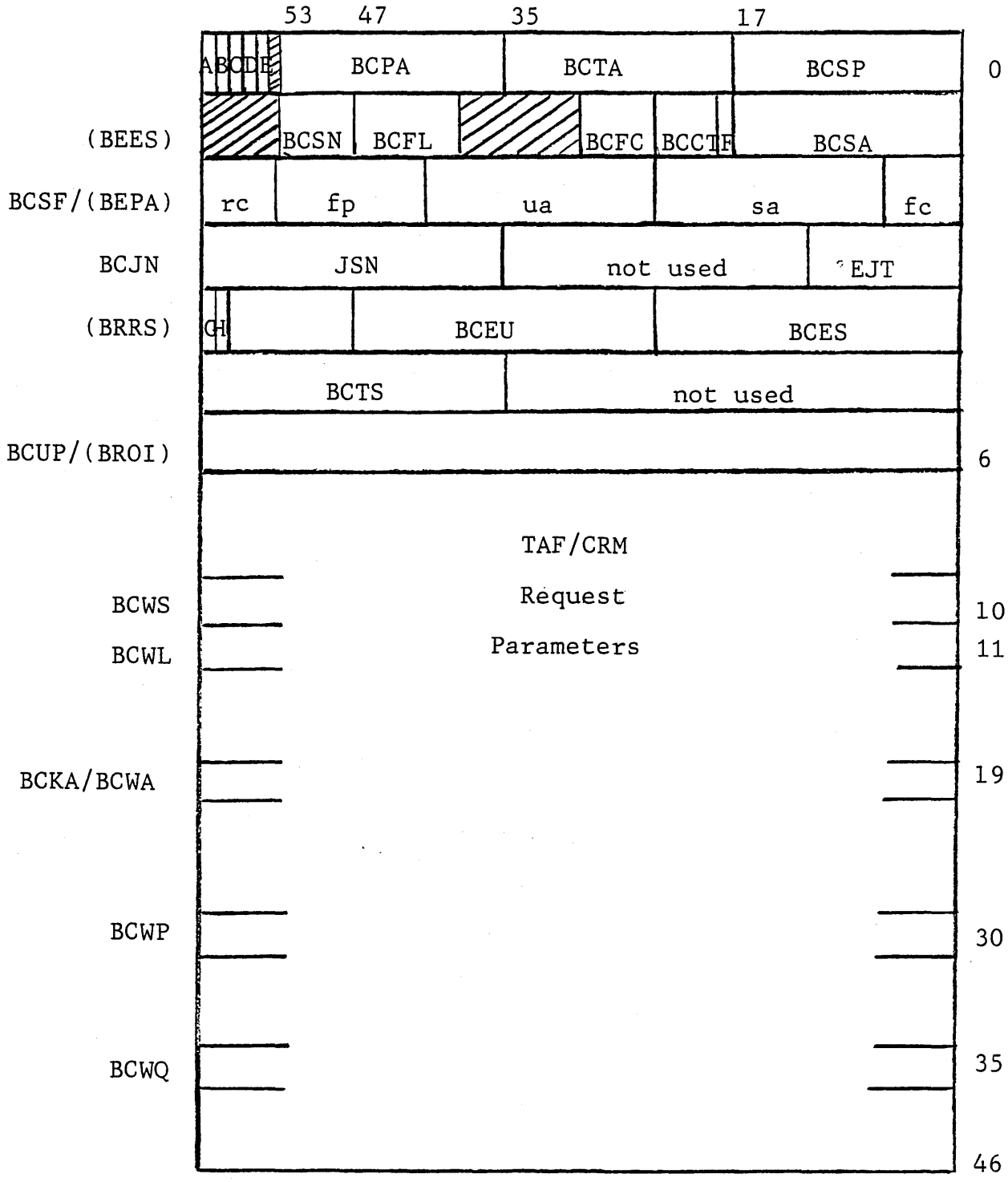
REQUESTS FROM A BATCH JOB ARE RECEIVED BY TAF IN THE SCP/UCP RECEIVING BUFFER, LOCATED AT TAG SSRP IN TAF'S LOW CORE. UPON VALIDATION OF THIS USER BY TAF A BCT ENTRY AND SUBCP TABLE ENTRY IS RESERVED FOR COMMUNICATION WITH THIS BATCH JOB. MEMORY FOR THE PARAMETERS FOR THE TAF/CRM REQUEST AS WELL AS ENOUGH MEMORY TO HOLD THE LARGEST RECORD IN ANY DATA BASE IS ASSOCIATED WITH THE SUBCP TABLE. THIS MEMORY IS ANALOGOUS WITH THAT AREA WHERE A TASK IS LOADED FOR INTERACTIVE PROCESSING.

LOW CORE POINTER WORD VBCT (BITS 24-41) CONTAIN THE FIRST WORD ADDRESS OF THE BCT.

System/User Control Point
Receiving Buffer



Batch Communication Table - BCT
(COMKBRD)

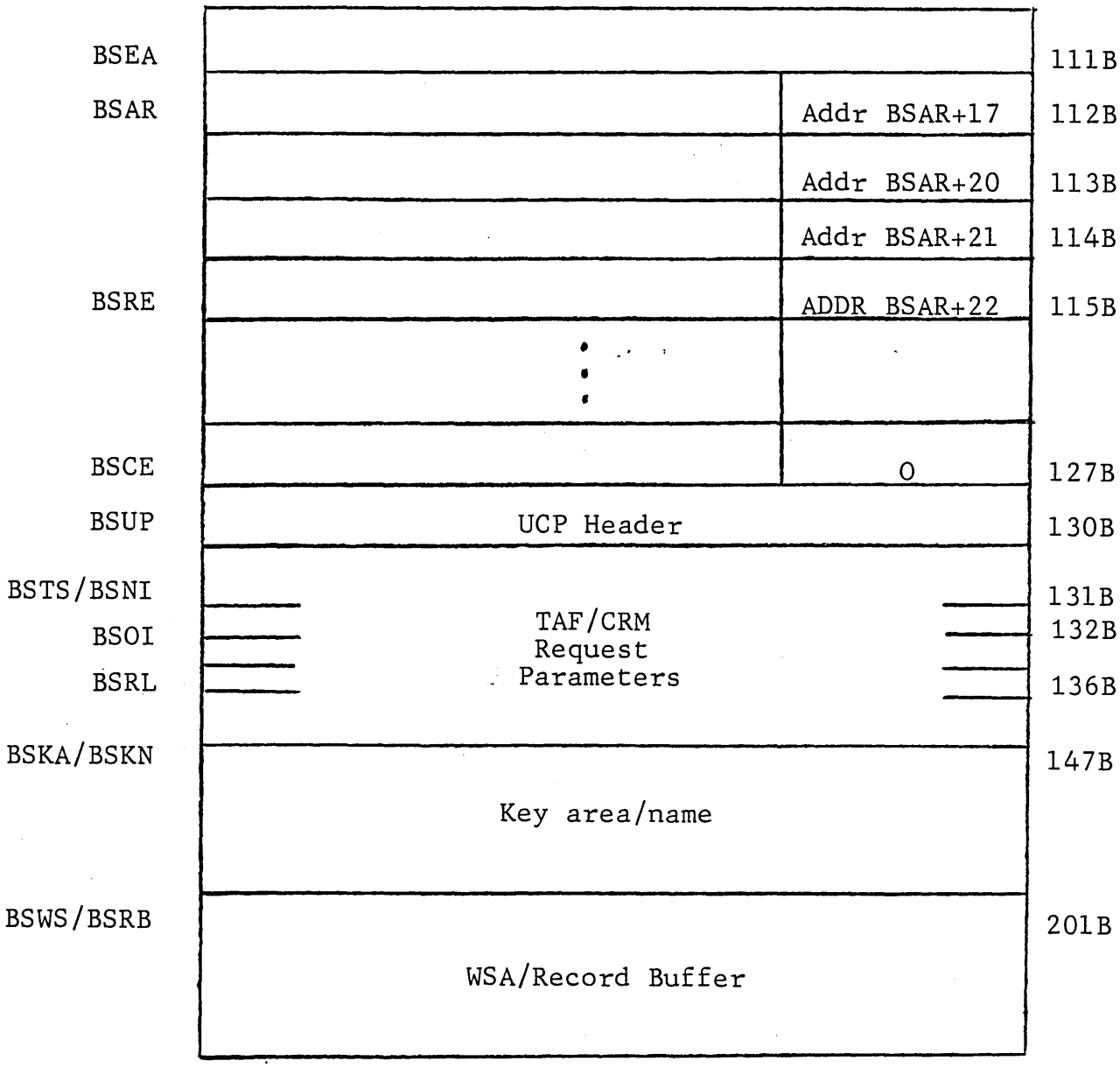


BATCH COMMUNICATION TABLE - BCT
(COMKBRD)

```
**        BATCH COMMUNICATION TABLE ENTRIES.

T  W1     1/AC,1/RA,1/AB,1/ER,1/AM,1/ ,18/PA,18/TA,18/SP
T, W2     6/ ,6/SN,9/FL,9/ ,6/FC,5/CT,1/CN,18/SA
T, W3     6/RC,12/FP,18/UA,18/SA,6/FC
T, W4     36/JS,12/ ,12/FO
T, W5     1/U,1/S,10/ ,24/EU,24/ES
T, W6     24/TS,36/
T, WN     60/UP

          WORD 1.
             AC  - 1, IF *BCT* ENTRY ACTIVE.
             RA  - 1, IF *BATCH/CRM* REQUEST ACTIVE.
             AB  - 1, IF *BATCH/CRM* USER HAS ABORTED.
             ER  - 1, IF VALIDATION ERROR.
             AM  - 1, IF *AAM* PREVIOUSLY CALLED.
             PA  - PARAMETER AREA ADDRESS WITHIN *UCP* FL.
             TA  - *TST* ADDRESS.
             SP  - SUBCP TABLE ADDRESS.

          WORD 2.
             SN  - SUBCP NUMBER.
             FL  - FUNCTION LIST.
             FC  - *TAF/CRM* FUNCTION CODE.
             CT  - CONSTRAINT TYPE.
             CN  - 1, IF REQUEST CONSTRAINED.
             SA  - *TAF* STORAGE AREA ADDRESS.

          WORD 3.
             RC  - SFCALL RETURN CODE.
             FP  - FUNCTION PARAMETER.
             UA  - RELATIVE CM ADDRESS WITHIN *UCP* FL.
             SA  - RELATIVE CM ADDRESS WITHIN *TAF* FL.
             FC  - SFCALL FUNCTION CODE.

          WORD 4.
             JS  - JOB SEQUENCE NUMBER.
             FO  - *FST* ORDINAL.

          WORD 5.
             U   - 1, *UCP* ADDRESS IS WITHIN ECS.
             S   - 1, *SCP* ADDRESS IS WITHIN ECS.
             EU  - EXTENDED *UCP* ADDRESS.
             ES  - EXTENDED *SCP* ADDRESS.

          WORD 6.
             TS  - TRANSACTION SEQUENCE NUMBER.

          WORD N.
             WORDS 7 THROUGH 57B CONTAIN THE PARAMETERS
             PASSED FROM THE *BATCH/CRM* USER.
```

```
*         BATCH COMMUNICATION TABLE FIELDS.

 BCAC     FIELD  0,59,59     *BCT* ACTIVE
 BCRA     FIELD  0,58,58     REQUEST ACTIVE
 BCAB     FIELD  0,57,57     *BATCH/CRM* USER ABORTED
 BCER     FIELD  0,56,56     VALIDATION ERROR
 BCAM     FIELD  0,55,55     *AAM* PREVIOUSLY CALLED
 BCPA     FIELD  0,53,36     PARAMETER ADDRESS WITHIN *UCP*
 BCTA     FIELD  0,35,18     *TST* ADDRESS
 BCSP     FIELD  0,17,00     SUBCP TABLE ADDRESS
 BCSN     FIELD  1,53,48     SUBCP NUMBER
 BCFL     FIELD  1,47,39     FUNCTION LIST
 BCFC     FIELD  1,29,24     *TAF/CRM* FUNCTION CODE
 BCCT     FIELD  1,23,19     CONSTRAINT TYPE
 BCCN     FIELD  1,18,18     REQUEST CONSTRAINED
 BCSA     FIELD  1,17,00     TEMPORARY STORAGE AREA ADDRESS
 BCSF     FIELD  2,59,00     WORD ONE OF SFCALL PARAMETERS
 BCJN     FIELD  3,59,00     JOB SEQUENCE NUMBER AND *FST* ORDINAL
 BCEU     FIELD  4,47,24     EXTENDED *UCP* ADDRESS
 BCES     FIELD  4,23,00     EXTENDED *SCP* ADDRESS
 BCTS     FIELD  5,59,36     TRANSACTION SEQUENCE NUMBER
 BCUP     FIELD  6,59,00     BEGINNING OF *UCP* PARAMETERS
 BCWS     FIELD  10,59,00    WORKING STORAGE AREA ADDRESS
 BCWL     FIELD  11,59,00    RECORD LENGTH FOR WRITE REQUESTS
 BCKA     FIELD  19,59,00    KEYAREA
 BCWA     FIELD  19,59,00    *WSTAT* ARGUMENT ARRAY
 BCWP     FIELD  30,59,00    *WSTAT* PARAMETER ARRAY
 BCWQ     FIELD  35,59,00    *WSTAT* QUEUEING AREA

*         *TSTAT* REQUEST AREA FIELDS.

 BRTO     FIELD  2,35,18     *TST* ORDINAL
 BRTA     FIELD  2,17,00     *TST* ADDRESS
 BRBI     FIELD  3,41,36     BATCH/BTRAN INDICATOR
 BRTS     FIELD  3,35,18     *TAF* STORAGE AREA ADDRESS
 BRRA     FIELD  3,17,00     RETURN ADDRESS
 BRRS     FIELD  4,59,00     RECOVERY REQUEST STATUS
 BROI     FIELD  6,59,00     OLDID
 BRNI     FIELD  8,59,00     NEWID
 BRTR     FIELD  10,59,00    TRANSACTION TYPE
 BRST     FIELD  12,59,00    STEP

*         *TAF* STORAGE AREA FIELDS FOR BATCH ERROR PROCESSING.

 BEES     FIELD  2,59,00     ERROR STATUS RETURN
 BEPA     FIELD  3,17,00     PARAMETER AREA ADDRESS
```

Batch Concurrency Job's
Subcontrol Point Area



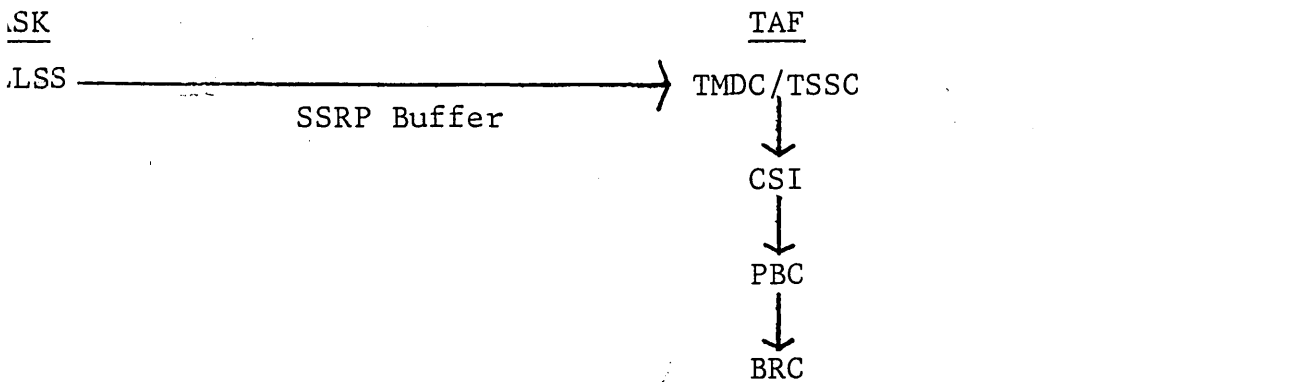
TAF/CRM Batch Concurrency Processing

- Validate Batch Job (UCP)
  - Get TAF Storage
  - Perform Validation Check
  - Set Long Term Connection
  - Request Subcontrol Point

- Process Request(s)
  - Queuing TAF/CRM requests for AAMI
  - Read/Write record(s) from/to UCP
  - Writing status to UCP
  - Write recovery information to CRF (DBCOMIT)
  - Issue SF.ENDT to UCP

- Job Termination
  - Clear long term connection
  - Write recovery information to CRF (STEP)
  - Issue Data Manager Cease
  - Release SCP/BCT
  - Set user inactive in TST
  - Release TAF storage

TAF/CRM Batch Concurrency Processing



1. Validate Batch Job
   VBA ⟶ (QTW/TFP) ⟶ SFR ⟶ ASN ⟶ next function

2. Request SCP
   RCP ⟶ [BAM(TRRI)] ⟶ next function

⟶ 3. Read Record from UCP
   SFR ⟶ next function

4. Queue TAF/CRM Request
   BAM ⟶ next function

⟵ 5. Write Record to SCP
   SFR ⟶ next function

6. Write commit history to CRF
   QTW ⟶ WFP ⟶ ... ⟶ next function

⟵ 7. Complete task
   SFR ⟶ next function

⟵ 8. Clear Long Term Connection
   SFR ⟶ QTW ⟶ WFP ⟶ BAM ⟶ ESCP ⟶ RTS

TSSC

DATA BASE RECOVERY

- Automatic Component
  1. Processes
  2. TAF/CRM - CTASK interface
  3. Batch Recovery Interface

- Batch Component (DMREC)
  1. Processes
  2. Interface to TAF/CRM

- Data Base Status
  1. States
  2. CRMTASK

- Recovery Considerations

AUTOMATIC COMPONENT PROCESSES

- Re-establish tables upon recovery
- Log before images to BRF
- Log after images to ARF
- Roll back uncommitted updates
- Switch after image log files

TAF/CRM - CTASK INTERFACE

- Interactive / Batch (BTRAN) Users
  1. During terminal login / BTRAN request processing
  2. TINVOKE → New transaction sequence number
  3. RSTDBI → Restore begin identifiers
  4. SRERUN → Rerun transaction

- CRM Date Base Recovery / Terminal Failures
  1. During TAF initialization/after terminal or network failure.
  2. CRMSTAT → Obtain TAF/CRM table information
  3. TSTAT → Obtain transaction environment.
  4. WSTAT → Write begin-commit history to CRF
  5, TRMREC → Roll back data base
  6. WSTAT → Terminate TAF or start input

BATCH RECOVERY INTERFACE

- Batch Job submitted when:

  1. An ARF must be dumped to tape
  2. TAF/CRM detects a defective data base file
  3. TAF/CRM detects a defective ARF or BRF

- DMREC notifies TAF/CRM of successful completion.

BATCH COMPONENT PROCESSES

- Dump full ARF's to tape
- Dump data base files to tape
- Restore data base files
- Maintain a directory of dumped files

INTERFACE TO TAF/CRM

- SIC RA+1 Request
  - Upon successful completion only

- TAF initiates CRMTASK
  - Issues CRMSIC request

DATA BASE STATUS

- States
  1. UP
  2. IDLE
  3. DOWN

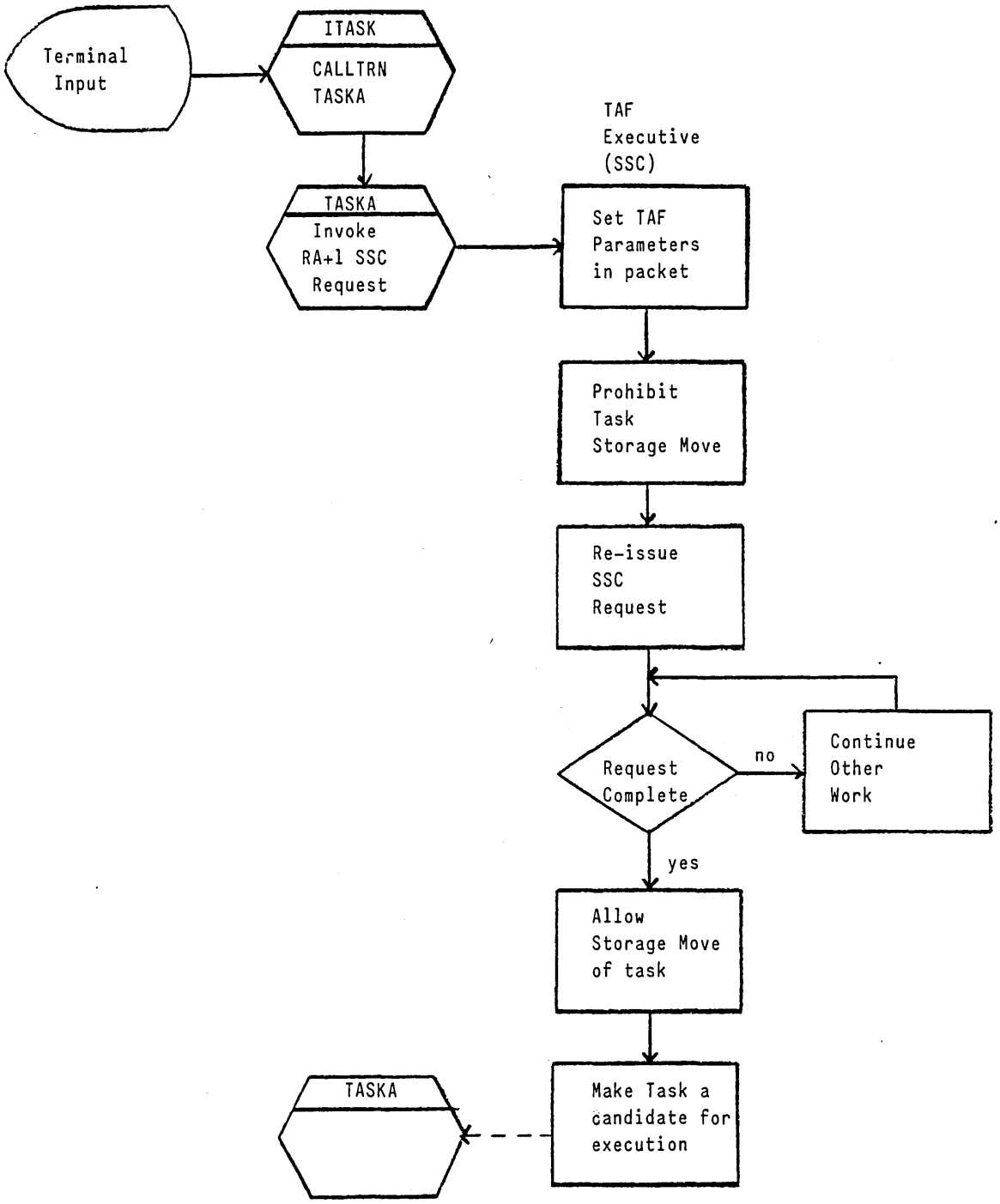
- CRMTASK
  1. K-display commands
  2. DMREC SIC request
  3. Terminal initiated

RECOVERY CONSIDERATIONS

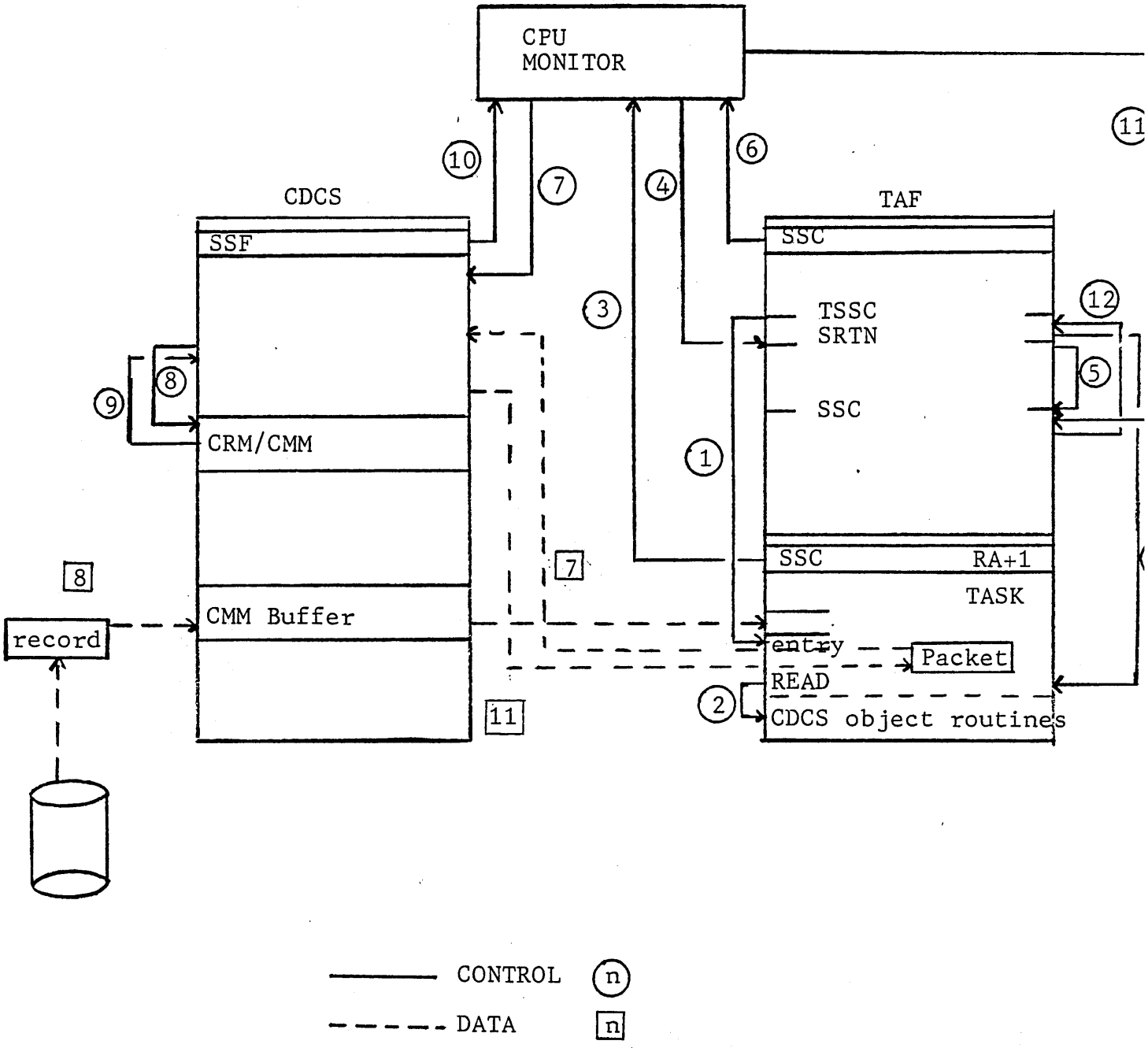
- Defining recoverable files
- Number of before image recovery files
- Size of after image recovery files
- Preallocation of data base files
- Recovery files on different disks/disk channels
- Frequency of data base file dumps
- Installation parameters for DMREC
  - TTIGL
  - FTABL
  - WBUFL
  - NUMARF
  - NCOPY
- TAF/batch DMREC job dayfiles
- Manual Maintenance of Directory

TAF/CDCS INTERFACE

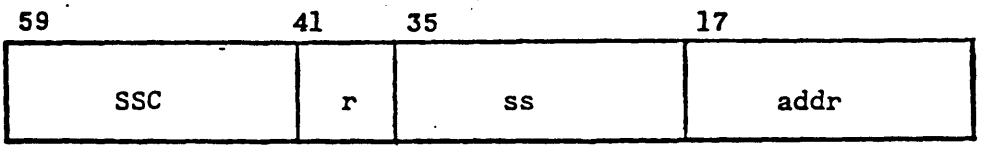
PROCESSING FOR CDCS TRANSACTIONS

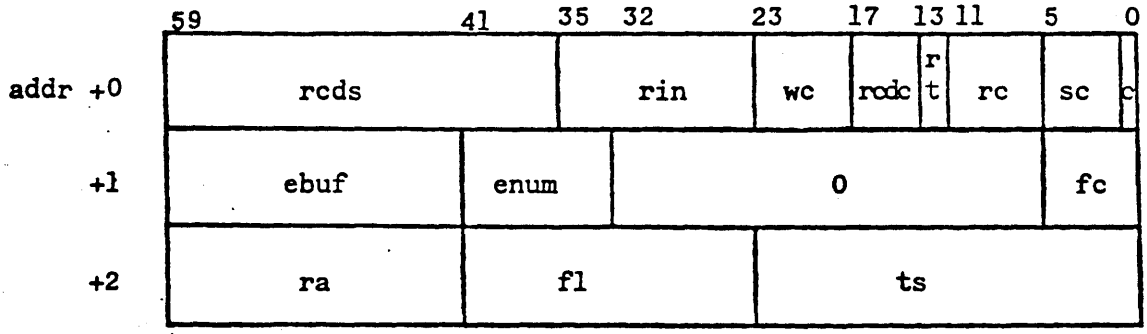


TAF/CDCS Processing



SSC REQUEST FORMAT





RECOVERY ASPECTS

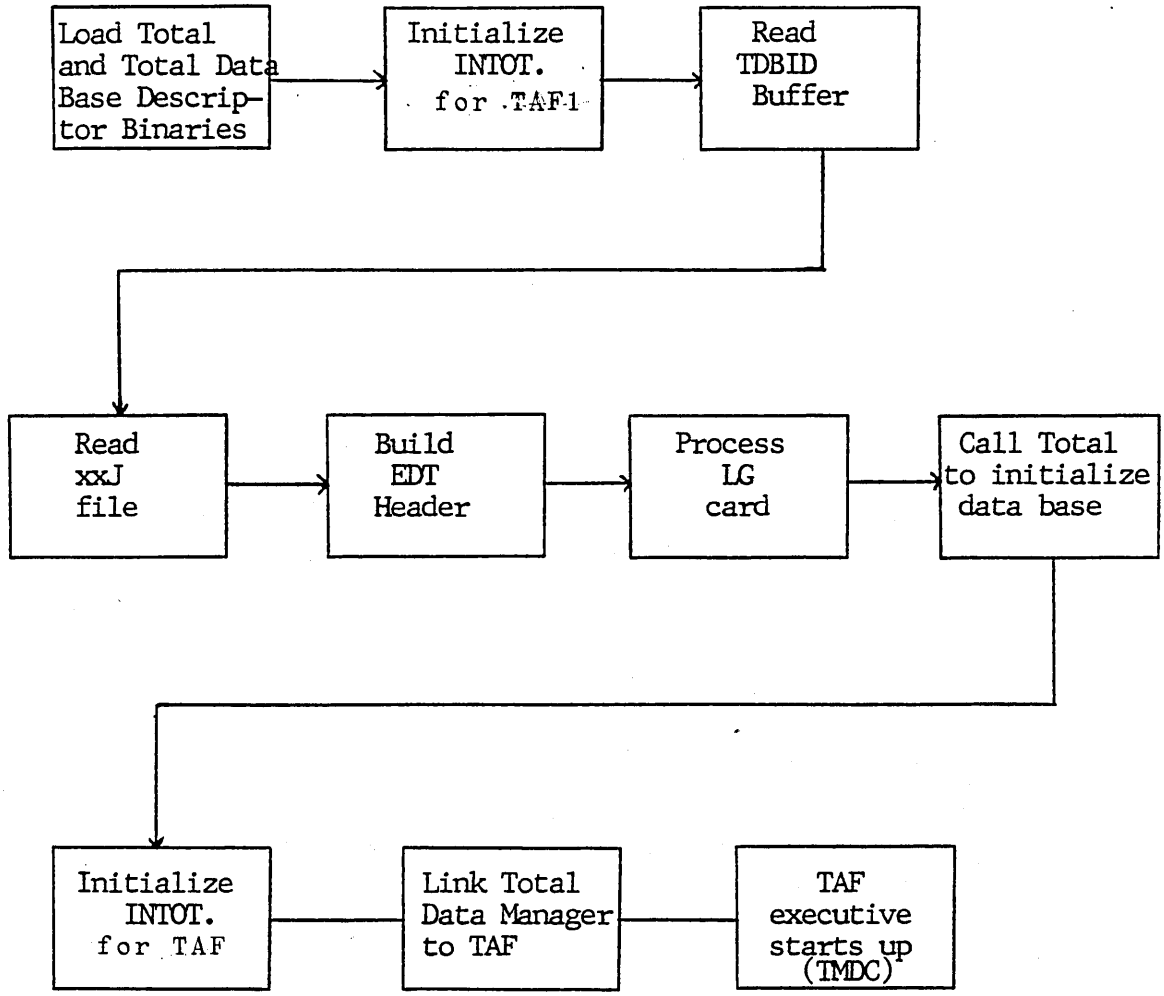
- Data base recovery
  - CDCS automatic recovery
  - Task requests

- Communication recovery
  - Recoverable transactions
  - CDCS specified as data manager

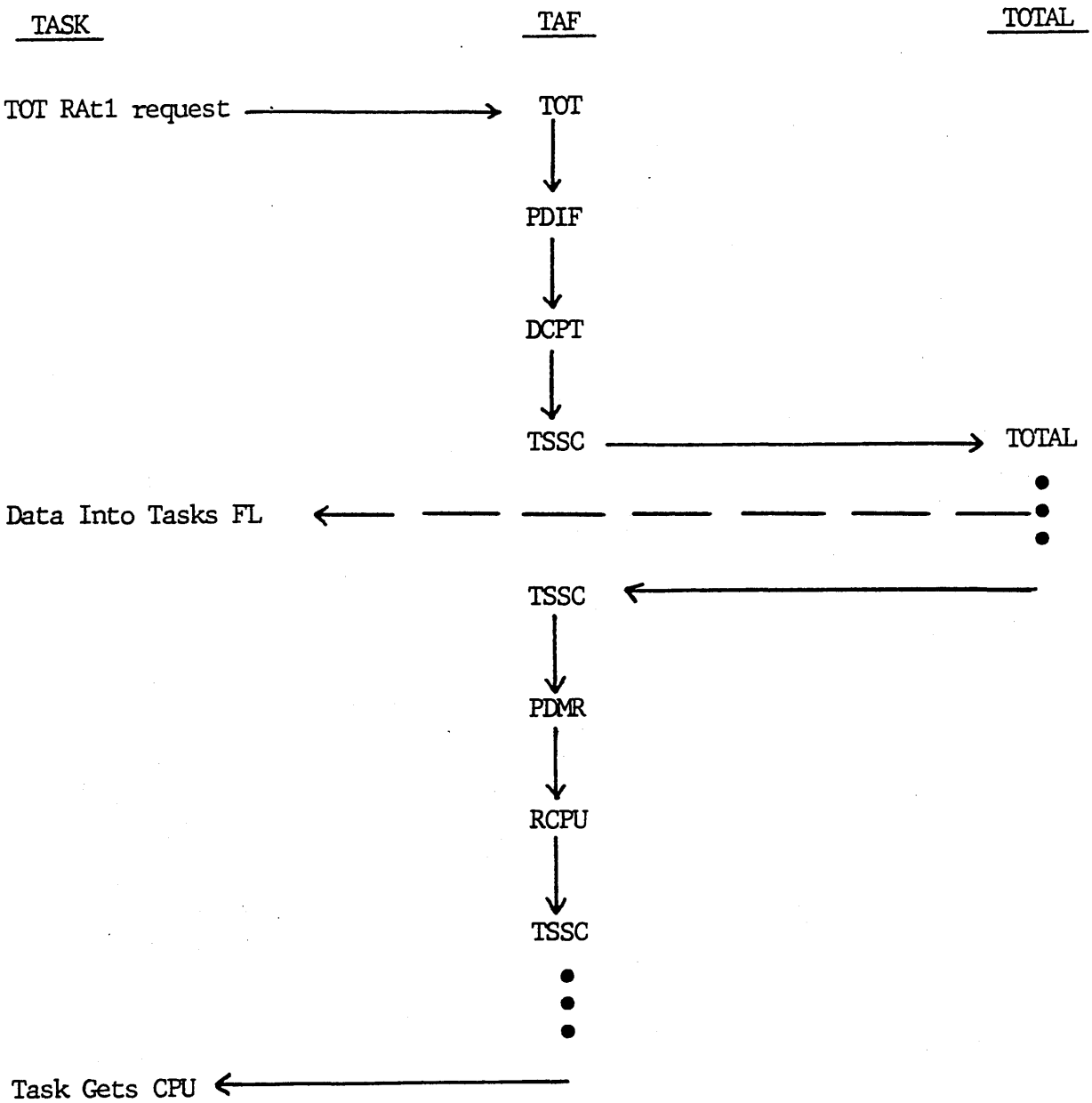
- RCTASK
  - Recovers rerunnable CDCS transactions
  - Scheduled whenever CDCS becomes active
  - Reruns transaction

TAF/TOTAL INTERFACE

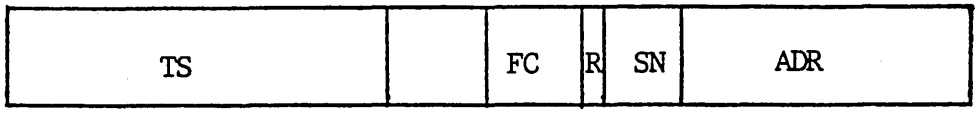
TOTAL INITIALIZATION



TOTAL DATA MANAGER INTERFACE

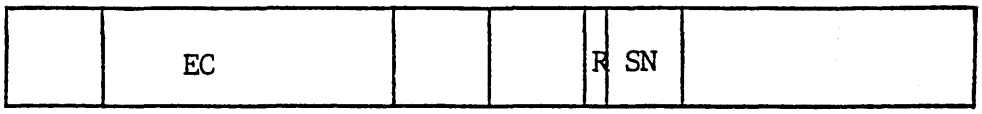


IDMIQ



TS - Transaction Sequence Number
FC - Request Function Code
R - Recall Flag (always set)
SN - Sub-control Point Number
ADR - FWA of Request Parameters

DMOQ



EC - Error Code
R - Recall Flag
SN - Sub-control Point Number

TAF/NAM INTERFACE

TAF/NAM COMMUNICATION

- Parallel mode
- Half duplex
- Input to communication block
- Two connection lists
- Additional buffers
    1. INDB - NETGETF
    2. INSB - input supervisory messages
    3. OTSJ - output supervisory messages
- NET macro
- COMKNWC

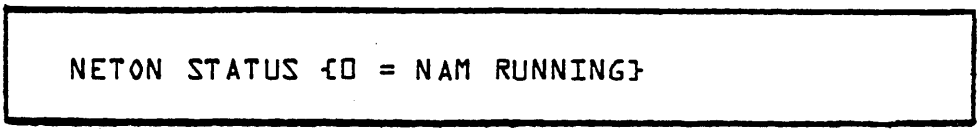
NSUP - Network Supervisory Status Word

*1 - bit 59 - Complete
56 - Input in queue
55 - Supervisory msg. in queue

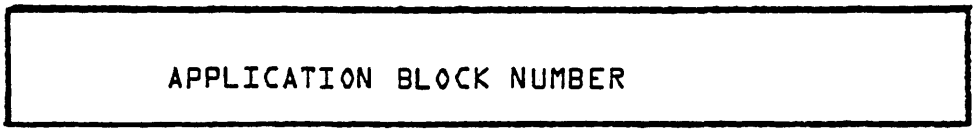
NM - number of msgs. on debug log file

VNCT

VNON



VSND



ABH - Application Block Header

Input/Output Data

ABT - Application block type
0 = NULL (input only)
1 = BLK/BLK
2 = MSG/BLK (last block in msg)

ACN - Application connection number

ABN - Application block number

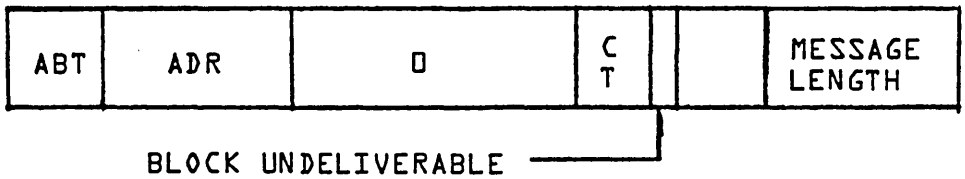
CT - Character type
2 = 8/8 (7.5 per word)
3 = 8/12 (5 per word)
4 = 6/6 (10 per word)

*1 bit 19 - block undeliverable
18-16 - Reserved
15 - No format effectors
14 - Input in transparent mode
(input) 13 - Input to be canceled
(input) 12 - Input has parity error
(output) 12 - Auto input

TL - Text length

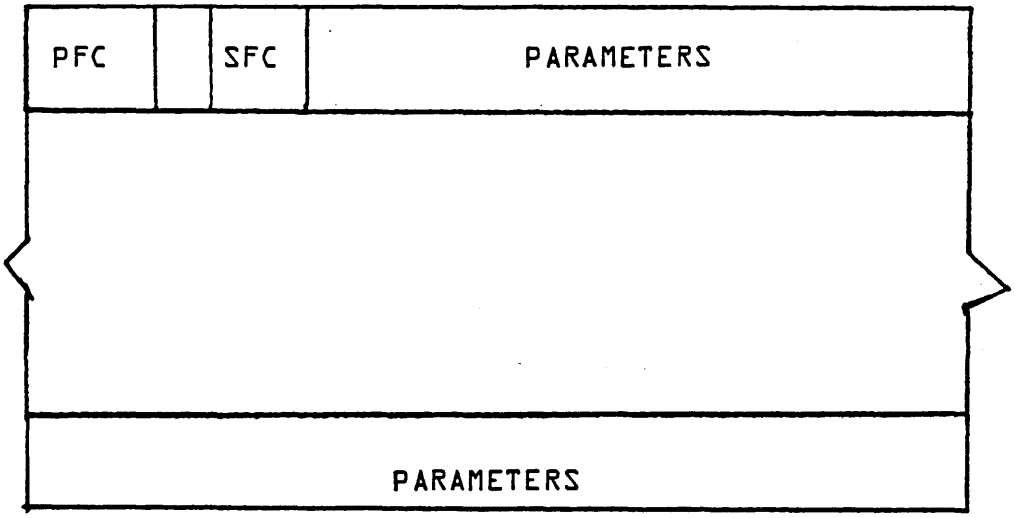
SUPERVISORY MESSAGES

BLOCK HEADER



ABT - Application Block Type
0 = NULL {INPUT}
3 = SUPERVISORY
ADR - 0 = TO/FROM HOST
ACN = TO/FROM CONNECTION

MESSAGE



PFC - Primary Function Code
SFC - Secondary Function Code

AIP CALLS

NETON	NAME, NSUP, STATUS, MINACN, MAXACN
NETOFF	
NETPUT	HEADER, TEXT
NETGETL	CONNECTION-LIST, HEADER, TEXT, LENGTH
NETGET	ACN, HEADER, TEXT, LENGTH
NETGETF	ACN, HEADER, NUMBER-OF-FRAGMENTS, TEXT
NETCHEK	

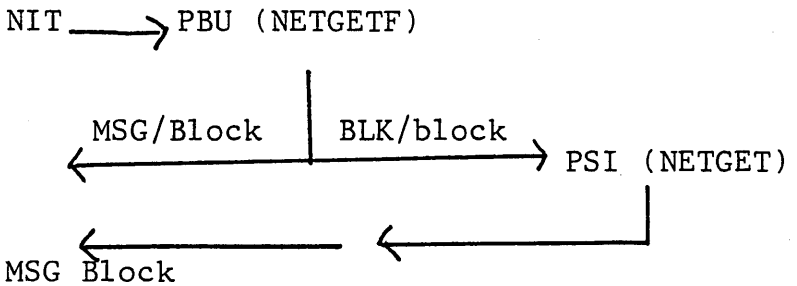
TAF/NAM INTERFACE

- Check for network Input
  1. TMDC
  2. TRO

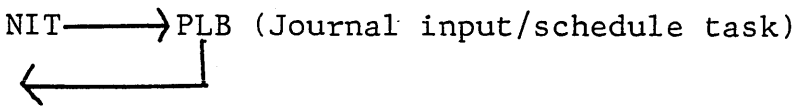
- Routine NGL
  1. Data messages (NIT)
  2. Supervisory messages (SMP)

DATA MESSAGES

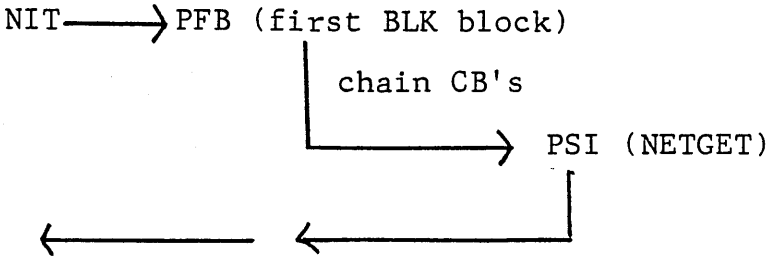
- Input Block Undeliverable



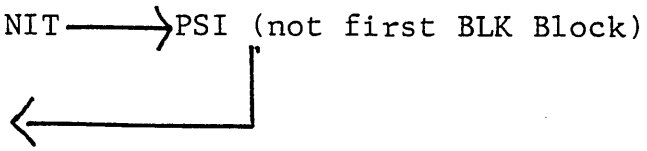
- MSG Block



- BLK Block



- BLK Block



SUPERVISORY MESSAGES

- Routine SMP

- ITASK or originating task (SEND W/RECALL)

- Queue supervisory messages
    1. Unable to schedule task
    2. Routines SMQ/SMR

RECOVERY REVISITED

COMMUNICATION AND DATA BASE RECOVERY

- Recoverable Transactions
  1. Initial terminal input
  2. TAF/CRM
  3. CDCS interface
  4. TOTAL or No data Manager
- Distinguished Participants
  1. TAF executive
  2. TAF/CRM
  3. CDCS and its utilities
  4. System Tasks
  5. DMREC
  6. Data Base Administrator

RECOVERY SITUATIONS

- Communication Failure
  - NAM
  - Line/Terminal
- TAF Failure
- Operating System Failure
- CDCS Failure
- Data Base Failure (TAF/CRM)
  - Data file
  - Recovery file(s)
- Combination of any of the above

COMMUNICATION FAILURE

1. TAF ⟶ RAN ⟶ TAFG
   - TAFG sets recovery flag in all active users TST
   - Tasks waiting on input will be rolled in immediately

2. Subsequent network requests cause initiation of CTASK out of routine, PNT

3. CTASK
   - Recover TAF/CRM Begin - Commit history

     TSTAT ⟷ CRMSTAT ⟷ WSTAT
   - Roll back data base

     TINVOKE ⟶ TRMREC
   - Start terminal input

     WSTAT

4. Upon login TAF will initiate RTASK to either send
   - READY
   - Transaction not rerunnable
   - Secure message
   - Call CTASK to rerun

5. CTASK
   - TINVOKE
   - RSTDBI
   - SRERUN

TAF or OPERATING SYSTEM FAILURE

1. TAFREC
   - Sets recovery flag in TST for all active users

2. TAF (PRE)
   - Calls IAM (AAMI) to restore abnormally terminated tasks
   - Initiates CTASK

3. CTASK
   - Recover TAF/CRM Begin-Commit history

     CRMSTAT ⟶ TSTAT ⟶ WSTAT
   - Roll back data base

     TINVOKE ⟷ TRMREC
   - CALLRTN to BTASK
   - Start terminal/job input

4. Upon login TAF will initiate RTASK to either SEND
   - READY
   - Transaction not rerunnable
   - Secure message
   - Or call CTASK to rerun

5. CTASK
   - TINVOKE
   - RSTDBI
   - SRERUN

CDCS FAILURE

1. TAF will set CDCS abort flag in SCP table

2. When CDCS becomes active TAF schedules RCTASK

3. RCTASK
   - Read entire CRF with TSTAT requests
   - BTRAN transactions are rerun
   - Interactive transactions are rerun if user active

DATA BASE FAILURE (TAF/CRM)

- Data File
  - Only DBCOMIT/DBREE/CLOSE
  - No more active users ⟶ file down
  - Batch job (DMREC)

- ARF failure
  - Active begin-commit sequences allowed to finish
  - No after image records are written
  - Batch job (DMREC) submitted ⟶ D. B. down
  - DMREC ⟶ Dump/Reallocate ARF
  - Data base administrator should dump data file

- BRF failure
  - No before image logging - begin/commit sequences left incomplete
  - Batch job (DMREC) submitted ⟶ D. B. down
  - DMREC ⟶ Dump ARF/Recover D. B/Reallocate BRF
  - Data base administrator intervention may be needed

MULTIPLE TYPES OF FAILURES

- Requires analysis
  - TAF/DMREC job/System dayfiles
  - List ARF headers (last ARF dump tape)
  - Information from JOURØ file
- Manual initiation of DMREC
  - Use of IGNORE directive

TAF INSTALLATION PARAMETERS AND TUNING

INSTALLATION PARAMETERS

- Reside in Decks
  - COMKIPR
  - TAF
  - COMKNWC
  - DMREC

TAF TUNING GUIDELINES

- TAF's two most valuable resources
  1. Communication blocks
  2. Memory
- Tuning - Where?
  1. Application
  2. TAF
  3. NOS

APPLICATION TUNING

- Application Tasks
  - Short in CPU time/field length
  - Pass information in CB not data files
  - Release resources ASAP
  - Avoid unnecessary time consuming requests
    1. Data Manager
    2. Send w/recall
    3. JOURNL
  - Avoid requests that keep resources reserved
    1. WAIT
    2. WAITINP
    3. File/record lock requests
  - Task attributes
    1. Queue limit
    2. Reusable
    3. ECS resident
    4. CM resident

APPLICATION TUNING (Con't)

- Application Files
  - Same file structure if possible
  - Small records/key size
  - Few index levels
  - MBL consistent across all files

TAF TUNING

- RRTTL - Time to evict a releasable task
- DMMTL - Min milliseconds between D. M. calls
- ITRTL - Milliseconds to idle before rollout
- RTDNL - Core time slice allowed after CALLRTN
- CORTL - Milliseconds between FL reductions
- RFLTL - Milliseconds after decrease (FL) before increase
- FCMFL - Minimum FL to retain
- REDFL - Maximum FL for one reduction
- INCFL - Minimum FL for one increment
- NCMB - Number of communication blocks
- NSCP - Number of sub-control points
- CMMBFL (K. BFL) - Min size of CMM buffer
- CMMEFL (K. EFL) - Additional memory available for CMM buffer
- MAXMFL/SCMFL - TAF's Maximum running FL

NOS SYSTEM TUNING

- Dual access all mass storage devices
- Separate heavily used files from system device
  - KTSROLL
  - Task libraries
  - Data base files
  - Recovery files
  - JOUR0
- Separate TEMP devices from SYSTEM/PF
- Two copies of system file on different device/channel

ANALYSIS OF TAF PROBLEMS

TAF PROBLEM ANALYSIS

- Dayfile/TAF Initialization and Recovery Report
- TAF's XP at EXIA + 11B
- B2-TSA of currently executing Subcp (SREG)
- B7 - Subcp entry of currently executing task (SREG)
- Internal trace buffer - PBUF (INTRACE)
- V - words

  CB - VCBRT/VCBSA
  TST - VTST
  Subcp - VCPA
  TLD - VTLD
  BCT - VBCT

- Task System Area

  XJPC
  CB1C/CB2C
  LRA1
  RCL
  ERRC
  DMEC
  AAMC
  CDFN

- AIP Debug File
- TAF Storage buffer (TSBM/TSB)
- CR - Subcp's that are candidates for execution
- RCR - Subcp's in recall
- AVAILCM - Available memory within TAF's FL
- STIN - Communications control word
- TAFQ - TAF queue pointer
- AAMA - Number of outstanding AAM requests (TDBAA)
- SSCC - Number of outstanding SSC requests
- TCMA - Last CMM request for memory
- LOVA - Current overlay in core

- TAF/CRM (AAMI)

  REQT

  RFCB

  RLNT

  RSEQ

  RDRF

  AMST

  AMIQ/AMOQ - VAAQ

  VAAM

  VAMB

TAF PSR SUPPORT MATERIAL

- Field Length Dump/Dayfile
- Network Trace File (ZZZZZDN) on Tape
- Listings to Match Field Length Dump (TAF/AAMI)
- Task Rollout File (KTSROLL) on Tape
- Failing Test Case
- Data Base(s), Task Library(s), Initialization Related Files for Initialization Problems
- Recovery files for recovery related problems - ARF/BRF/CRF
- As complete a description as possible
- Suggested code that may solve the problem

EXERCISES

EXERCISES

1. How do the following installation parameters affect performance?

   NCMB
   NSCP
   RRTTL
   RTDNL
   DMMTL
   ITRTL
   CORTL
   SCMFL
   REDFL
   INCFL

2. How can the TAF analyst help the application programmer solve a problem in his task?

3. Where does the input from NAM get stored within TAF's field length?

4. How does TAF dynamically build tables during initialization and know where they are for execution?

5. Where is the task's exchange package kept?

6. What is the purpose of the sub-control point table?

7. What stays constant throughout a transaction as different tasks get called to process input?

8. What is meant by a reusable task? What programming considerations should there be when defining a task as reusable? What is meant by a reusable task from TAF's point of view?

9. What tables show which communication blocks are queued for execution on a reusable task?

10. Where would you look to find tasks waiting to be scheduled?

11. If there were tasks being requested, but not scheduled, what could be a possible explanation?

12. How can you determine which RTL entry corresponds to which CB?

13. All reserved CB's have to be associated with a task that is in some state of execution or waiting to execute. Where are the places (tables) this task could be located? Knowing the CB address, how could you locate the task associated with this CB?

14. How could the queue limit attribute for a task affect performance?

15. If you have a TAF dump and want to know the last thing a certain task at a Subcp did, where would you look?

16. How can you determine if a task is in recall and why it is in recall?

17. How does TAF know if input from a terminal is a new transaction or response to a WAITINP request?

18. How can you determine if a task at a Subcp is waiting for the CPU?

19. Which routine gives the CPU to a task?

20. Which registers should NEVER be touched unless they are restored?

21. What happens when a task makes an RA+1 request? exceeds its time limit? gets a mode error?

22. Where are flags, table entries, etc. Kept for CALLRTN tasks with nest levels greater than one?

Use the TAF dump for the following exercises

1. Locate the following tables:

   Subcp
   Start of CB's
   RTL
   ATL
   TLS
   ROLT
   TLD
   TRD
   TST
   TTFT
   TTRF
   TDRF
   TLNT
   TFCB
   TKOK
   TSEQ

2. What tasks are present at Subcp's? Which of these are active (CB's queued)? Any tasks being loaded?

3. Is the CPU assigned to a task?

4. How many CB's are in use?

5. Are there any tasks in recall? If so, why?
   Are there any waiting for the CPU?

6. Are there any outstanding TAF/CRM requests?

7. Are there any tasks rolled out? If so, why?

8. What are the names of the transactions in the
   TRD? Any recoverable transactions defined?

9. Are there any batch concurrency jobs active?

10. What was the last RA+1 request? From which sub-control point?

11. At what address did TAF detect an internal error condition?

		Location	
TERMINAL STATUS TABLE	VTST	12	VII-37
ACTIVE TRANSACTION LIST	VATL	16	VII-24
COMMUNICATIONS BLOCKS	VCBSA	21	VII-2
TASK LIBRARY DIRECTORY	VTLD	22	VII-28
ELEMENT DESCRIPTOR TABLES	VEDT	23	VII-54
FIRST SUBCONTROL POINT TABLE	VCPA	36	VII-17
FETS FOR TAF CRM QUEUES	VAAQ	42	XV-39
REQUESTED TASK LIST	VRTLW	53	VII-21
NAM COMMUNICATION TABLE	VNCT	55	VII-44
Pointer Description			VII. 56, 57, 58
INTERNAL TRACE Packet Building			VII. 51

```
 SCR1       BSS    0                                                     TAF
 SCR1       FILEB  OBUF,OBUFL,EPR,FET=14 SCRATCH FILE FOR TAPE ASSIGNMENT TAF
                                                                         TAF
 PJRNL      BSS    0                                                     TAF
 JOURO      FILEB  JBUFO,JBUFL,FET=15      FOURNAL FILE FET              TAF31
                                                                         TAF
 RO         BSS    0                                                     TAF
 KTSROLL    RFILEB OBUF,OBUFL,(FET=8)  TASK ROLLOUT FILE                 TAF
                                                                         TAF
 INT        BSS    0                                                     TAF
 INTRACE    FILEB  PBUF,PBUFL  INTERNAL TRACE BUFFER FET                 TAF
                                                                         TAF
*           THE FOLLOWING STATISTICS ARE MAINTAINED FOR TUNING PURPOSES. TAF
                                                                         TAF
 STAT1      BSSZ   1            NUMBER OF TIMES A TASK WAS RELOADED      TAF
 STAT2      BSSZ   1            NUMBER OF TIMES INITIAL TASK WAS RELOADED TAF
 STAT4      BSSZ   1            NUMBER OF STORAGE MOVES OF TASKS         TAF
 STAT5      BSSZ   1            NUMBER OF TIMES A TASK ABORT OCCURRED    TAF
 STAT6      BSSZ   1            NUMBER OF TIMES *TAF* FL WAS INCREASED   TAF
 STAT9      BSSZ   1            RECALLS FOR *NAM* OUTPUT BLOCK LIMIT     TAF
 STAT10     BSSZ   1            NUMBER OF TIMES NO FL FOR TASK LOAD      TAF
 STAT11     BSSZ   1            NUMBER OF TIMES NO AVAILABLE SUBCP       TAF
 STAT12     BSSZ   1            TIMES NO COMMUNICATION BLOCKS AVAILABLE  TAF
 STAT13     BSSZ   1            NUMBER OF TASK ROLLOUT COMPLETES         TAF
 STAT14     BSSZ   1            NUMBER OF ROLLOUT INITIATIONS FOR TASKS  TAF
 STAT15     BSSZ   1            NUMBER OF TIMES TASK IN RECALL           TAF
 STAT16     BSSZ   1            NUMBER OF ACTIVE SUBCONTROL POINTS       TAF
            BSSZ   1            NUMBER OF SAMPLES                        TAF
 STAT17     BSSZ   1            NUMBER OF OUTSTANDING *CDCS* REQUESTS    TAF
 STAT18     BSSZ   1            NUMBER OF *CDCS* REJECTS FOR *MAXR*      TAF
 STAT19     BSSZ   1            NUMBER OF *CDCS* REQUESTS REJECTS FOR BUSY TAF
 STAT20     BSSZ   1            NUMBER OF *CDCS* TASK REQUESTS           TAF
                                                                         TAF
*           *TAF* TIMERS.  THE FOLLOWING TIMERS ARE USED TO SCHEDULE     TAF
*           PROCESSES IN *TAF*.                                          TAF
                                                                         TAF
*           ASSEMBLY CONSTANTS FOR TIMED LOOPS.                          TAF
                                                                         TAF
 TMDTL      EQU    1008         MILLISECONDS BETWEEN TIME DEPENDENT CALL TAF
 RCLTL      EQU    20B          MILLISECONDS FOR TASK RECALL             TAF
 SCHTL      EQU    20B          MILLISECONDS BETWEEN TIMED SCHEDULER RUNS TAF
 SICTL      EQU    4            MINIMUM MILLISECONDS BETWEEN *SIC* CALLS TAF
 SFCTL      EQU    1*1000       MILLISECONDS BETWEEN STATUS REQUESTS     TAF
                                                                         TAF
*           THE FOLLOWING MILLISECOND COUNTS MAY EXCEED 131,000 AND      TAF
*           THEREFORE REQUIRE A FULL WORD TO CONTAIN THEIR VALUE.        TAF
                                                                         TAF
 TACTL      CON    2*60*1000    MILLISECONDS BETWEEN ACTIVITY CHECKS     TAF
 SJTTL      CON    20*60*1000   MILLISECONDS BETWEEN PERIODIC JOURNALING TAF
 ITPTL      CON    15000        MILLISECONDS TO IDLE BEFORE ROLLING OUT  TAF
```

TEST FILE ENVIRONMENT TABLE