DECSYSTEM-20
SWS MONITOR INTERNALS
MONITOR TABLES
PRELIMINARY VERSION

Monitor Program Logic Manual

DIGITAL

The material in this document is for informational purposes and is subject to change without notice; it should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may only be used or copied in accordance with the terms of such license. Digital Equipment Corporation assumes no responsibility for the use or reliability of its software on equipment that is not supplied by Digital or its affiliated companies.

The following are trademarks of Digital Equipment Corporation, Maynard, Massachusetts:

COMPUTER LABS	COMTEX	DBMS-10
DBMS-11	DBMS-20	DDT
DEC	DECCOMM	DECsystem-10
DECSYSTEM-20	DECtape	DECUS
DIBOL	DIGITAL	EDUSYSTEM
FLIPCHIP	FOCAL	INDAC
LAB-8	MASSBUS	OMNIBUS
OS/8	PDP	PHA
RSTS	RSX	TYPESET-8
TYPESET-10	TYPESET-11	TYPESET-20
UNIBUS	DECSYSTEM-2020	

<<For Internal Use Only>>

DECSYSTEM-2Ø MONITOR FLOWCHARTS

I. Scheduler

II. Page Fault Handling

III. JSYS Calls - Device Independent Level

IV. JSYS Calls - Disk Dependent Level

V. JSYS Calls - Magtape Dependent Level

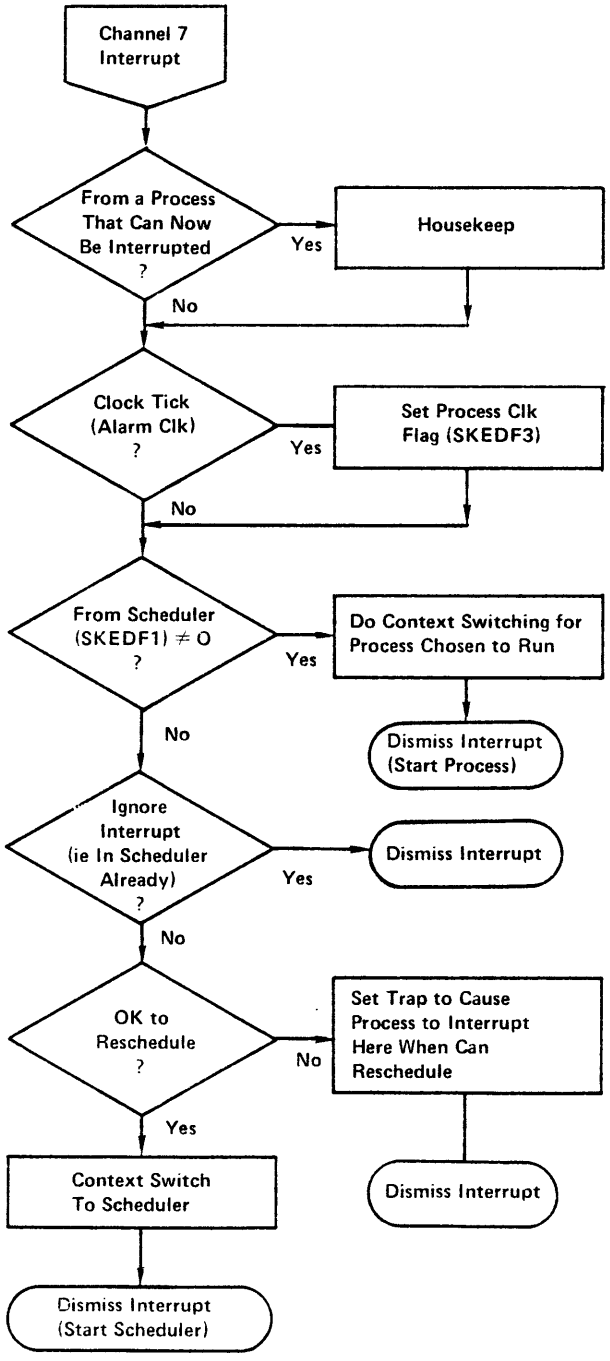
VI. Requesting DSK/MTA I/O & Interrupt Handling

VII. JSYS Calls - TTY Dependent Level

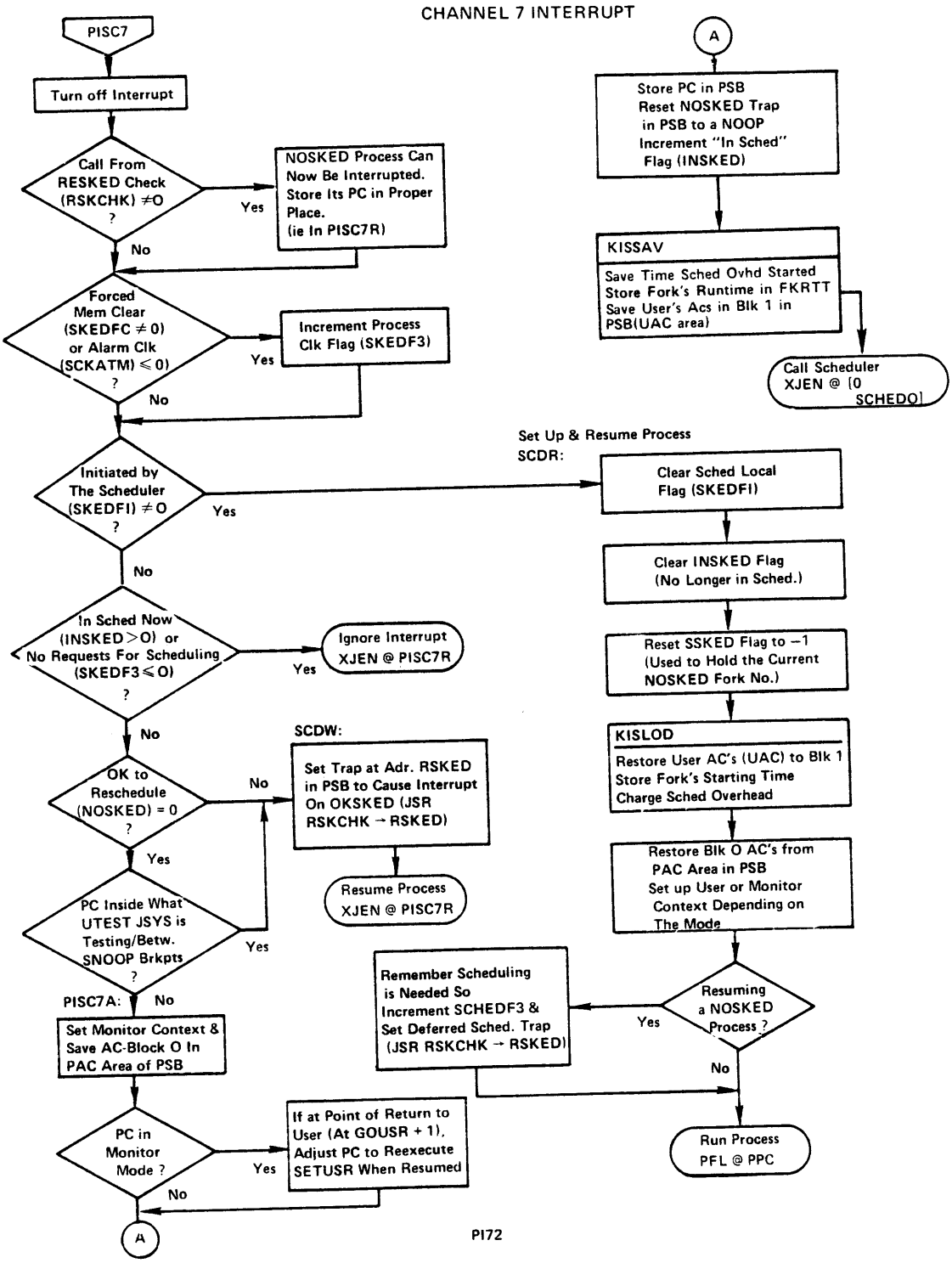
VIII. Requesting TTY I/O & Interrupt Handling

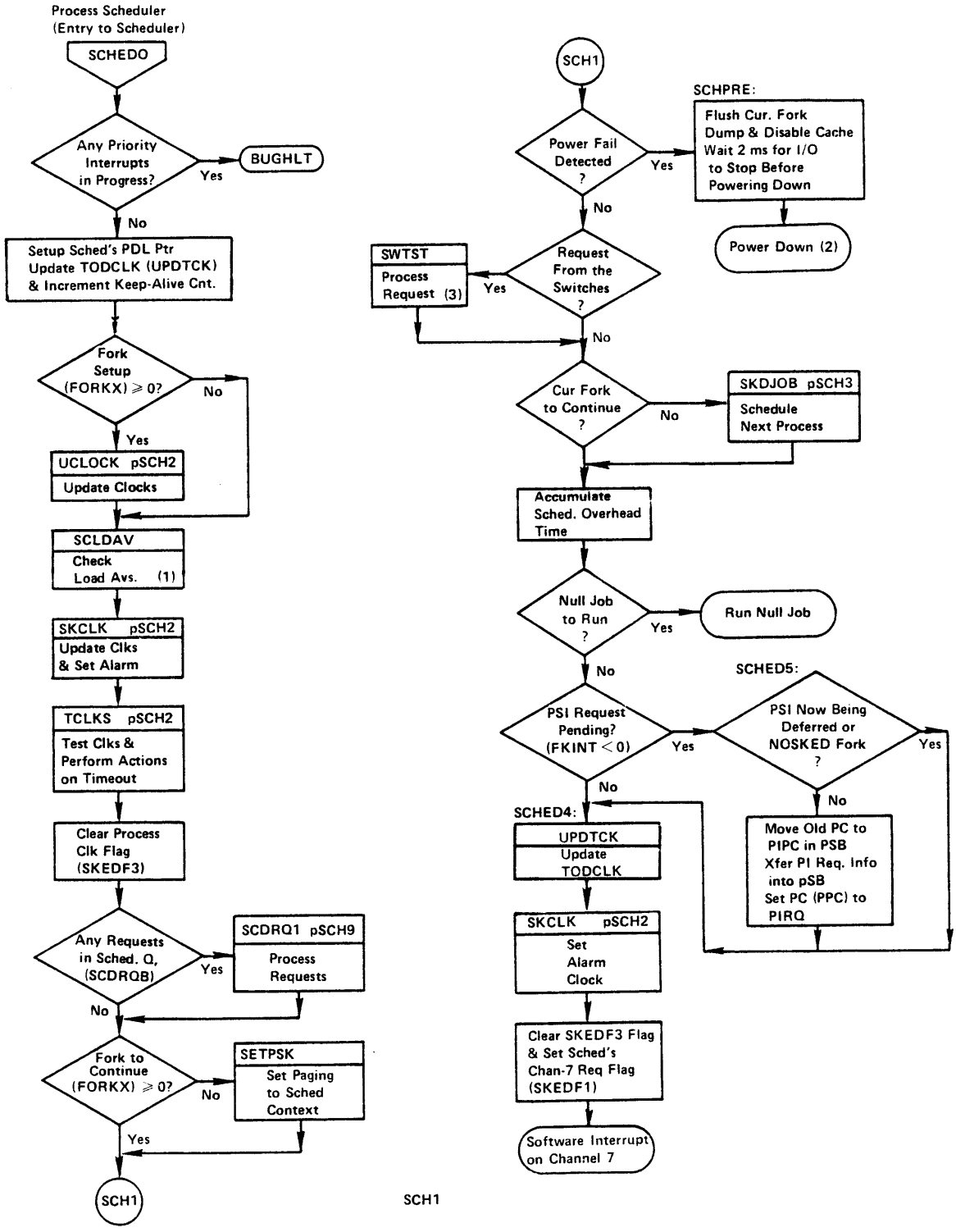
SCHEDULER FLOWCHARTS

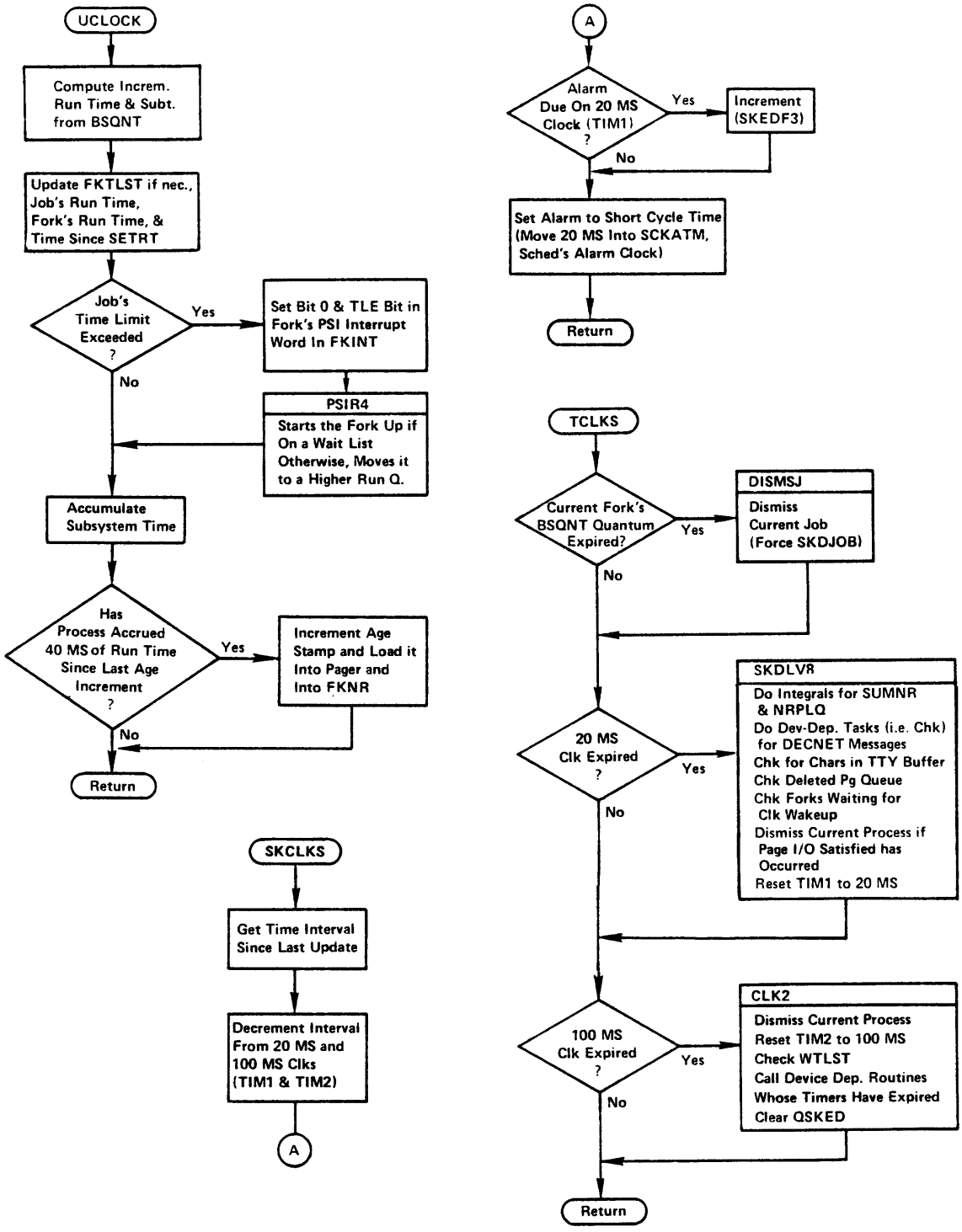
CHANNEL 7 INTERRUPT
AN OVERVIEW



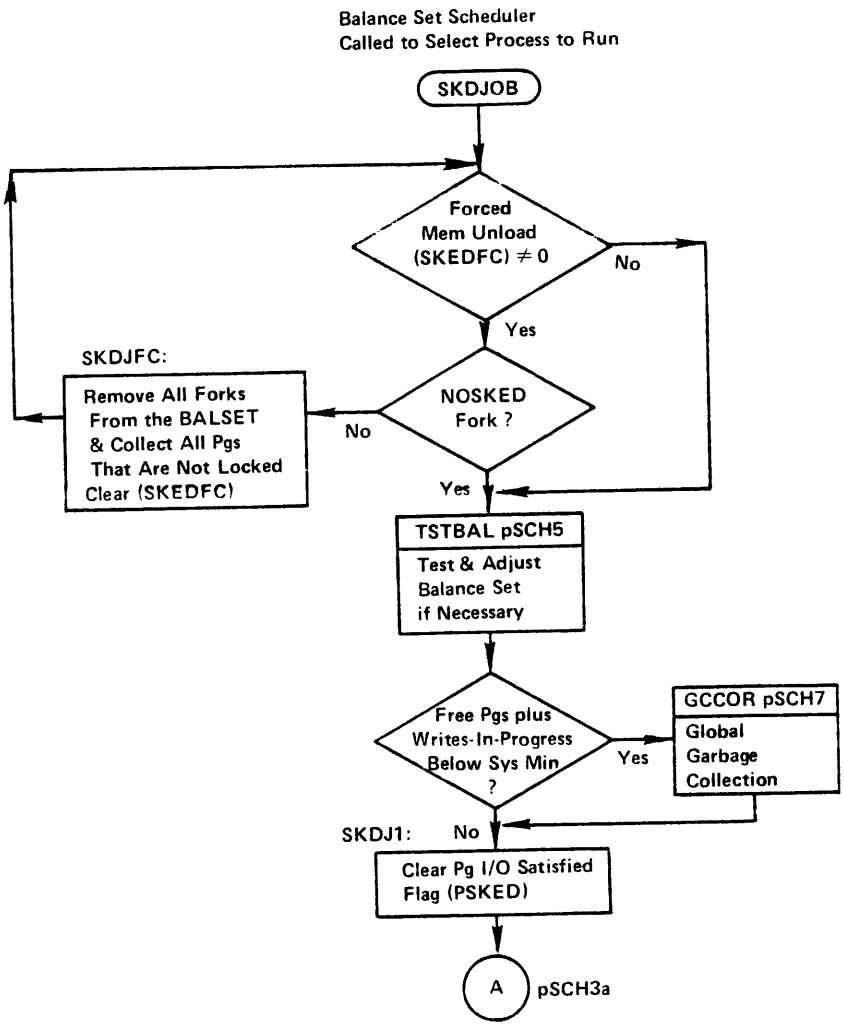
PI71


CHANNEL 7 INTERRUPT
PISC7
Turn off Interrupt
Call From RESKED Check (RSKCHK) ≠0 ?
Yes
NOSKED Process Can Now Be Interrupted. Store Its PC in Proper Place. (ie In PISC7R)
No
Forced Mem Clear (SKEDFC ≠ 0) or Alarm Clk (SCKATM) ≤ 0) ?
Yes
Increment Process Clk Flag (SKEDF3)
No
Initiated by The Scheduler (SKEDFI) ≠ 0 ?
Yes
No
In Sched Now (INSKED > 0) or No Requests For Scheduling (SKEDF3 ≤ 0) ?
Yes
Ignore Interrupt XJEN @ PISC7R
No
SCDW:
OK to Reschedule (NOSKED) = 0 ?
No
Set Trap at Adr. RSKED in PSB to Cause Interrupt On OKSKED (JSR RSKCHK → RSKED)
Resume Process XJEN @ PISC7R
Yes
PC Inside What UTEST JSYS is Testing/Betw. SNOOP Brkpts ?
Yes
PISC7A:
No
Set Monitor Context & Save AC-Block O In PAC Area of PSB
PC in Monitor Mode ?
Yes
If at Point of Return to User (At GOUSR + 1), Adjust PC to Reexecute SETUSR When Resumed
No
A
A
Store PC in PSB
Reset NOSKED Trap in PSB to a NOOP
Increment "In Sched" Flag (INSKED)
KISSAV
Save Time Sched Ovhd Started
Store Fork's Runtime in FKRTT
Save User's Acs in Blk 1 in PSB(UAC area)
Call Scheduler XJEN @ [0 SCHEDO]
Set Up & Resume Process
SCDR:
Clear Sched Local Flag (SKEDFI)
Clear INSKED Flag (No Longer in Sched.)
Reset SSKED Flag to −1 (Used to Hold the Current NOSKED Fork No.)
KISLOD
Restore User AC's (UAC) to Blk 1
Store Fork's Starting Time
Charge Sched Overhead
Restore Blk O AC's from PAC Area in PSB
Set up User or Monitor Context Depending on The Mode
Resuming a NOSKED Process ?
Yes
Remember Scheduling is Needed So Increment SCHEDF3 & Set Deferred Sched. Trap (JSR RSKCHK → RSKED)
No
Run Process PFL @ PPC
PI72

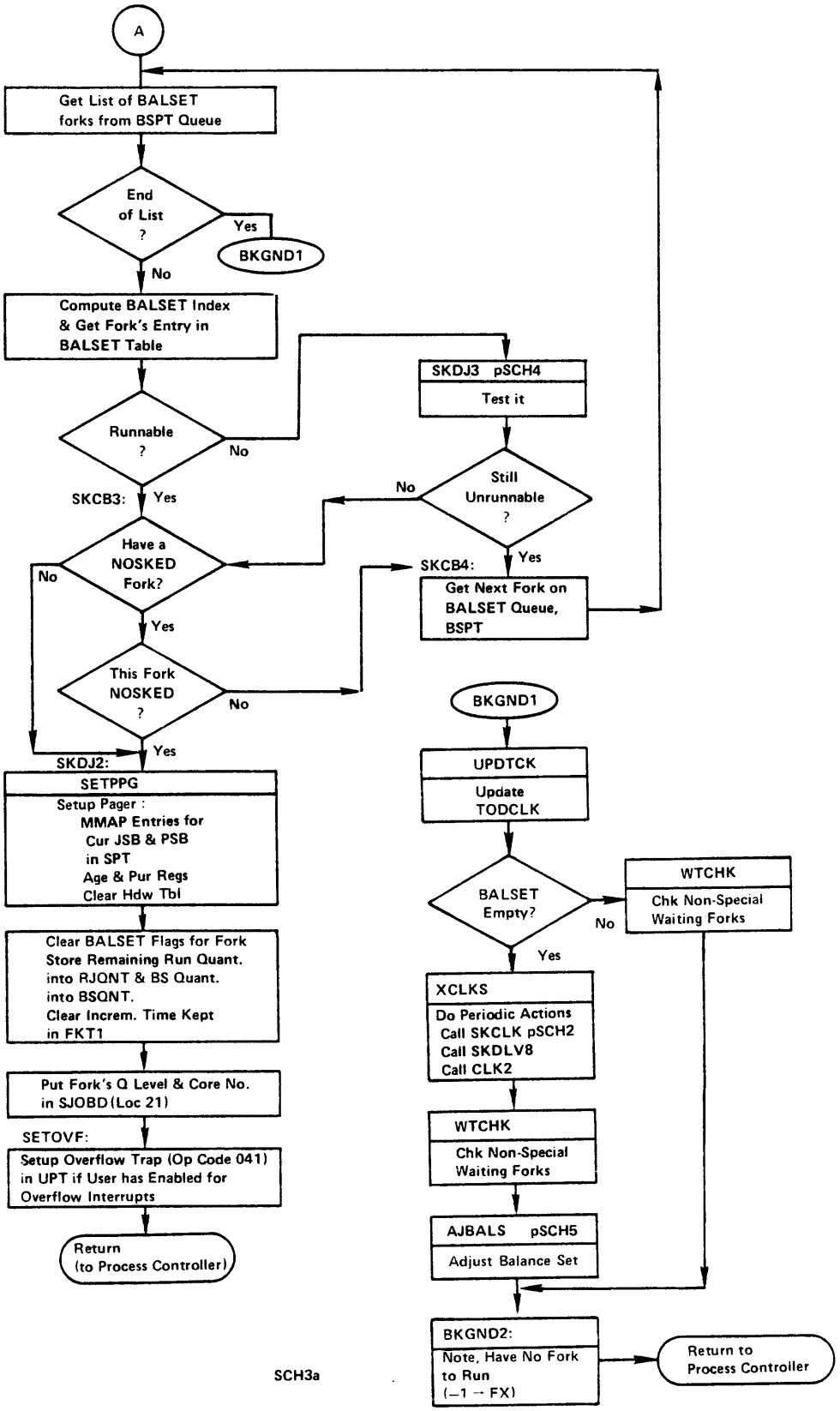

Process Scheduler
(Entry to Scheduler)
SCHEDO
Any Priority Interrupts in Progress?
Yes
BUGHLT
No
Setup Sched's PDL Ptr
Update TODCLK (UPDTCK)
& Increment Keep-Alive Cnt.
Fork Setup (FORKX) ≥ 0?
No
Yes
UCLOCK pSCH2
Update Clocks
SCLDAV
Check Load Avs. (1)
SKCLK pSCH2
Update Clks & Set Alarm
TCLKS pSCH2
Test Clks & Perform Actions on Timeout
Clear Process Clk Flag (SKEDF3)
Any Requests in Sched. Q, (SCDRQB)
Yes
SCDRQ1 pSCH9
Process Requests
No
Fork to Continue (FORKX) ≥ 0?
No
SETPSK
Set Paging to Sched Context
Yes
SCH1
SCH1
Power Fail Detected ?
Yes
SCHPRE:
Flush Cur. Fork
Dump & Disable Cache
Wait 2 ms for I/O
to Stop Before
Powering Down
Power Down (2)
No
Request From the Switches ?
Yes
SWTST
Process Request (3)
No
Cur Fork to Continue ?
No
SKDJOB pSCH3
Schedule Next Process
Accumulate Sched. Overhead Time
Null Job to Run ?
Yes
Run Null Job
No
SCHED5:
PSI Request Pending? (FKINT < 0)
Yes
PSI Now Being Deferred or NOSKED Fork ?
Yes
No
Move Old PC to PIPC in PSB
Xfer PI Req. Info into pSB
Set PC (PPC) to PIRQ
No
SCHED4:
UPDTCK
Update TODCLK
SKCLK pSCH2
Set Alarm Clock
Clear SKEDF3 Flag & Set Sched's Chan-7 Req Flag (SKEDF1)
Software Interrupt on Channel 7


UCLOCK
Compute Increm. Run Time & Subt. from BSQNT
Update FKTLST if nec., Job's Run Time, Fork's Run Time, & Time Since SETRT
Job's Time Limit Exceeded ?
Yes
No
Set Bit 0 & TLE Bit in Fork's PSI Interrupt Word In FKINT
PSIR4
Starts the Fork Up if On a Wait List Otherwise, Moves it to a Higher Run Q.
Accumulate Subsystem Time
Has Process Accrued 40 MS of Run Time Since Last Age Increment ?
Yes
No
Increment Age Stamp and Load it Into Pager and Into FKNR
Return
SKCLKS
Get Time Interval Since Last Update
Decrement Interval From 20 MS and 100 MS Clks (TIM1 & TIM2)
A
A
Alarm Due On 20 MS Clock (TIM1) ?
Yes
No
Increment (SKEDF3)
Set Alarm to Short Cycle Time (Move 20 MS Into SCKATM, Sched's Alarm Clock)
Return
TCLKS
Current Fork's BSQNT Quantum Expired?
Yes
No
DISMSJ
Dismiss Current Job (Force SKDJOB)
20 MS Clk Expired ?
Yes
No
SKDLVR
Do Integrals for SUMNR & NRPLQ
Do Dev-Dep. Tasks (i.e. Chk) for DECNET Messages
Chk for Chars in TTY Buffer
Chk Deleted Pg Queue
Chk Forks Waiting for Clk Wakeup
Dismiss Current Process if Page I/O Satisfied has Occurred
Reset TIM1 to 20 MS
100 MS Clk Expired ?
Yes
No
CLK2
Dismiss Current Process
Reset TIM2 to 100 MS
Check WTLST
Call Device Dep. Routines Whose Timers Have Expired
Clear QSKED
Return

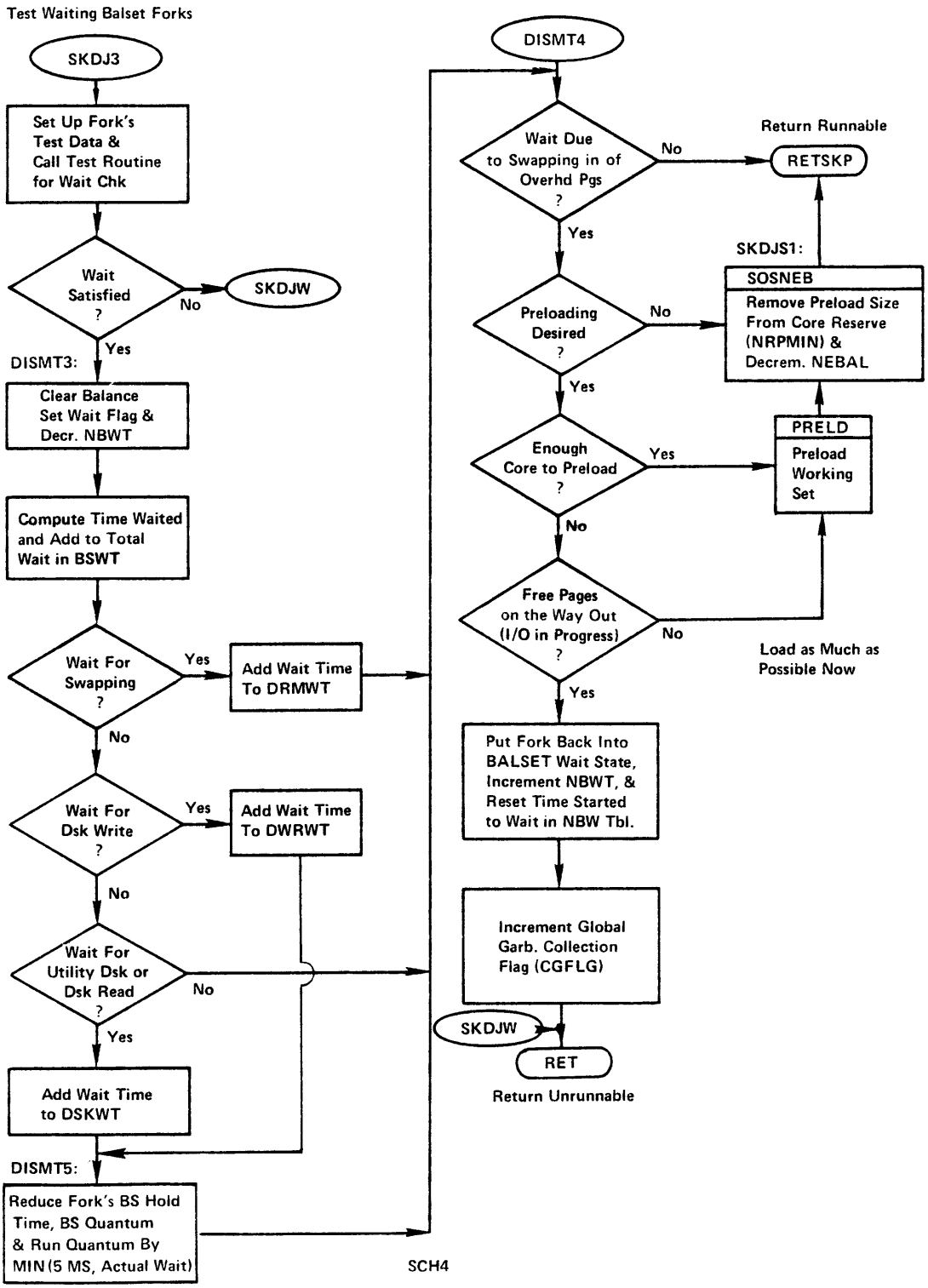
SCH2

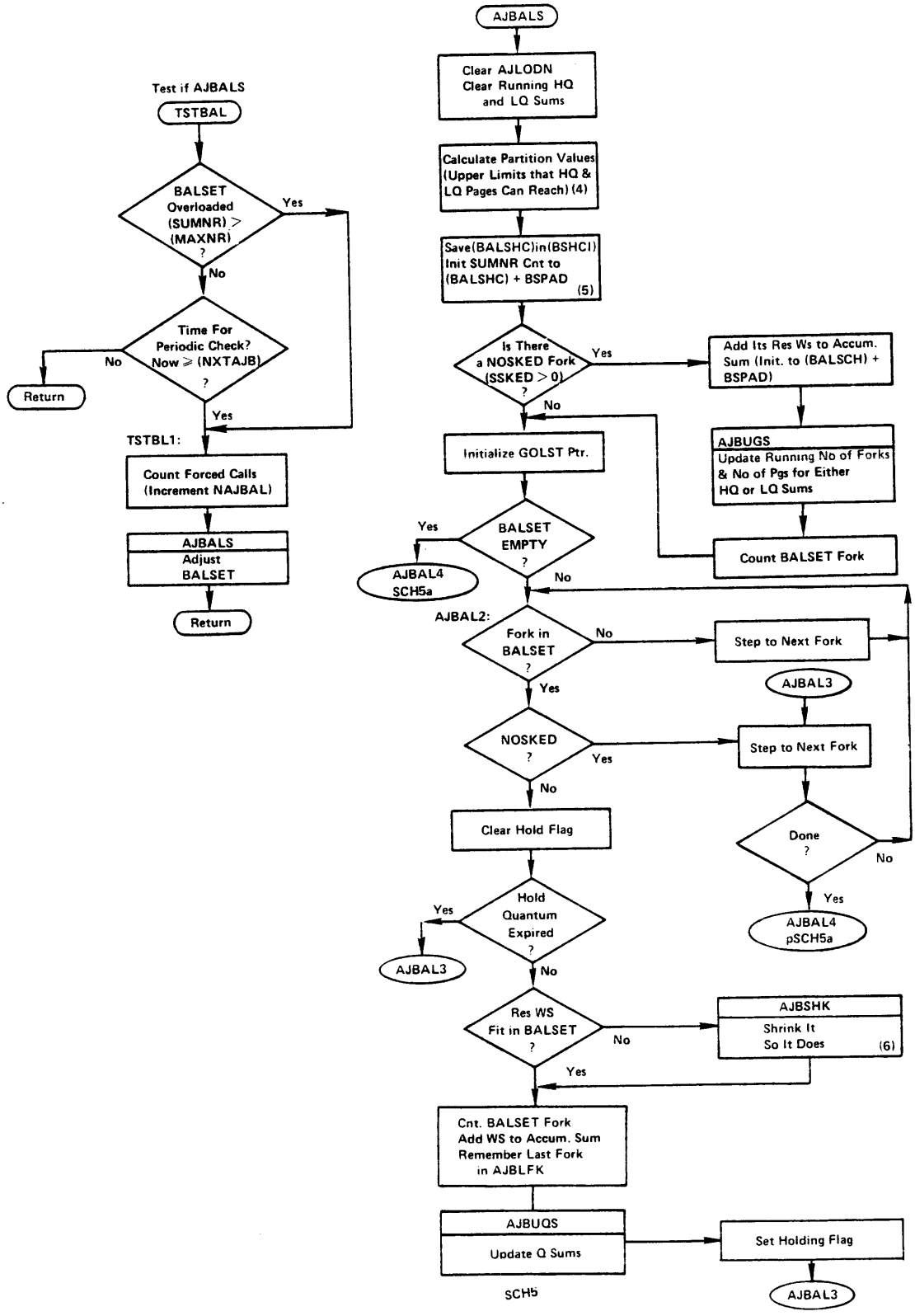

Balance Set Scheduler
Called to Select Process to Run
SKDJOB
Forced
Mem Unload
(SKEDFC) ≠ 0
No
Yes
SKDJFC:
Remove All Forks
From the BALSET
& Collect All Pgs
That Are Not Locked
Clear (SKEDFC)
NOSKED
Fork ?
No
Yes
TSTBAL pSCH5
Test & Adjust
Balance Set
if Necessary
Free Pgs plus
Writes-In-Progress
Below Sys Min
?
Yes
GCCOR pSCH7
Global
Garbage
Collection
SKDJ1:
No
Clear Pg I/O Satisfied
Flag (PSKED)
A
pSCH3a

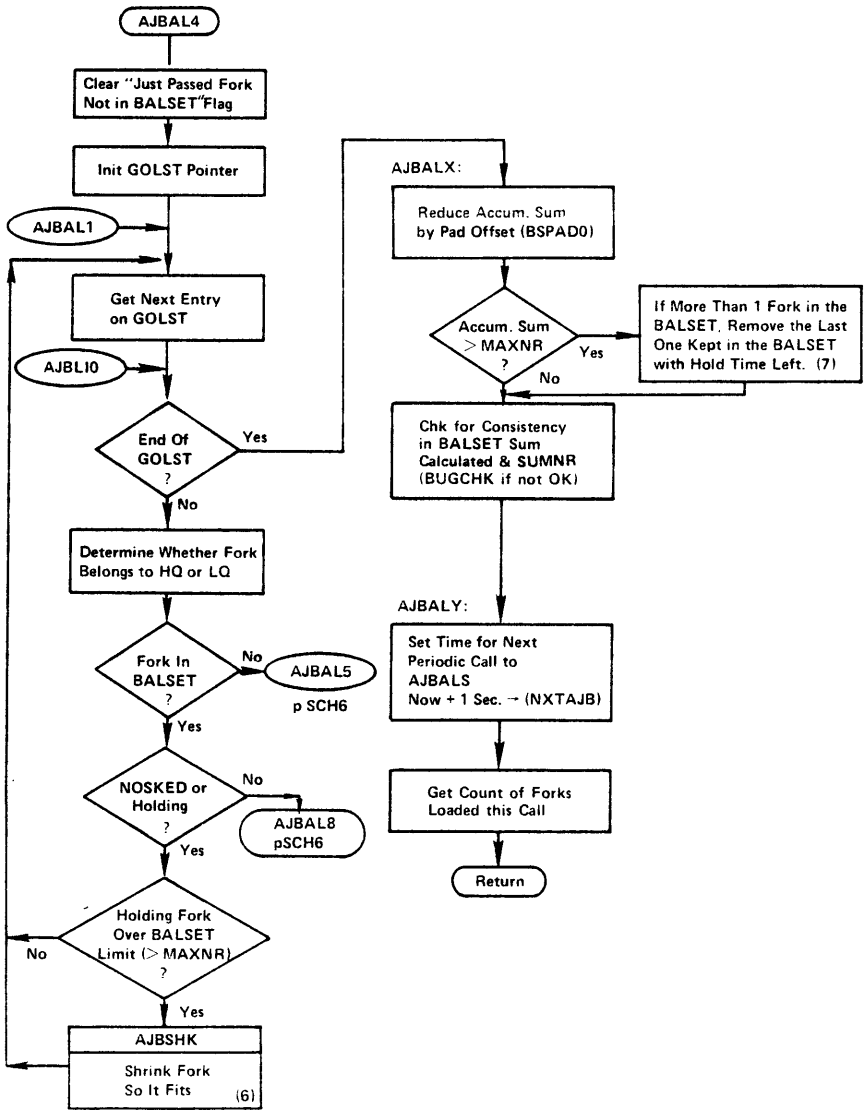
SCH3


A
Get List of BALSET forks from BSPT Queue
End of List ?
Yes
BKGND1
No
Compute BALSET Index & Get Fork's Entry in BALSET Table
Runnable ?
No
SKCB3:
Yes
Have a NOSKED Fork?
No
Yes
This Fork NOSKED ?
No
Yes
SKDJ2:
SETPPG
Setup Pager :
MMAP Entries for
Cur JSB & PSB
in SPT
Age & Pur Regs
Clear Hdw Tbl
Clear BALSET Flags for Fork
Store Remaining Run Quant.
into RJQNT & BS Quant.
into BSQNT.
Clear Increm. Time Kept
in FKT1
Put Fork's Q Level & Core No.
in SJOBD(Loc 21)
SETOVF:
Setup Overflow Trap (Op Code 041)
in UPT if User has Enabled for
Overflow Interrupts
Return
(to Process Controller)
SKDJ3 pSCH4
Test it
Still Unrunnable ?
No
Yes
SKCB4:
Get Next Fork on BALSET Queue, BSPT
BKGND1
UPDTCK
Update TODCLK
BALSET Empty?
No
Yes
WTCHK
Chk Non-Special Waiting Forks
XCLKS
Do Periodic Actions
Call SKCLK pSCH2
Call SKDLV8
Call CLK2
WTCHK
Chk Non-Special Waiting Forks
AJBALS pSCH5
Adjust Balance Set
BKGND2:
Note, Have No Fork to Run
(–1 → FX)
Return to Process Controller

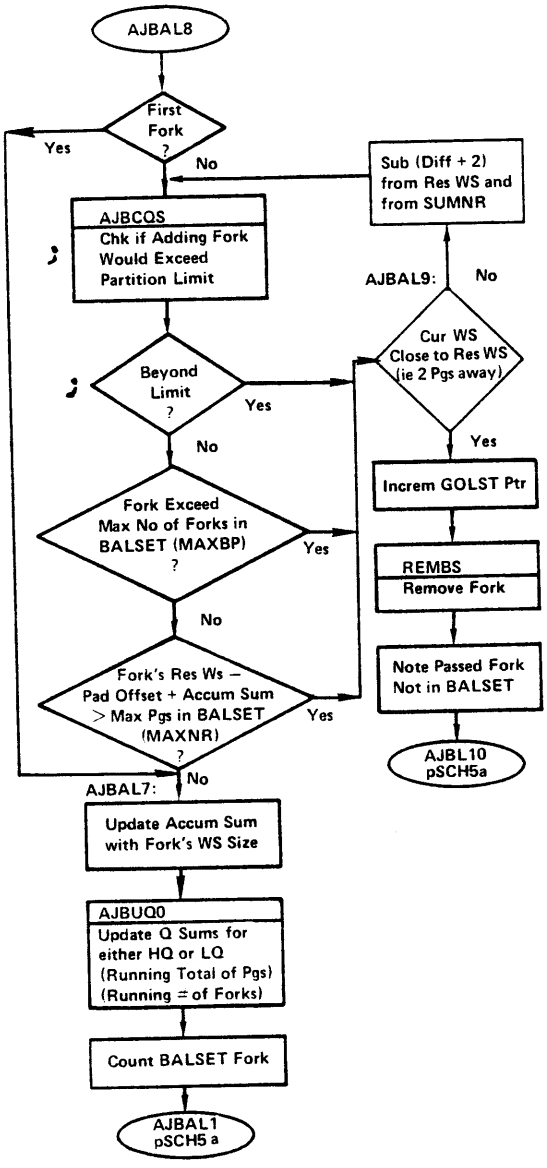
SCH3a

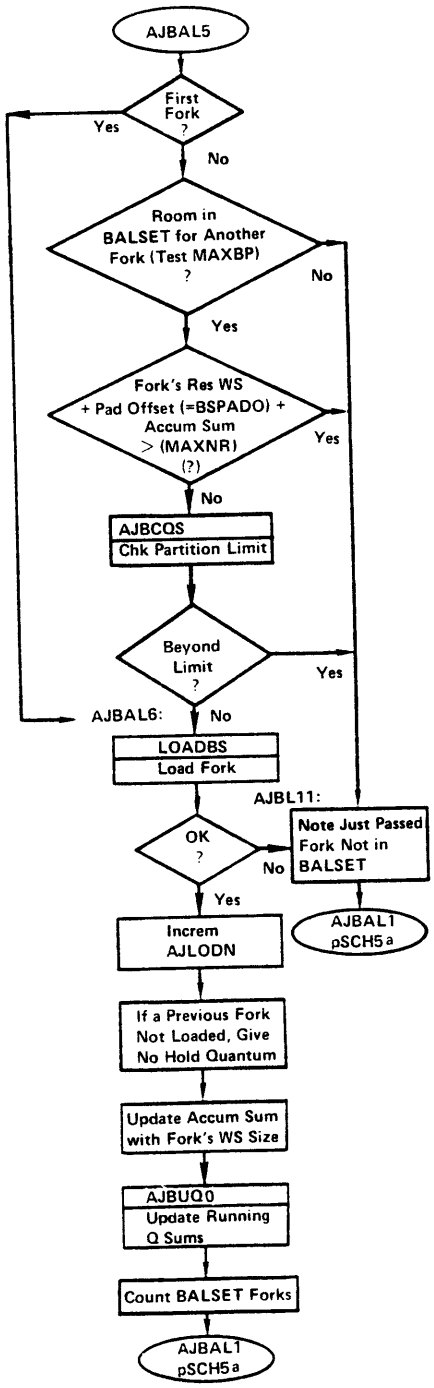

Test Waiting Balset Forks
SKDJ3
Set Up Fork's Test Data & Call Test Routine for Wait Chk
Wait Satisfied ?
No
SKDJW
Yes
DISMT3:
Clear Balance Set Wait Flag & Decr. NBWT
Compute Time Waited and Add to Total Wait in BSWT
Wait For Swapping ?
Yes
Add Wait Time To DRMWT
No
Wait For Dsk Write ?
Yes
Add Wait Time To DWRWT
No
Wait For Utility Dsk or Dsk Read ?
No
Yes
Add Wait Time to DSKWT
DISMT5:
Reduce Fork's BS Hold Time, BS Quantum & Run Quantum By MIN(5 MS, Actual Wait)
SCH4
DISMT4
Wait Due to Swapping in of Overhd Pgs ?
No
Return Runnable
RETSKP
Yes
SKDJS1:
SOSNEB
Remove Preload Size From Core Reserve (NRPMIN) & Decrem. NEBAL
Preloading Desired ?
No
Yes
PRELD
Preload Working Set
Enough Core to Preload ?
Yes
No
Free Pages on the Way Out (I/O in Progress) ?
No
Load as Much as Possible Now
Yes
Put Fork Back Into BALSET Wait State, Increment NBWT, & Reset Time Started to Wait in NBW Tbl.
Increment Global Garb. Collection Flag (CGFLG)
SKDJW
RET
Return Unrunnable


Test if AJBALS
TSTBAL
BALSET Overloaded (SUMNR) > (MAXNR) ?
Yes
No
Time For Periodic Check? Now ≥ (NXTAJB) ?
No
Return
Yes
TSTBL1:
Count Forced Calls (Increment NAJBAL)
AJBALS
Adjust BALSET
Return
AJBALS
Clear AJLODN
Clear Running HQ and LQ Sums
Calculate Partition Values (Upper Limits that HQ & LQ Pages Can Reach) (4)
Save(BALSHC)in(BSHCI)
Init SUMNR Cnt to (BALSHC) + BSPAD (5)
Is There a NOSKED Fork (SSKED > 0) ?
Yes
Add Its Res Ws to Accum. Sum (Init. to (BALSCH) + BSPAD)
No
AJBUGS
Update Running No of Forks & No of Pgs for Either HQ or LQ Sums
Initialize GOLST Ptr.
Count BALSET Fork
BALSET EMPTY ?
Yes
AJBAL4 SCH5a
No
AJBAL2:
Fork in BALSET ?
No
Step to Next Fork
AJBAL3
Yes
NOSKED ?
Yes
Step to Next Fork
No
Clear Hold Flag
Done ?
No
Yes
AJBAL4 pSCH5a
Hold Quantum Expired ?
Yes
AJBAL3
No
Res WS Fit in BALSET ?
No
AJBSHK
Shrink It So It Does (6)
Yes
Cnt. BALSET Fork
Add WS to Accum. Sum
Remember Last Fork in AJBLFK
AJBUQS
Update Q Sums
Set Holding Flag
AJBAL3
SCH5

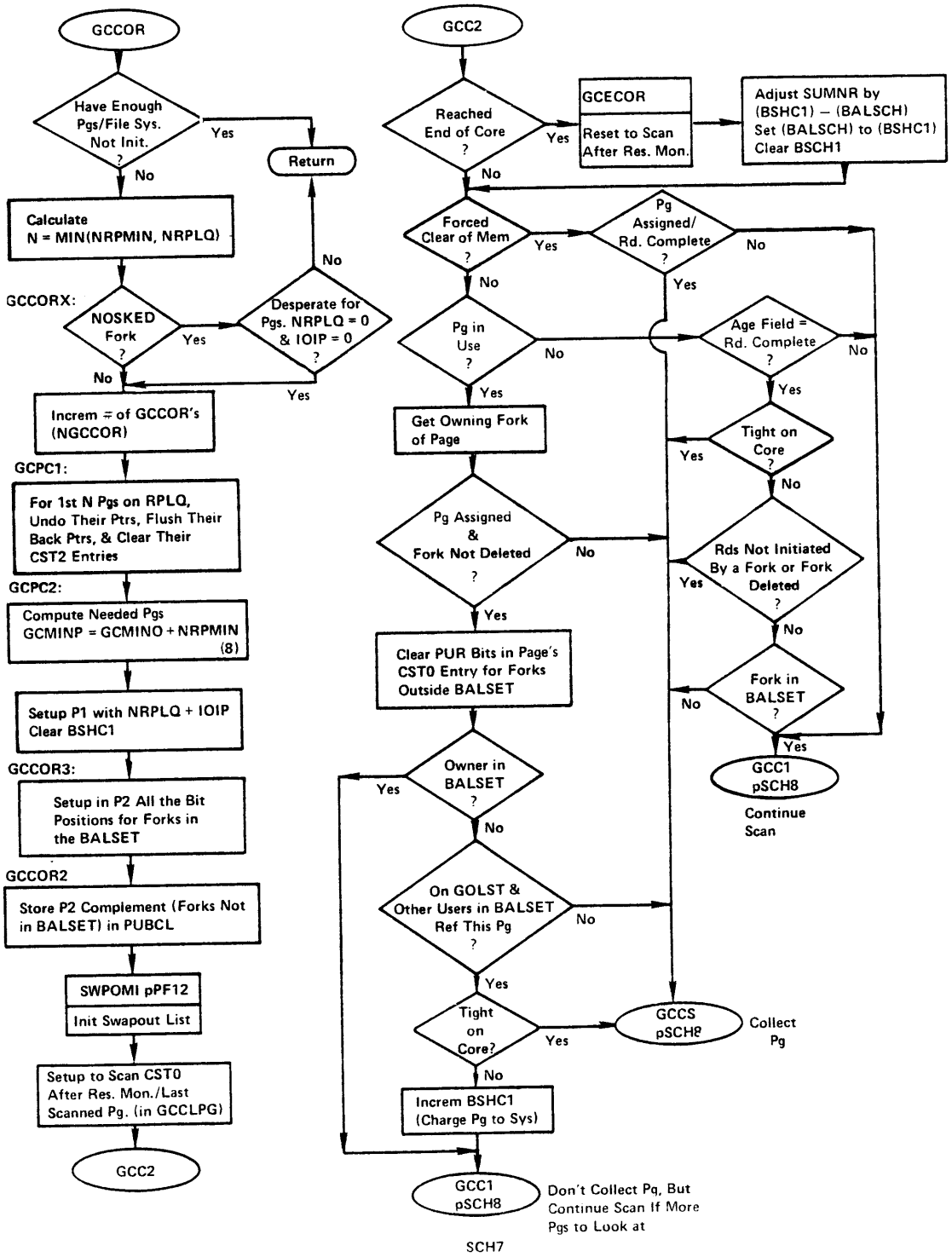

AJBAL4
Clear "Just Passed Fork Not in BALSET" Flag
Init GOLST Pointer
AJBAL1
Get Next Entry on GOLST
AJBL10
End Of GOLST ?
Yes
No
Determine Whether Fork Belongs to HQ or LQ
Fork In BALSET ?
No
AJBAL5
p SCH6
Yes
NOSKED or Holding ?
No
AJBAL8
pSCH6
Yes
Holding Fork Over BALSET Limit (> MAXNR) ?
No
Yes
AJBSHK
Shrink Fork So It Fits (6)
AJBALX:
Reduce Accum. Sum by Pad Offset (BSPAD0)
Accum. Sum > MAXNR ?
Yes
If More Than 1 Fork in the BALSET, Remove the Last One Kept in the BALSET with Hold Time Left. (7)
No
Chk for Consistency in BALSET Sum Calculated & SUMNR (BUGCHK if not OK)
AJBALY:
Set Time for Next Periodic Call to AJBALS Now + 1 Sec. → (NXTAJB)
Get Count of Forks Loaded this Call
Return

SCH5a

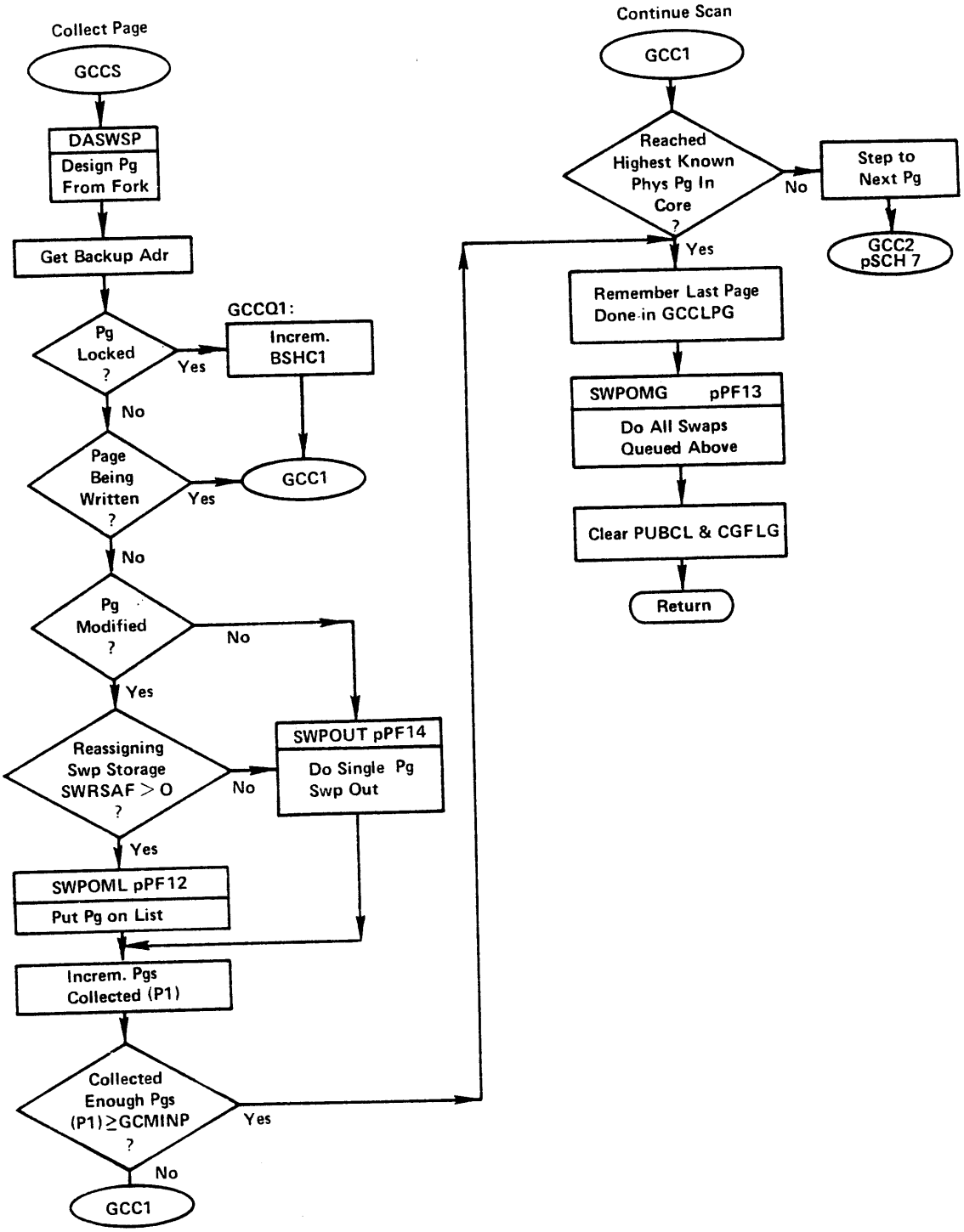

AJBAL8
First Fork ?
Yes
No
AJBCQS
Chk if Adding Fork Would Exceed Partition Limit
Beyond Limit ?
Yes
No
Fork Exceed Max No of Forks in BALSET (MAXBP) ?
Yes
No
Fork's Res Ws – Pad Offset + Accum Sum > Max Pgs in BALSET (MAXNR) ?
Yes
No
AJBAL7:
Update Accum Sum with Fork's WS Size
AJBUQ0
Update Q Sums for either HQ or LQ (Running Total of Pgs) (Running # of Forks)
Count BALSET Fork
AJBAL1 pSCH5a
Sub (Diff + 2) from Res WS and from SUMNR
AJBAL9:
No
Cur WS Close to Res WS (ie 2 Pgs away)
Yes
Increm GOLST Ptr
REMBS
Remove Fork
Note Passed Fork Not in BALSET
AJBL10 pSCH5a


AJBAL5
First Fork ?
Yes
No
Room in BALSET for Another Fork (Test MAXBP) ?
No
Yes
Fork's Res WS + Pad Offset (=BSPADO) + Accum Sum > (MAXNR) (?)
Yes
No
AJBCQS
Chk Partition Limit
Beyond Limit ?
Yes
AJBAL6:
No
LOADBS
Load Fork
AJBL11:
OK ?
No
Note Just Passed Fork Not in BALSET
AJBAL1 pSCH5a
Yes
Increm AJLODN
If a Previous Fork Not Loaded, Give No Hold Quantum
Update Accum Sum with Fork's WS Size
AJBUQ0
Update Running Q Sums
Count BALSET Forks
AJBAL1 pSCH5a

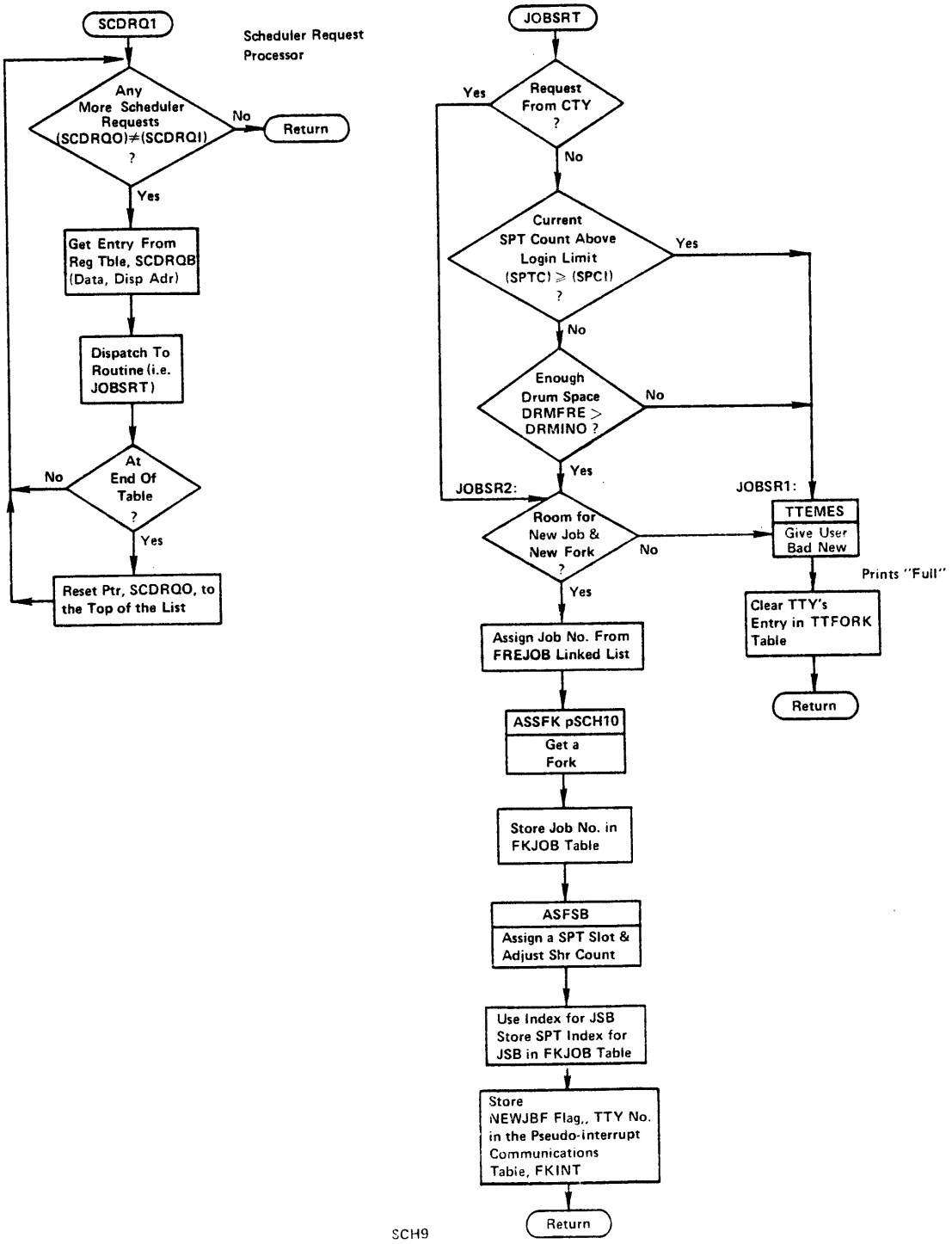
SCH6


GCCOR
Have Enough Pgs/File Sys. Not Init. ?
Yes
Return
No
Calculate N = MIN(NRPMIN, NRPLQ)
GCCORX:
NOSKED Fork ?
Yes
Desperate for Pgs. NRPLQ = 0 & IOIP = 0 ?
No
Yes
No
Increm # of GCCOR's (NGCCOR)
GCPC1:
For 1st N Pgs on RPLQ, Undo Their Ptrs, Flush Their Back Ptrs, & Clear Their CST2 Entries
GCPC2:
Compute Needed Pgs GCMINP = GCMINO + NRPMIN (8)
Setup P1 with NRPLQ + IOIP Clear BSHC1
GCCOR3:
Setup in P2 All the Bit Positions for Forks in the BALSET
GCCOR2
Store P2 Complement (Forks Not in BALSET) in PUBCL
SWPOMI pPF12
Init Swapout List
Setup to Scan CST0 After Res. Mon./Last Scanned Pg. (in GCCLPG)
GCC2
GCC2
Reached End of Core ?
Yes
GCECOR
Reset to Scan After Res. Mon.
Adjust SUMNR by (BSHC1) – (BALSCH) Set (BALSCH) to (BSHC1) Clear BSCH1
No
Forced Clear of Mem ?
Yes
Pg Assigned/ Rd. Complete ?
No
Yes
No
Pg in Use ?
No
Age Field = Rd. Complete ?
No
Yes
Yes
Get Owning Fork of Page
Tight on Core ?
Yes
No
Pg Assigned & Fork Not Deleted ?
No
Rds Not Initiated By a Fork or Fork Deleted ?
Yes
Yes
No
Clear PUR Bits in Page's CST0 Entry for Forks Outside BALSET
Fork in BALSET ?
No
Yes
GCC1 pSCH8
Continue Scan
Owner in BALSET ?
Yes
No
On GOLST & Other Users in BALSET Ref This Pg ?
No
Yes
Tight on Core?
Yes
GCCS pSCH8
Collect Pg
No
Increm BSHC1 (Charge Pg to Sys)
GCC1 pSCH8
Don't Collect Pg, But Continue Scan If More Pgs to Look at

SCH7

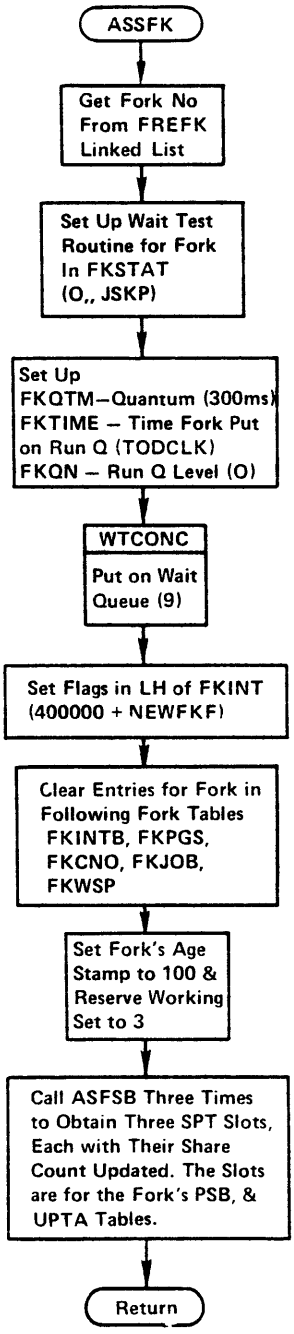

Collect Page
GCCS
DASWSP
Design Pg From Fork
Get Backup Adr
GCCQ1:
Pg Locked ?
Yes
Increm. BSHC1
No
Page Being Written ?
Yes
GCC1
No
Pg Modified ?
No
Yes
Reassigning Swp Storage SWRSAF > 0 ?
No
SWPOUT pPF14
Do Single Pg Swp Out
Yes
SWPOML pPF12
Put Pg on List
Increm. Pgs Collected (P1)
Collected Enough Pgs (P1)≥GCMINP ?
Yes
No
GCC1
Continue Scan
GCC1
Reached Highest Known Phys Pg In Core ?
No
Step to Next Pg
GCC2 pSCH 7
Yes
Remember Last Page Done in GCCLPG
SWPOMG pPF13
Do All Swaps Queued Above
Clear PUBCL & CGFLG
Return

SCH8


SCDRQ1
Scheduler Request Processor
Any More Scheduler Requests (SCDRQO)≠(SCDRQI) ?
No
Return
Yes
Get Entry From Reg Tble, SCDRQB (Data, Disp Adr)
Dispatch To Routine (i.e. JOBSRT)
At End Of Table ?
No
Yes
Reset Ptr, SCDRQO, to the Top of the List
JOBSRT
Request From CTY ?
Yes
No
Current SPT Count Above Login Limit (SPTC) ≥ (SPCI) ?
Yes
No
Enough Drum Space DRMFRE > DRMINO ?
No
Yes
JOBSR2:
Room for New Job & New Fork ?
No
Yes
JOBSR1:
TTEMES
Give User Bad New
Prints "Full"
Clear TTY's Entry in TTFORK Table
Return
Assign Job No. From FREJOB Linked List
ASSFK pSCH10
Get a Fork
Store Job No. in FKJOB Table
ASFSB
Assign a SPT Slot & Adjust Shr Count
Use Index for JSB Store SPT Index for JSB in FKJOB Table
Store NEWJBF Flag,, TTY No. in the Pseudo-interrupt Communications Tablie, FKINT
Return

SCH9

Assign Fork Slot



SCHIO

Scheduler Comments

SCHEDO:

(1) Running averages, exponentially weighed over intervals of 1, 5, and 15 minutes,are maintained for the number of runnable processes overall, as well as for those in High Run Queues and those in the Low Run Queues.

(2) Final phase of powerdown seq. clears the priority interrupt system and causes the system to loop in the ACs until power actually vanishes. If the power fail interrupt was spurious, the loop will time out after a few seconds and the system will be continued at address SYSRST.

(3) A very limited set of central functions for debugging purposes has been built into the Scheduler. To invoke a function, the appropriate bit or bits are set into loc 20 (SCTLW) via MDDT. The word is scanned from left to right (JFFO); the first bit found set on the scan selects the function.

Bit 0 Causes the scheduler to dismiss the current process and to stop timesharing. Useful to effect a clean manual transfer to Exec-mode DDT. System may be resumed at SCHEDO if no IOB reset is done.

Bit 1 Causes job specified by $(20)_{RH}$ to be run exclusively.

Bit 2 Forces running of Job 0 back-up function before halting the system.

If loc 30 (SHLTW) is set not equal to Ø, the system will crash. (Same as setting bit 2 of SCTLW word.)

AJBALS

(4) Upper Limit for LQ=MAXNR-MIN [Max HQ Reserve, HQ Load Avg.* (16)]

Upper Limit for HQ=MAXNR-MIN [Max LQ Reserve, No. of LQ forks * (32)]

(5) SUMNR reflects the number of timesharing pages in use. Its value after AJBALS equals the number of pages reserved for balance set members plus BALSHC (the number of pages shared, but not owned, by balance set members plus the number of locked pages).

BSPAD reflects the number of pages set aside for balance set members as their working set reserves grow. The real value of BSPAD is offset by a factor of BSPADO. When forks are trying to stay in the balance set, the adjustment algorithm allows the pad offset to be subtracted from the accumulated sum before it checks if the fork can fit.

$$\text{i.e.,} \quad \left(\text{BSPAD} + \sum_{i=1}^{n} \text{Res. WS}_i\right) - \text{BSPADO} + \text{Res. WS}_{n+1} \geq \text{MAXNR}$$

The adjustment algorithm does the opposite (i.e., adds the BSPADO factor) for forks trying to get into the balance set. The overall affect of this is to ensure (as much as possible) a certain number of pages be available for balance set forks.

(6) The shrink algorithm shrinks the fork's reserve working set by:

MIN [Reserve WS - Current WS, Accum. Sum + Fork's Res WS-MAXNR]

Notice that the fork's reserve working set will not be reduced below its current working set.

(7) This is the rare case of forks, with hold-time left, expanding. The lowest priority one is removed. If there is only one fork in the balance set, it is not removed. (Note: it is possible for one fork to be greater than MAXNR due to the BALSHC count changing).

GCCOR

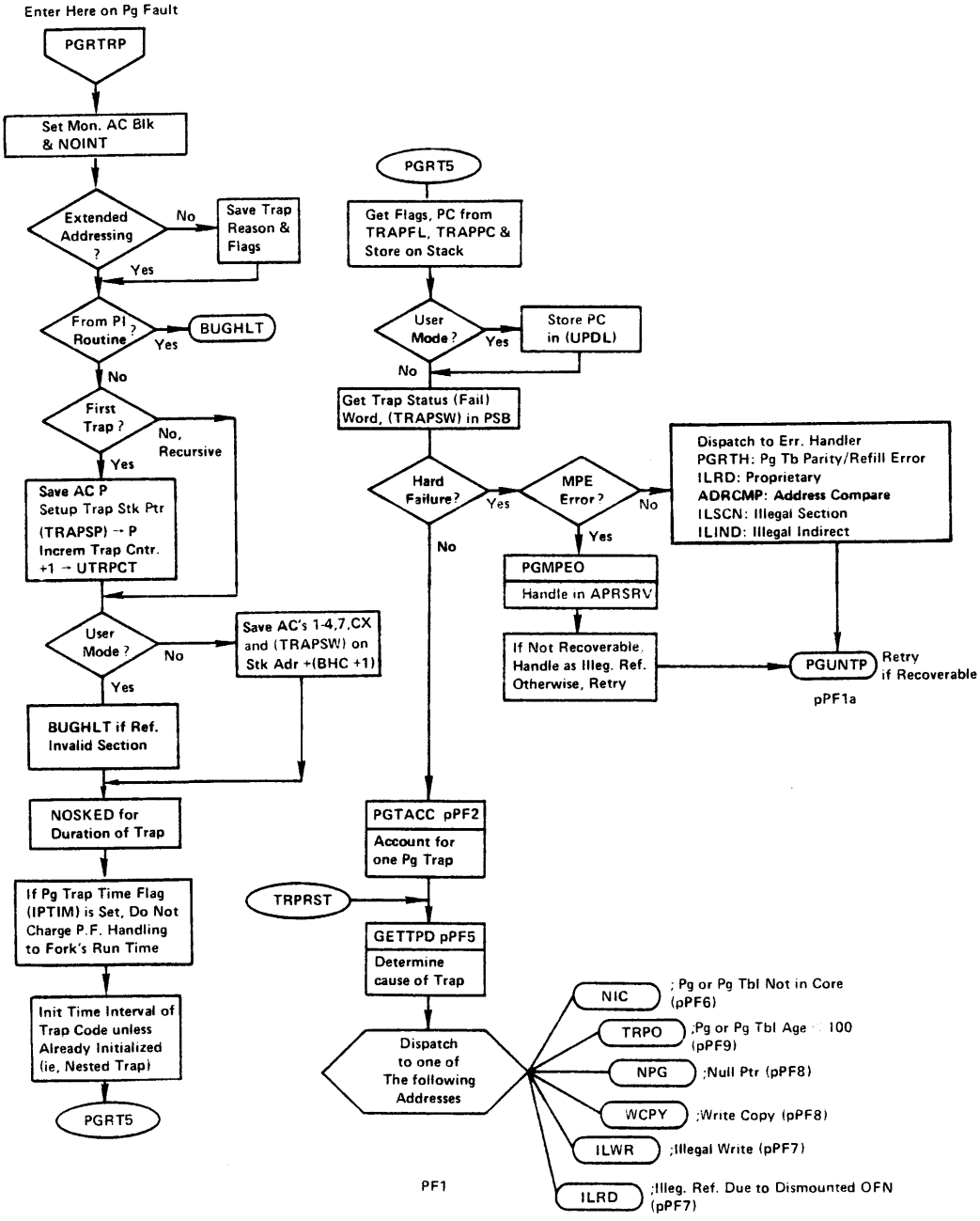
(8) If it is a forced clear, then GCMINO is made very large so all of core will be collected. However, its usual value is much lower. (Currently 64 decimal).

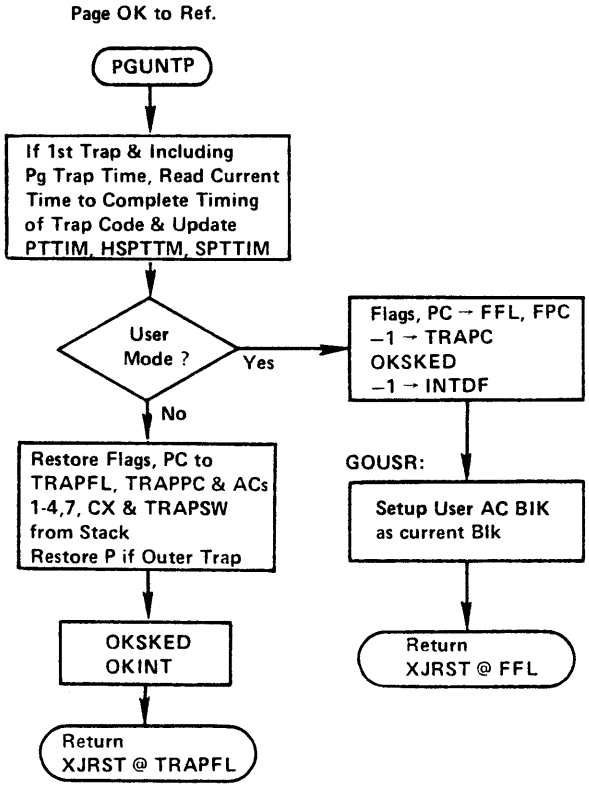
ASSFK

(9) The fork is actually placed on the GOLST at this time. WTCONC, after putting a fork on WTLST, checks if the wait condition is satisfied. The test routine, JSKP, gives a skip return indicating that the wait is satisfied. This causes UNBLK1 to be called which in turn calls SCHEDJ to unblock the fork and to requeue it from the WTLST to the GOLST.

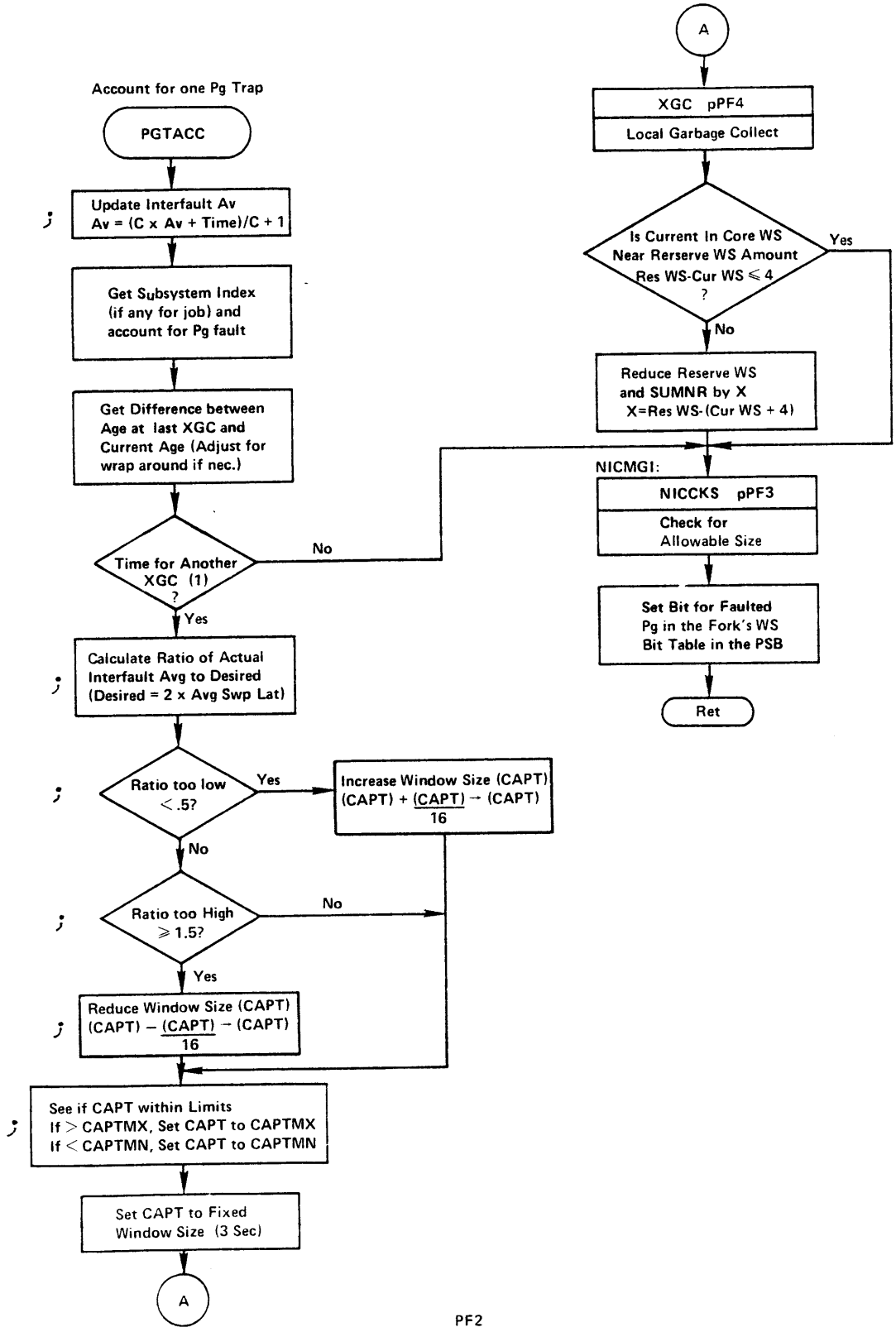
PAGE FAULT HANDLING FLOWCHARTS

PGRTRP -	Performs the Principal Accounting, Analysis, and Resolution of Page Faults		PF1
	PGTACC -	Accounts for Page Traps	PF2
		XGC - Local Garbage Collection	PF4
		SWPOUT - Swapping Out a Page	PF14
		NICCKS - Check In-Core Size Limits	PF3
	GETTPD -	Determine Cause of Trap	PF5
	NIC -	Not in Core Trap	PF6
		SWPINW - Swap In and Wait	PF10
		SWPIN - Swap In a Page	PF11
	WCPY -	Write Copy Trap	PF8
	ILRD -	Illegal Read Trap	PF7
	ILWR -	Illegal Write Trap	PF7
	TRPO -	Age < 100 Trap	PF9

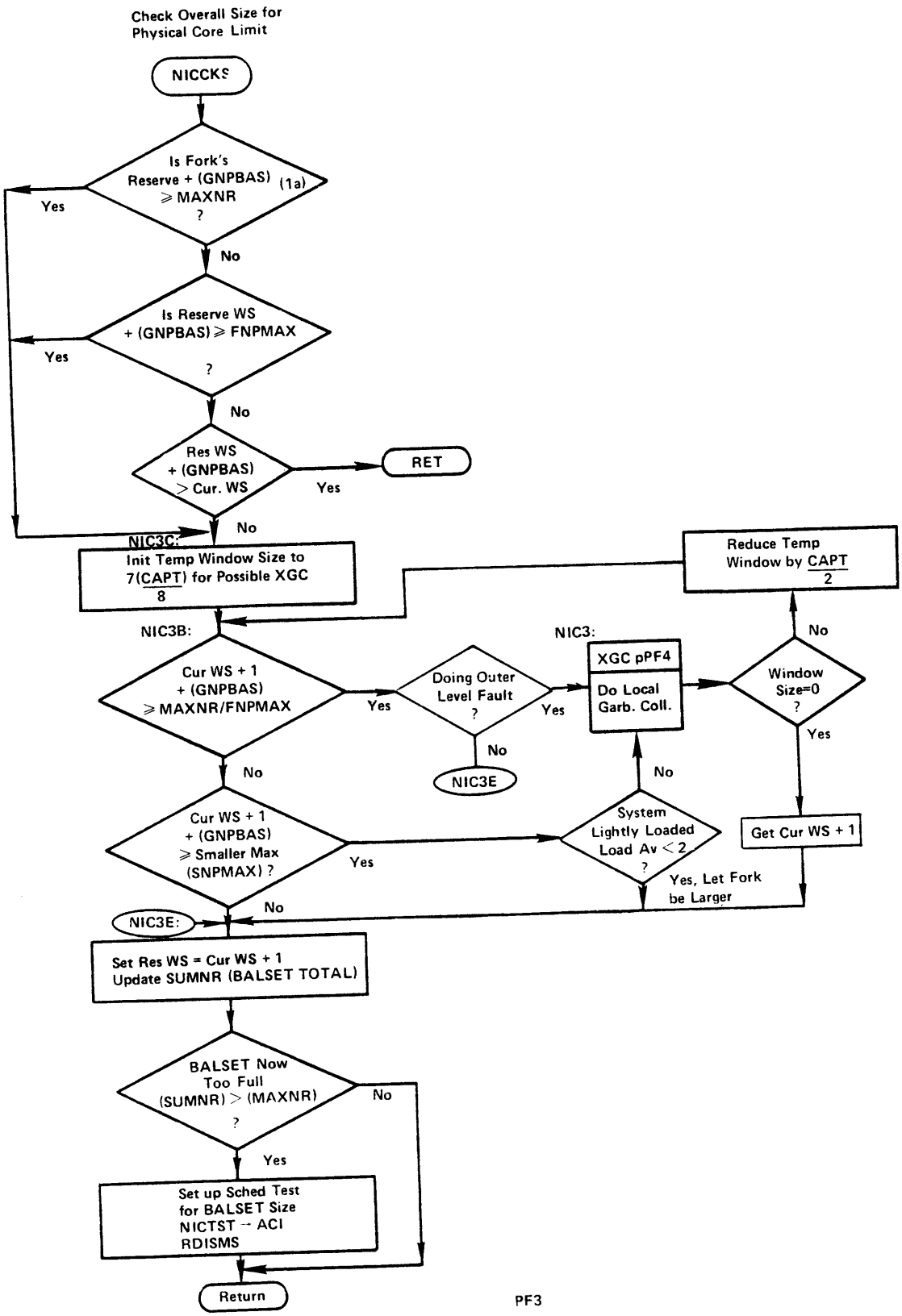

Enter Here on Pg Fault
PGRTRP
Set Mon. AC Blk & NOINT
Extended Addressing ?
No
Save Trap Reason & Flags
Yes
From PI Routine ?
Yes
BUGHLT
No
First Trap ?
No, Recursive
Yes
Save AC P
Setup Trap Stk Ptr
(TRAPSP) → P
Increm Trap Cntr.
+1 → UTRPCT
User Mode ?
No
Save AC's 1-4,7,CX and (TRAPSW) on Stk Adr +(BHC +1)
Yes
BUGHLT if Ref. Invalid Section
NOSKED for Duration of Trap
If Pg Trap Time Flag (IPTIM) is Set, Do Not Charge P.F. Handling to Fork's Run Time
Init Time Interval of Trap Code unless Already Initialized (ie, Nested Trap)
PGRT5
PGRT5
Get Flags, PC from TRAPFL, TRAPPC & Store on Stack
User Mode ?
Yes
Store PC in (UPDL)
No
Get Trap Status (Fail) Word, (TRAPSW) in PSB
Hard Failure?
Yes
MPE Error ?
No
Yes
Dispatch to Err. Handler
PGRTH: Pg Tb Parity/Refill Error
ILRD: Proprietary
ADRCMP: Address Compare
ILSCN: Illegal Section
ILIND: Illegal Indirect
PGMPEO
Handle in APRSRV
If Not Recoverable. Handle as Illeg. Ref. Otherwise, Retry
PGUNTP
Retry if Recoverable
pPF1a
No
PGTACC pPF2
Account for one Pg Trap
TRPRST
GETTPD pPF5
Determine cause of Trap
Dispatch to one of The following Addresses
NIC
; Pg or Pg Tbl Not in Core (pPF6)
TRPO
;Pg or Pg Tbl Age > 100 (pPF9)
NPG
;Null Ptr (pPF8)
WCPY
;Write Copy (pPF8)
ILWR
;Illegal Write (pPF7)
ILRD
;Illeg. Ref. Due to Dismounted OFN (pPF7)
PF1


Page OK to Ref.
PGUNTP
If 1st Trap & Including Pg Trap Time, Read Current Time to Complete Timing of Trap Code & Update PTTIM, HSPTTM, SPTTIM
User Mode ?
Yes
No
Flags, PC → FFL, FPC
−1 → TRAPC
OKSKED
−1 → INTDF
GOUSR:
Setup User AC BIK as current Blk
Return XJRST @ FFL
Restore Flags, PC to TRAPFL, TRAPPC & ACs 1-4,7, CX & TRAPSW from Stack
Restore P if Outer Trap
OKSKED
OKINT
Return XJRST @ TRAPFL

PF1a

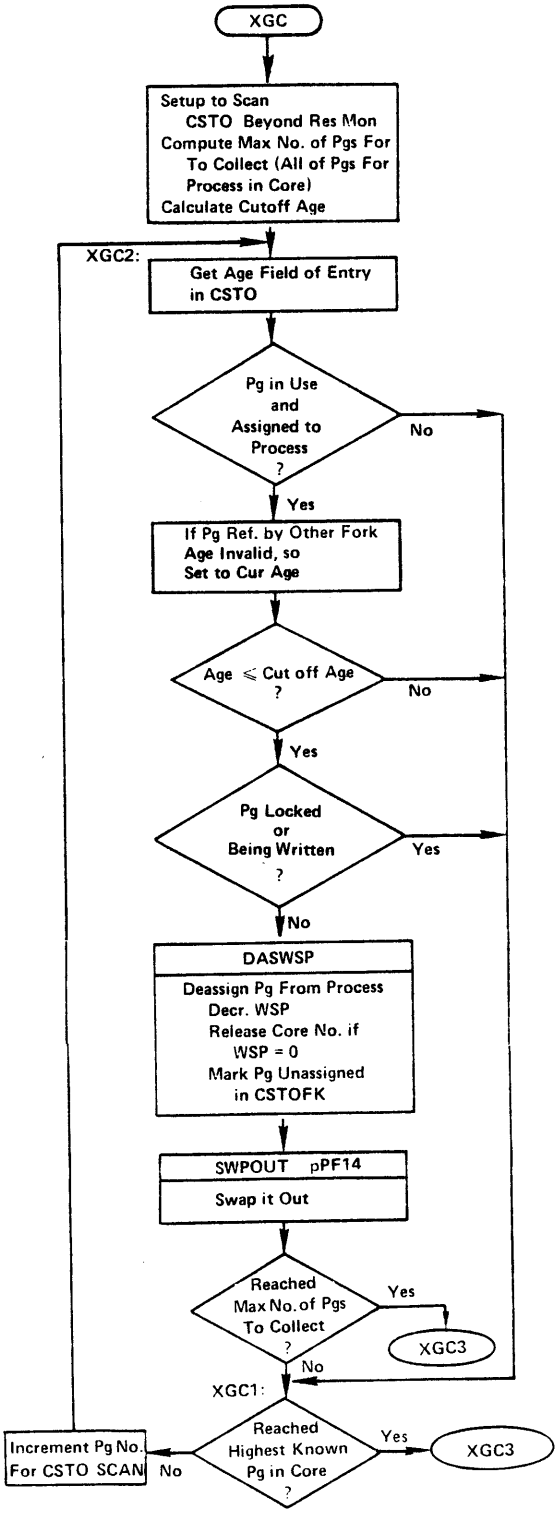

Account for one Pg Trap
PGTACC
Update Interfault Av
Av = (C x Av + Time)/C + 1
Get Subsystem Index
(if any for job) and
account for Pg fault
Get Difference between
Age at last XGC and
Current Age (Adjust for
wrap around if nec.)
Time for Another
XGC (1)
?
No
Yes
Calculate Ratio of Actual
Interfault Avg to Desired
(Desired = 2 x Avg Swp Lat)
Ratio too low
< .5?
Yes
Increase Window Size (CAPT)
(CAPT) + (CAPT)/16 → (CAPT)
No
Ratio too High
≥ 1.5?
No
Yes
Reduce Window Size (CAPT)
(CAPT) − (CAPT)/16 → (CAPT)
See if CAPT within Limits
If > CAPTMX, Set CAPT to CAPTMX
If < CAPTMN, Set CAPT to CAPTMN
Set CAPT to Fixed
Window Size (3 Sec)
A
A
XGC pPF4
Local Garbage Collect
Is Current In Core WS
Near Rerserve WS Amount
Res WS-Cur WS ≤ 4
?
Yes
No
Reduce Reserve WS
and SUMNR by X
X=Res WS-(Cur WS + 4)
NICMGI:
NICCKS pPF3
Check for
Allowable Size
Set Bit for Faulted
Pg in the Fork's WS
Bit Table in the PSB
Ret

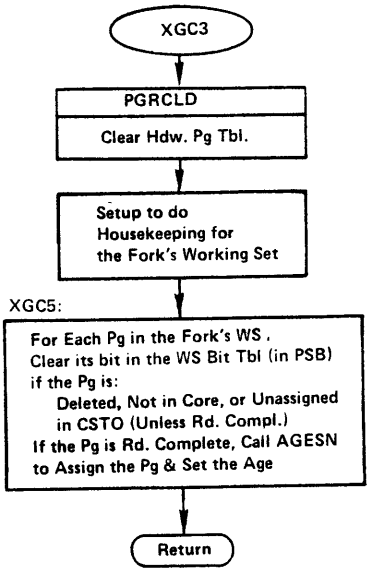
PF2


Check Overall Size for Physical Core Limit
NICCKS
Is Fork's Reserve + (GNPBAS) ≥ MAXNR ? (1a)
Yes
No
Is Reserve WS + (GNPBAS) ≥ FNPMAX ?
Yes
No
Res WS + (GNPBAS) > Cur. WS
Yes
RET
No
NIC3C:
Init Temp Window Size to 7(CAPT)/8 for Possible XGC
NIC3B:
Cur WS + 1 + (GNPBAS) ≥ MAXNR/FNPMAX
Yes
Doing Outer Level Fault ?
Yes
No
NIC3E
NIC3:
XGC pPF4
Do Local Garb. Coll.
Window Size=0 ?
No
Yes
Reduce Temp Window by CAPT/2
No
Cur WS + 1 + (GNPBAS) ≥ Smaller Max (SNPMAX) ?
Yes
System Lightly Loaded Load Av < 2 ?
Yes, Let Fork be Larger
Get Cur WS + 1
No
NIC3E:
Set Res WS = Cur WS + 1
Update SUMNR (BALSET TOTAL)
BALSET Now Too Full (SUMNR) > (MAXNR) ?
No
Yes
Set up Sched Test for BALSET Size
NICTST → ACI
RDISMS
Return

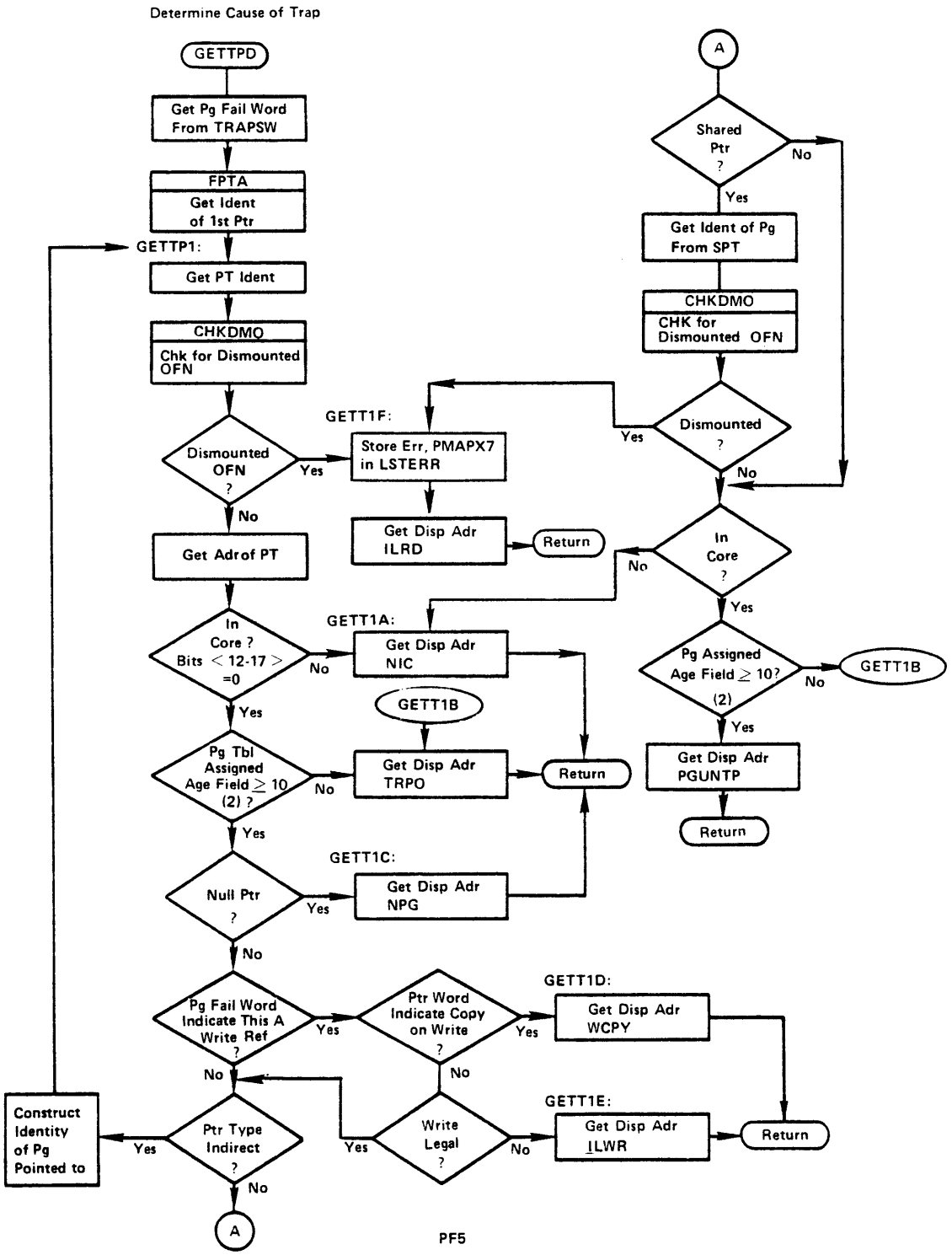
PF3

Local Garbage Collection

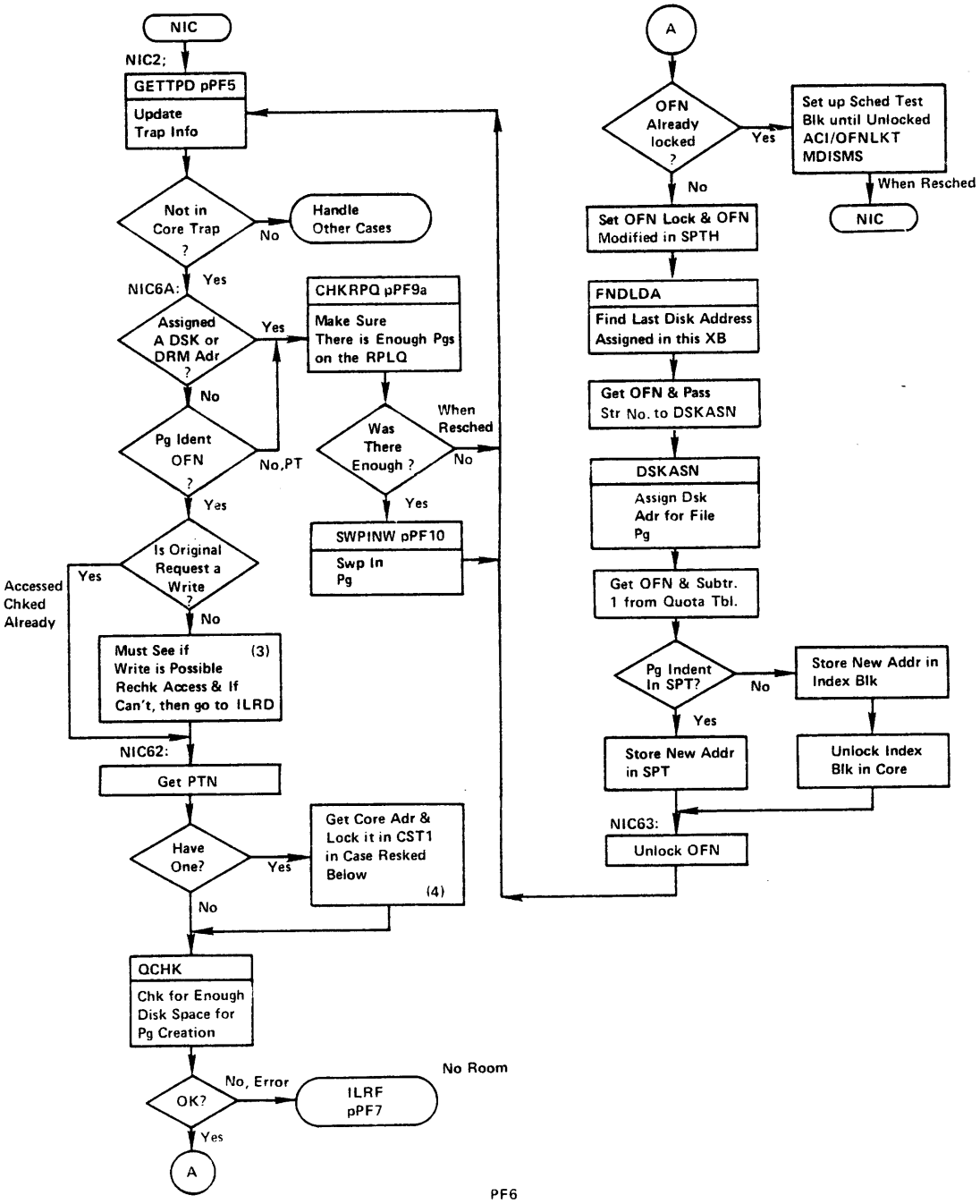




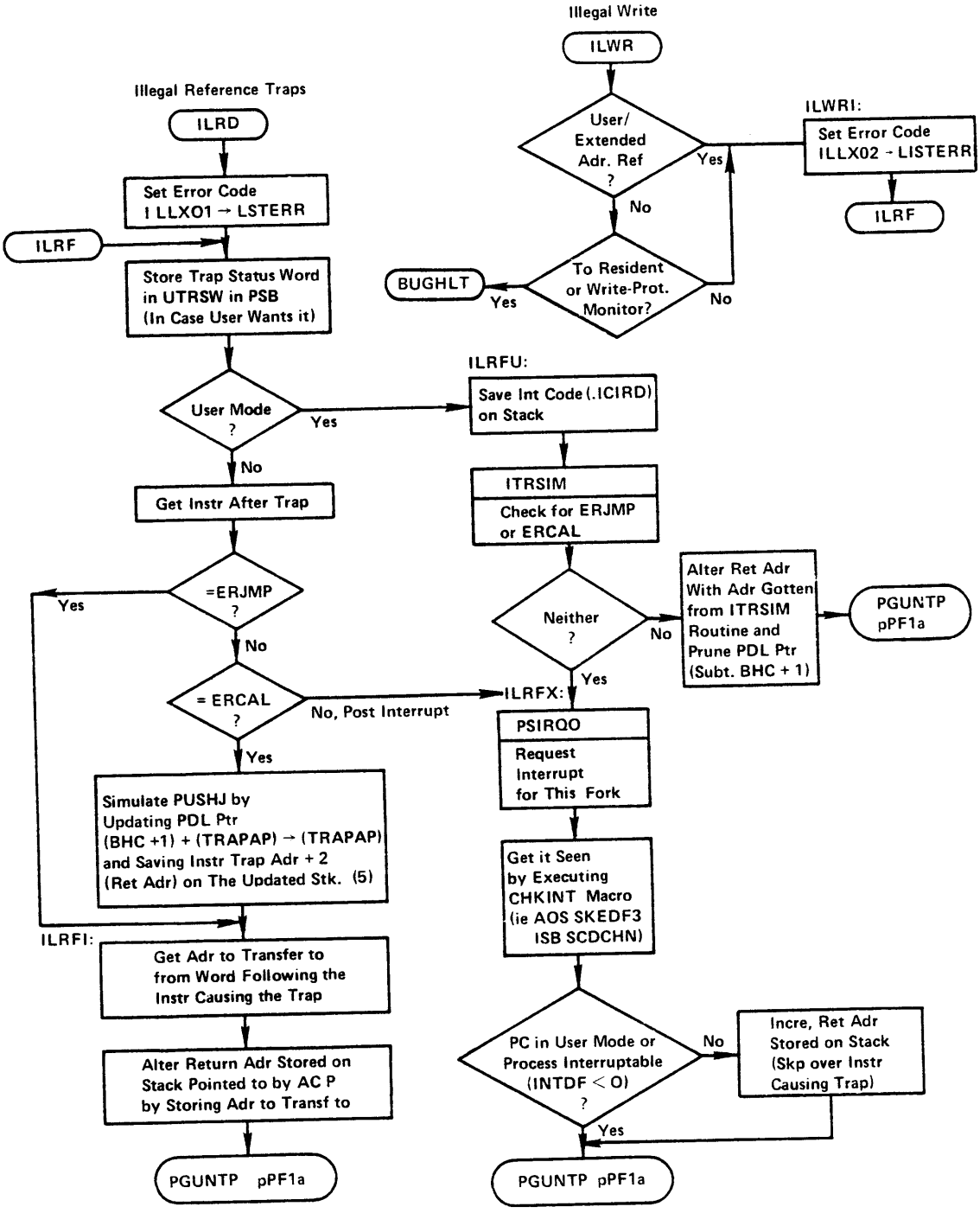
pF4


Determine Cause of Trap
GETTPD
Get Pg Fail Word From TRAPSW
FPTA
Get Ident of 1st Ptr
GETTP1:
Get PT Ident
CHKDMO
Chk for Dismounted OFN
Dismounted OFN ?
Yes
No
GETT1F:
Store Err, PMAPX7 in LSTERR
Get Disp Adr ILRD
Return
Get Adrof PT
In Core ? Bits < 12-17 > =0
No
Yes
GETT1A:
Get Disp Adr NIC
GETT1B
Pg Tbl Assigned Age Field ≥ 10 (2) ?
No
Yes
Get Disp Adr TRPO
Return
GETT1C:
Null Ptr ?
Yes
Get Disp Adr NPG
No
Pg Fail Word Indicate This A Write Ref ?
Yes
Ptr Word Indicate Copy on Write ?
Yes
GETT1D:
Get Disp Adr WCPY
No
No
Construct Identity of Pg Pointed to
Ptr Type Indirect ?
Yes
No
Write Legal ?
Yes
No
GETT1E:
Get Disp Adr ILWR
Return
A
A
Shared Ptr ?
No
Yes
Get Ident of Pg From SPT
CHKDMO
CHK for Dismounted OFN
Dismounted ?
Yes
No
In Core ?
No
Yes
Pg Assigned Age Field ≥ 10? (2)
No
GETT1B
Yes
Get Disp Adr PGUNTP
Return

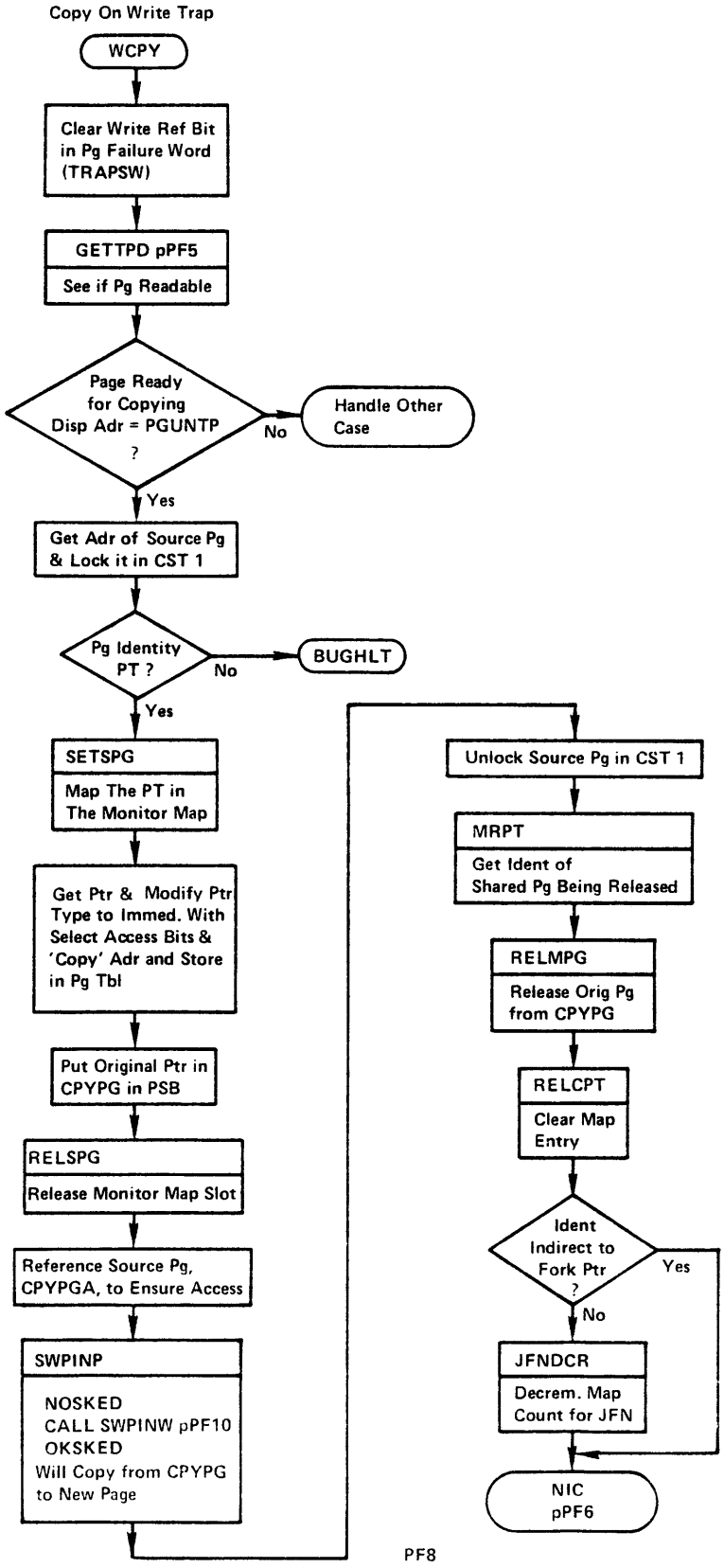
PF5


NIC
NIC2;
GETTPD pPF5
Update Trap Info
Not in Core Trap ?
No
Handle Other Cases
Yes
NIC6A:
Assigned A DSK or DRM Adr ?
Yes
CHKRPQ pPF9a
Make Sure There is Enough Pgs on the RPLQ
No
Pg Ident OFN ?
No,PT
Was There Enough ?
When Resched
No
Yes
SWPINW pPF10
Swp In Pg
Is Original Request a Write ?
Yes
Accessed Chked Already
No
Must See if Write is Possible Rechk Access & If Can't, then go to ILRD (3)
NIC62:
Get PTN
Have One?
Yes
Get Core Adr & Lock it in CST1 in Case Resked Below (4)
No
QCHK
Chk for Enough Disk Space for Pg Creation
OK?
No, Error
ILRF pPF7
No Room
Yes
A
A
OFN Already locked ?
Yes
Set up Sched Test Blk until Unlocked ACI/OFNLKT MDISMS
When Resched
NIC
No
Set OFN Lock & OFN Modified in SPTH
FNDLDA
Find Last Disk Address Assigned in this XB
Get OFN & Pass Str No. to DSKASN
DSKASN
Assign Dsk Adr for File Pg
Get OFN & Subtr. 1 from Quota Tbl.
Pg Indent In SPT?
No
Store New Addr in Index Blk
Yes
Store New Addr in SPT
Unlock Index Blk in Core
NIC63:
Unlock OFN

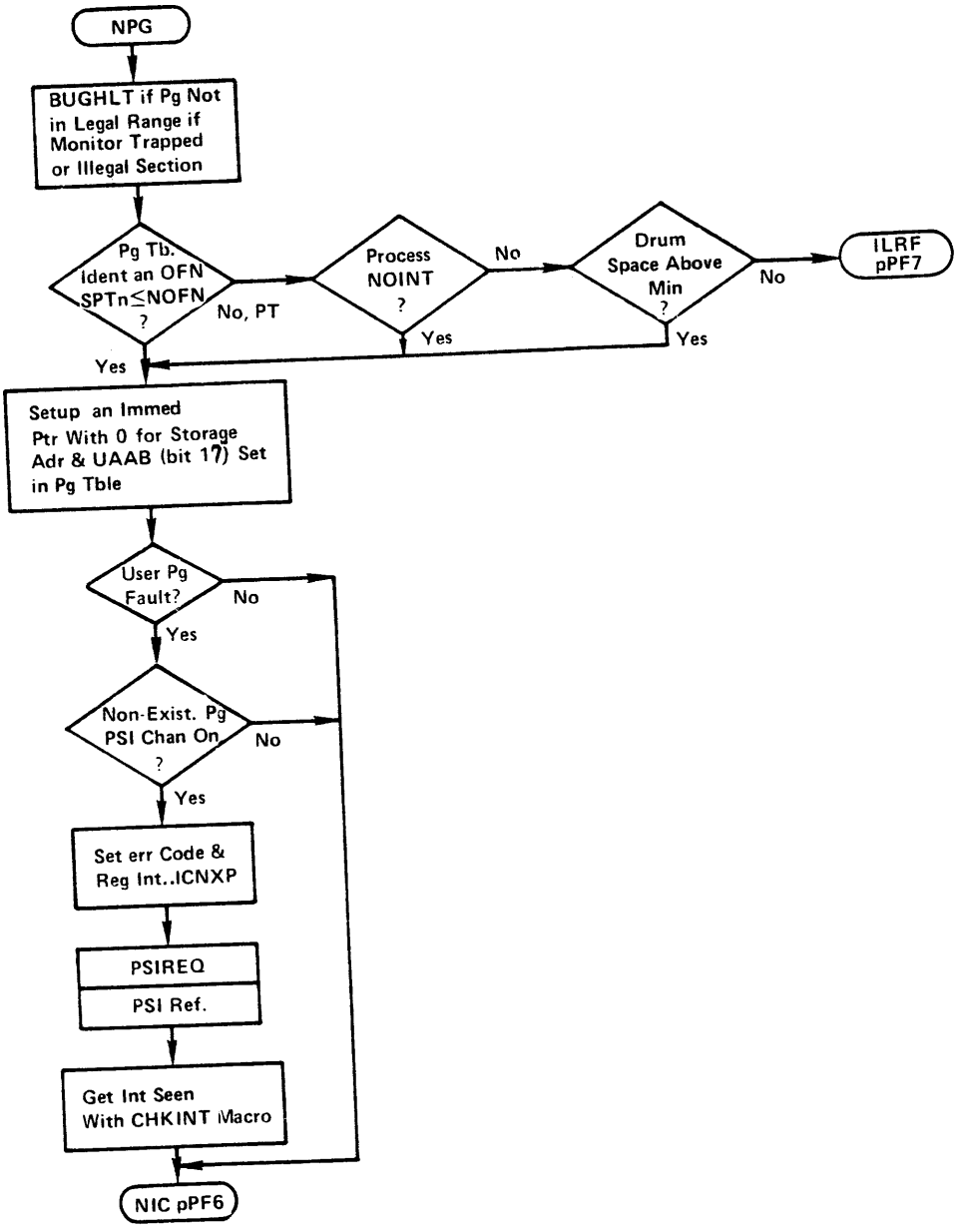
PF6


Illegal Write
ILWR
User/ Extended Adr. Ref ?
Yes
ILWRI:
Set Error Code ILLX02 - LISTERR
ILRF
No
To Resident or Write-Prot. Monitor?
Yes
BUGHLT
No
Illegal Reference Traps
ILRD
Set Error Code I LLXO1 → LSTERR
ILRF
Store Trap Status Word in UTRSW in PSB (In Case User Wants it)
User Mode ?
Yes
ILRFU:
Save Int Code (.ICIRD) on Stack
No
Get Instr After Trap
ITRSIM
Check for ERJMP or ERCAL
=ERJMP ?
Yes
Neither ?
No
Alter Ret Adr With Adr Gotten from ITRSIM Routine and Prune PDL Ptr (Subt. BHC + 1)
PGUNTP pPF1a
No
Yes
= ERCAL ?
No, Post Interrupt
ILRFX:
PSIRQO
Request Interrupt for This Fork
Yes
Simulate PUSHJ by Updating PDL Ptr (BHC +1) + (TRAPAP) → (TRAPAP) and Saving Instr Trap Adr + 2 (Ret Adr) on The Updated Stk. (5)
Get it Seen by Executing CHKINT Macro (ie AOS SKEDF3 ISB SCDCHN)
ILRFI:
Get Adr to Transfer to from Word Following the Instr Causing the Trap
PC in User Mode or Process Interruptable (INTDF < 0) ?
No
Incre, Ret Adr Stored on Stack (Skp over Instr Causing Trap)
Alter Return Adr Stored on Stack Pointed to by AC P by Storing Adr to Transf to
Yes
PGUNTP pPF1a
PGUNTP pPF1a

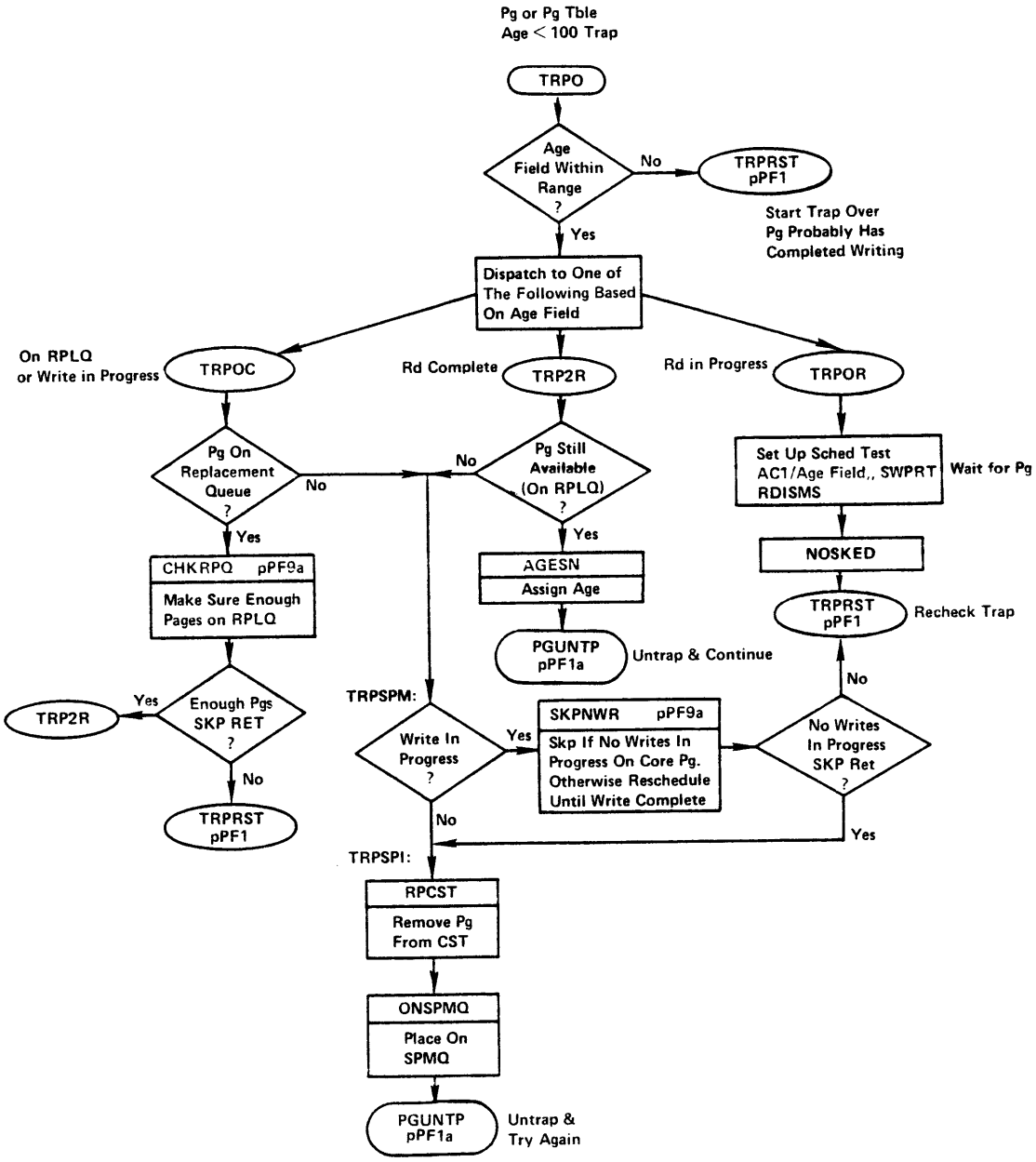
PF7


Copy On Write Trap
WCPY
Clear Write Ref Bit in Pg Failure Word (TRAPSW)
GETTPD pPF5
See if Pg Readable
Page Ready for Copying Disp Adr = PGUNTP ?
No
Handle Other Case
Yes
Get Adr of Source Pg & Lock it in CST 1
Pg Identity PT ?
No
BUGHLT
Yes
SETSPG
Map The PT in The Monitor Map
Get Ptr & Modify Ptr Type to Immed. With Select Access Bits & 'Copy' Adr and Store in Pg Tbl
Put Original Ptr in CPYPG in PSB
RELSPG
Release Monitor Map Slot
Reference Source Pg, CPYPGA, to Ensure Access
SWPINP
NOSKED
CALL SWPINW pPF10
OKSKED
Will Copy from CPYPG to New Page
PF8
Unlock Source Pg in CST 1
MRPT
Get Ident of Shared Pg Being Released
RELMPG
Release Orig Pg from CPYPG
RELCPT
Clear Map Entry
Ident Indirect to Fork Ptr ?
Yes
No
JFNDCR
Decrem. Map Count for JFN
NIC pPF6

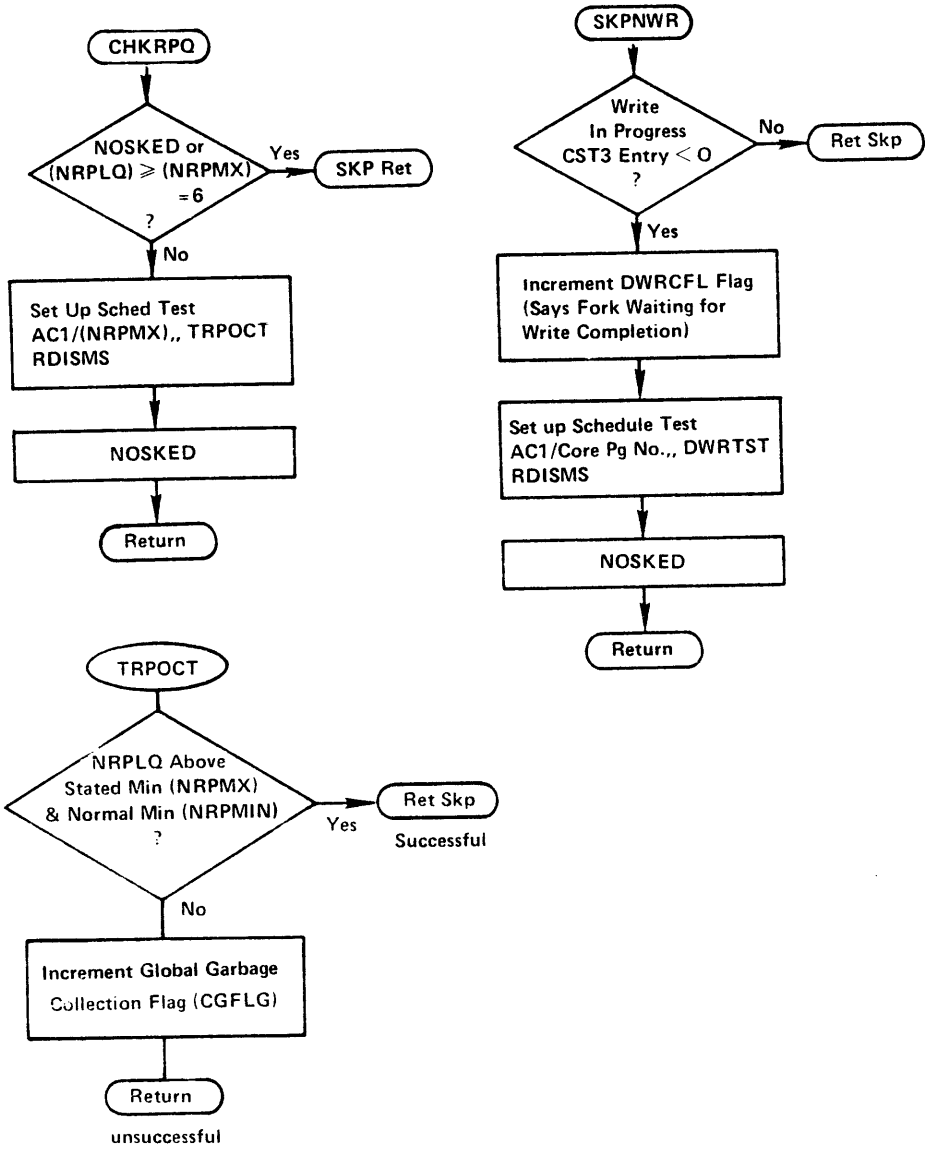
Pg Not in Existance Trap



PF 8a


Pg or Pg Tble
Age < 100 Trap
TRPO
Age
Field Within
Range
?
No
TRPRST
pPF1
Start Trap Over
Pg Probably Has
Completed Writing
Yes
Dispatch to One of
The Following Based
On Age Field
On RPLQ
or Write in Progress
TRPOC
Rd Complete
TRP2R
Rd in Progress
TRPOR
Pg On
Replacement
Queue
?
No
Pg Still
Available
(On RPLQ)
?
No
Yes
Yes
Set Up Sched Test
AC1/Age Field,, SWPRT
RDISMS
Wait for Pg
CHKRPQ pPF9a
Make Sure Enough
Pages on RPLQ
AGESN
Assign Age
NOSKED
TRPRST
pPF1
Recheck Trap
PGUNTP
pPF1a
Untrap & Continue
Yes
Enough Pgs
SKP RET
?
TRP2R
No
TRPRST
pPF1
TRPSPM:
Write In
Progress
?
Yes
SKPNWR pPF9a
Skp If No Writes In
Progress On Core Pg.
Otherwise Reschedule
Until Write Complete
No Writes
In Progress
SKP Ret
?
No
Yes
No
TRPSPI:
RPCST
Remove Pg
From CST
ONSPMQ
Place On
SPMQ
PGUNTP
pPF1a
Untrap &
Try Again

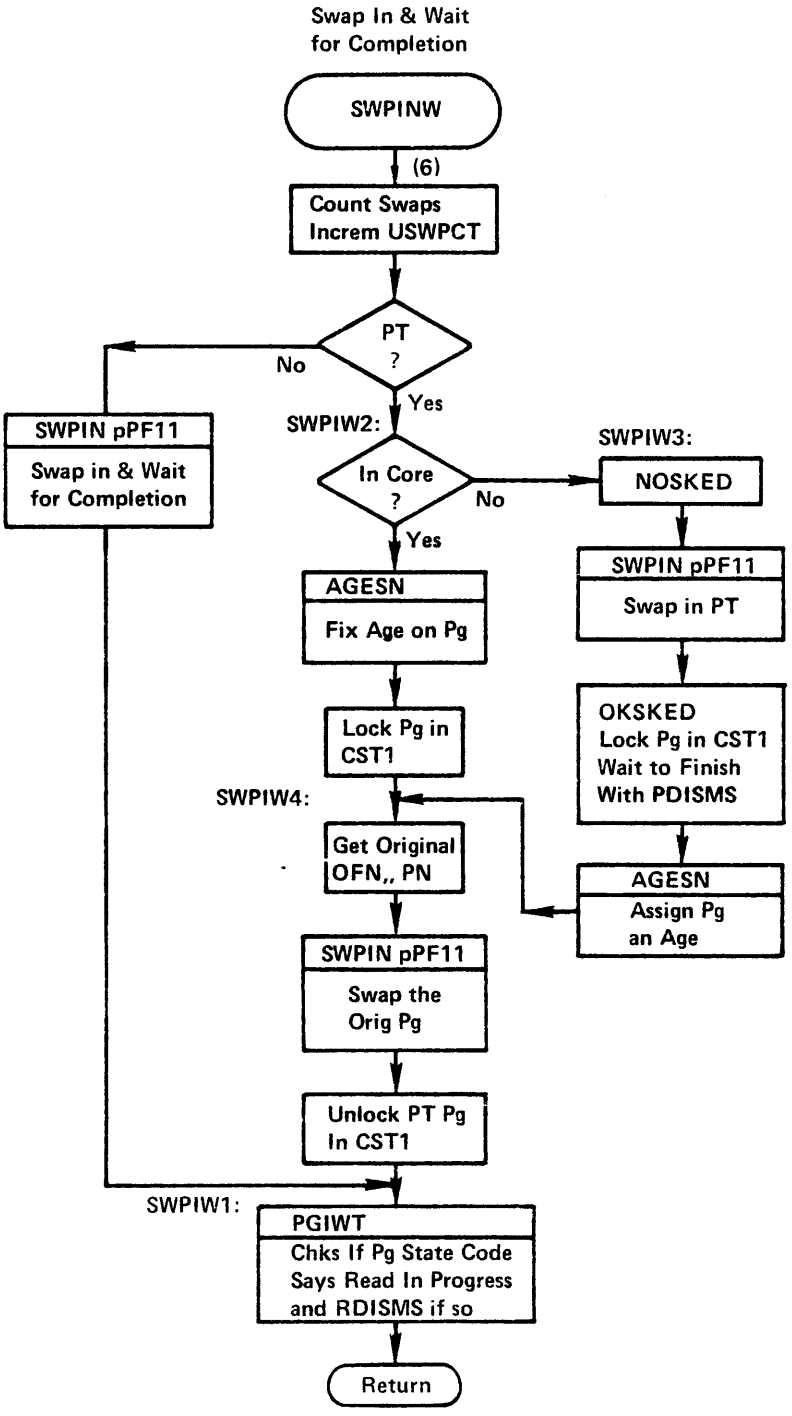
PF9


CHKRPQ
NOSKED or (NRPLQ) ≥ (NRPMX) =6 ?
Yes
SKP Ret
No
Set Up Sched Test AC1/(NRPMX),, TRPOCT RDISMS
NOSKED
Return
SKPNWR
Write In Progress CST3 Entry < 0 ?
No
Ret Skp
Yes
Increment DWRCFL Flag (Says Fork Waiting for Write Completion)
Set up Schedule Test AC1/Core Pg No.,, DWRTST RDISMS
NOSKED
Return
TRPOCT
NRPLQ Above Stated Min (NRPMX) & Normal Min (NRPMIN) ?
Yes
Ret Skp
Successful
No
Increment Global Garbage Collection Flag (CGFLG)
Return
unsuccessful

PF9a

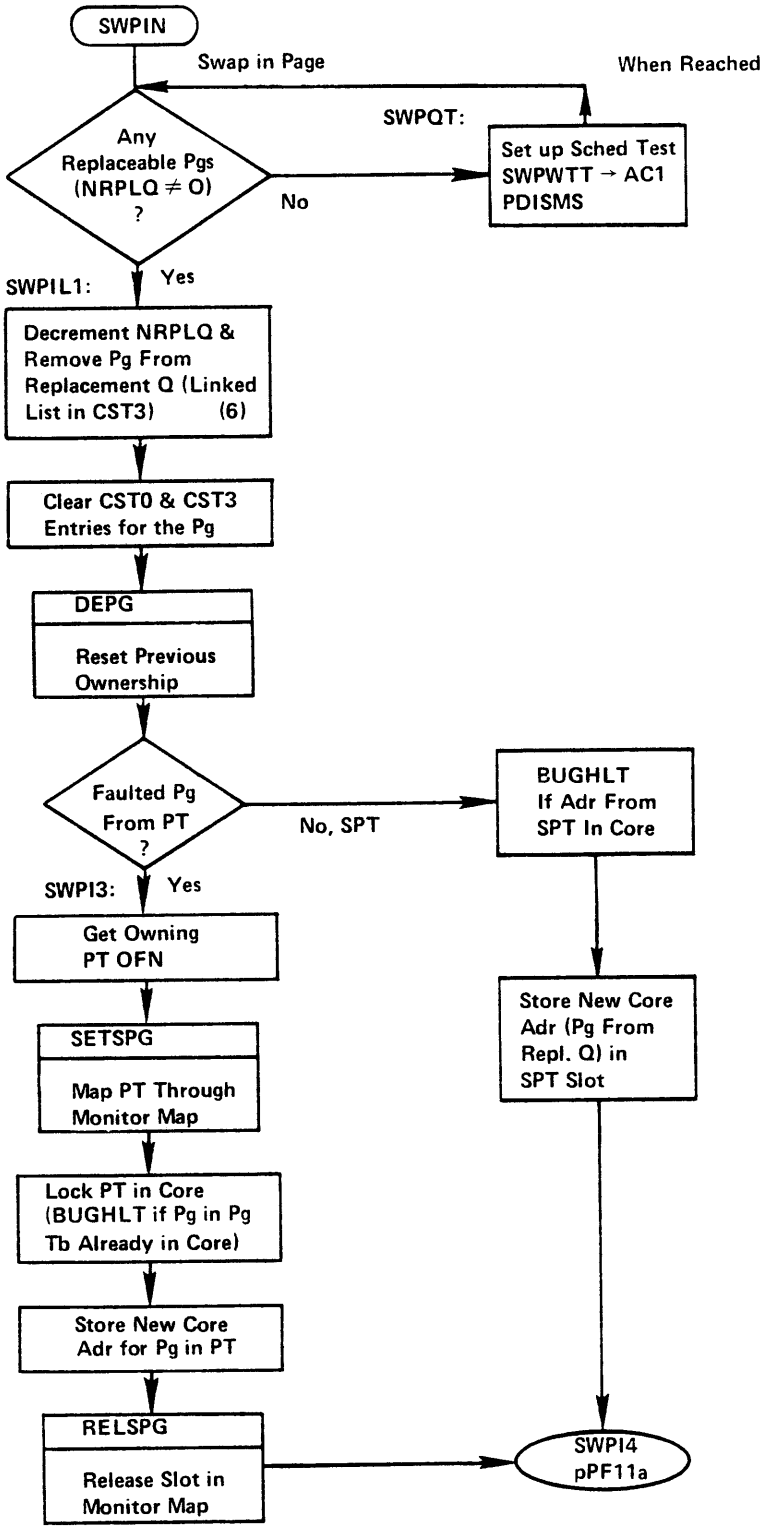
REQUESTING DRUM OR DISK READ

(PAGEM LEVEL)

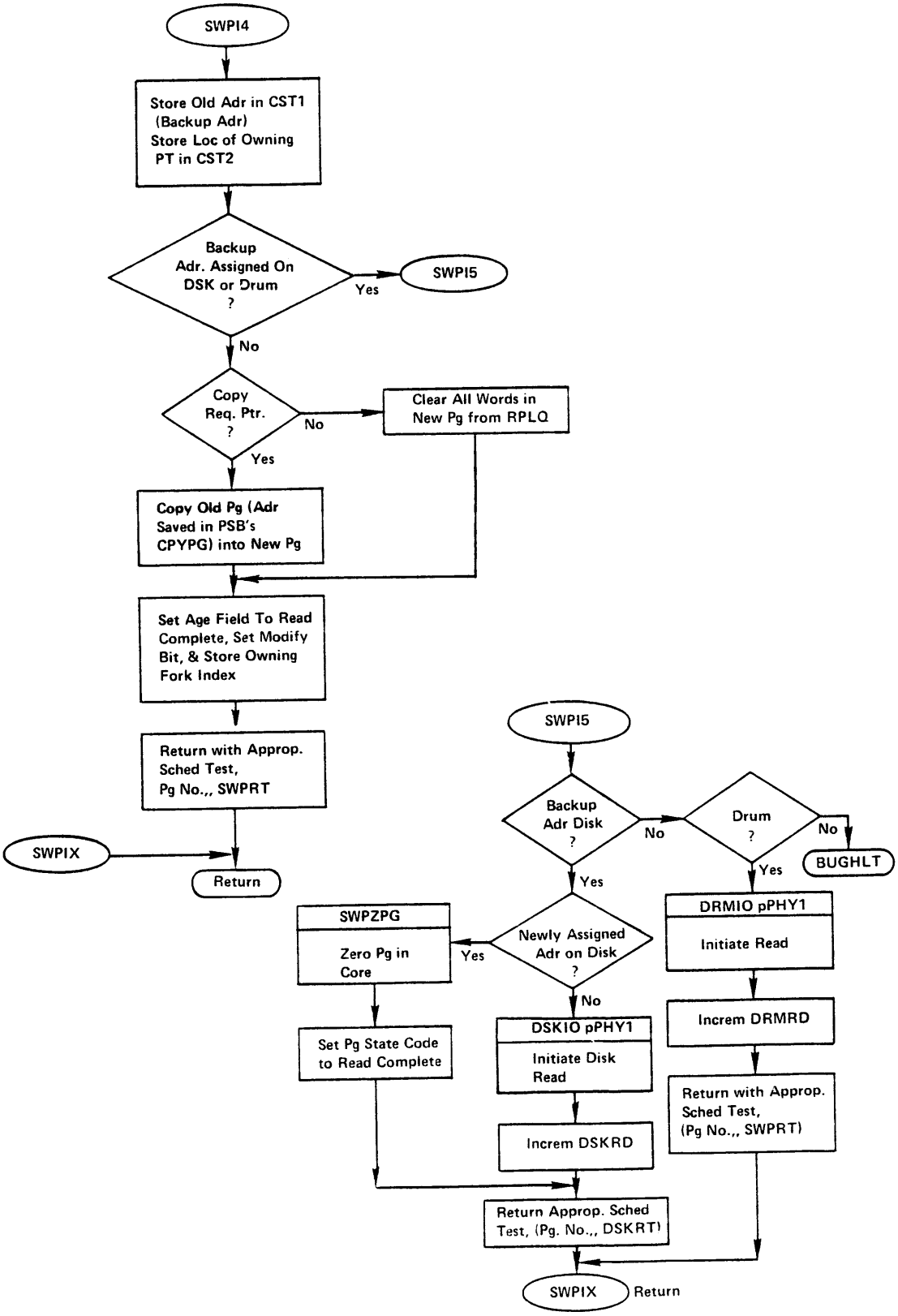


PF10

REQUESTING DRUM OR DISK READ (Continued)
(PAGEM LEVEL)

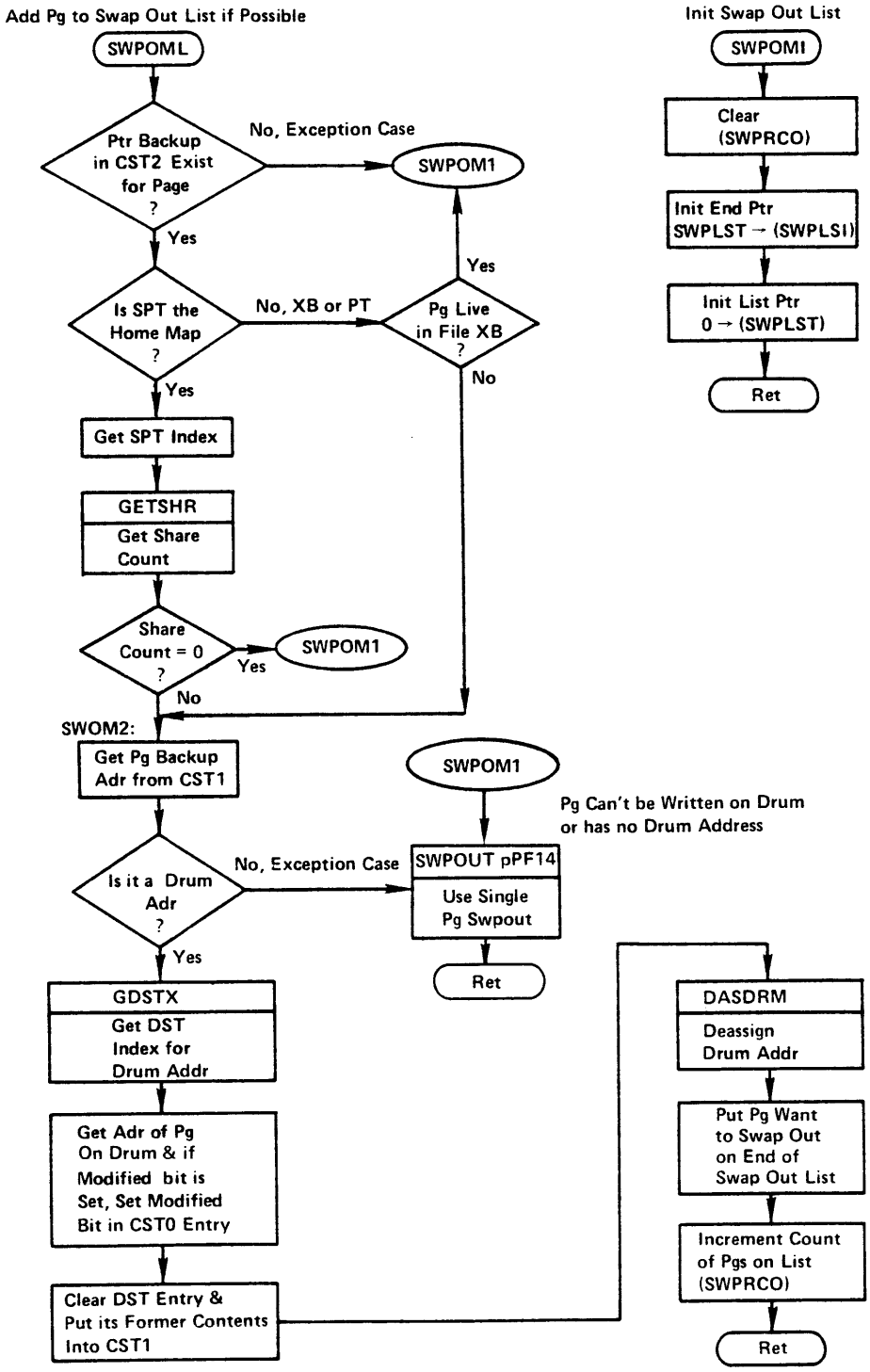


PF11

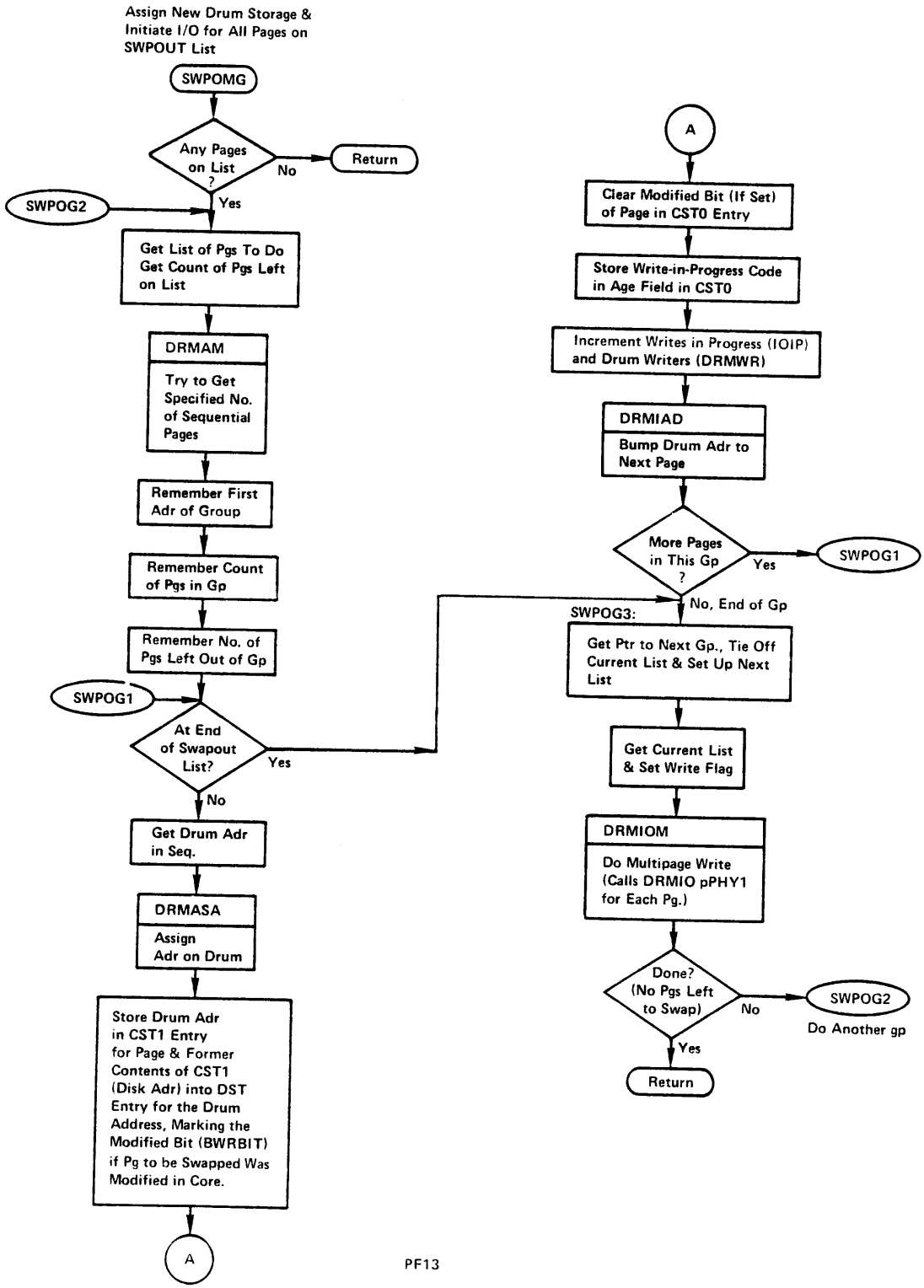

SWPI4
Store Old Adr in CST1 (Backup Adr) Store Loc of Owning PT in CST2
Backup Adr. Assigned On DSK or Drum ?
Yes
SWPI5
No
Copy Req. Ptr. ?
No
Clear All Words in New Pg from RPLQ
Yes
Copy Old Pg (Adr Saved in PSB's CPYPG) into New Pg
Set Age Field To Read Complete, Set Modify Bit, & Store Owning Fork Index
Return with Approp. Sched Test, Pg No.,, SWPRT
SWPIX
Return
SWPI5
Backup Adr Disk ?
No
Drum ?
No
BUGHLT
Yes
Yes
SWPZPG
Zero Pg in Core
Yes
Newly Assigned Adr on Disk ?
No
DRMIO pPHY1
Initiate Read
Increm DRMRD
Return with Approp. Sched Test, (Pg No.,, SWPRT)
Set Pg State Code to Read Complete
DSKIO pPHY1
Initiate Disk Read
Increm DSKRD
Return Approp. Sched Test, (Pg. No.,, DSKRT)
SWPIX
Return

PF11a

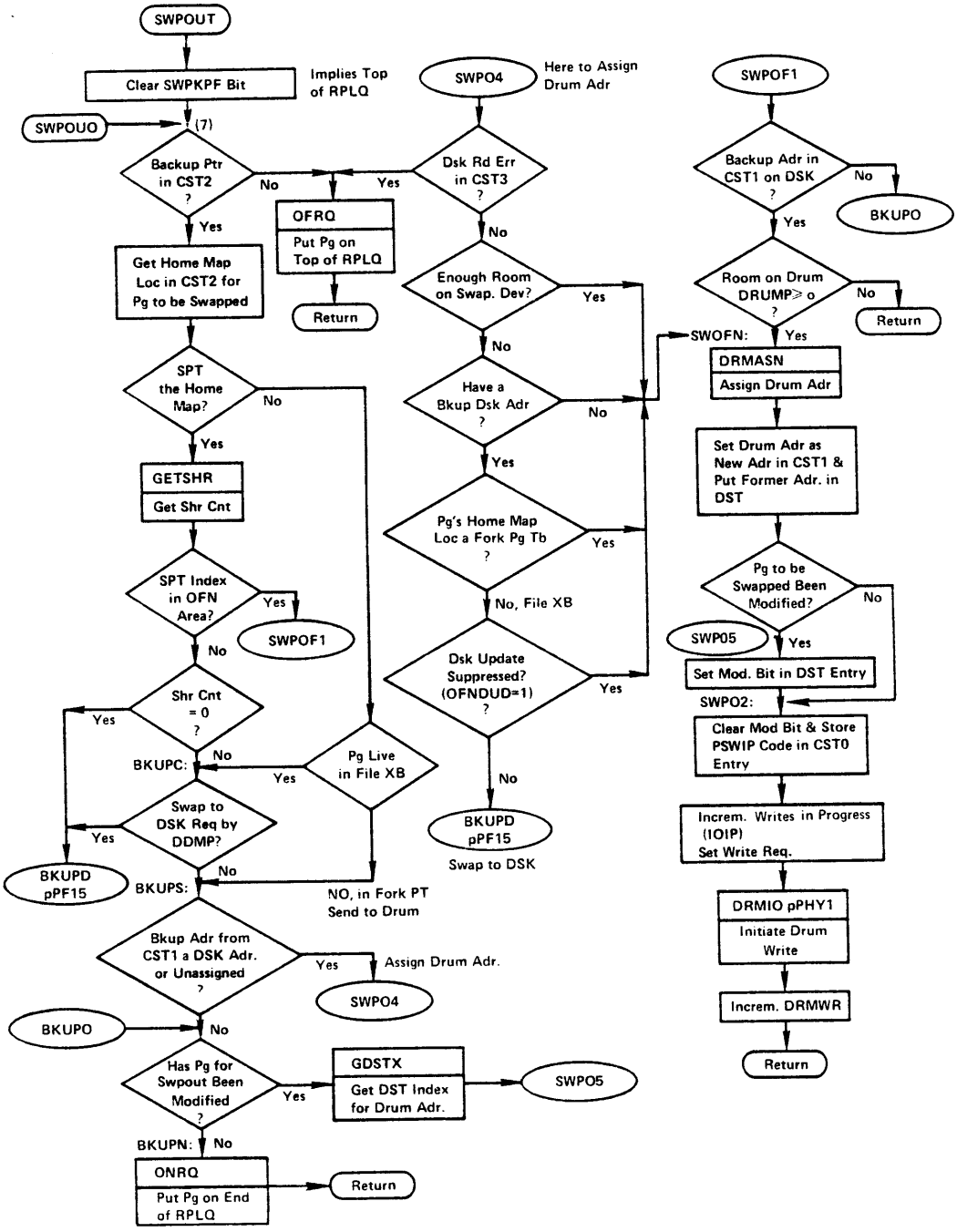
MULTIPLE PAGE SWAP OUT ROUTINE
SWPOMI - Init List
SWPOML - Called to Add Page to Swap Out List if Possible
SWPOMG - To Begin I/O for All Pages on Swap Out List
SWPOUT - Initiate Swap Out of Single Page



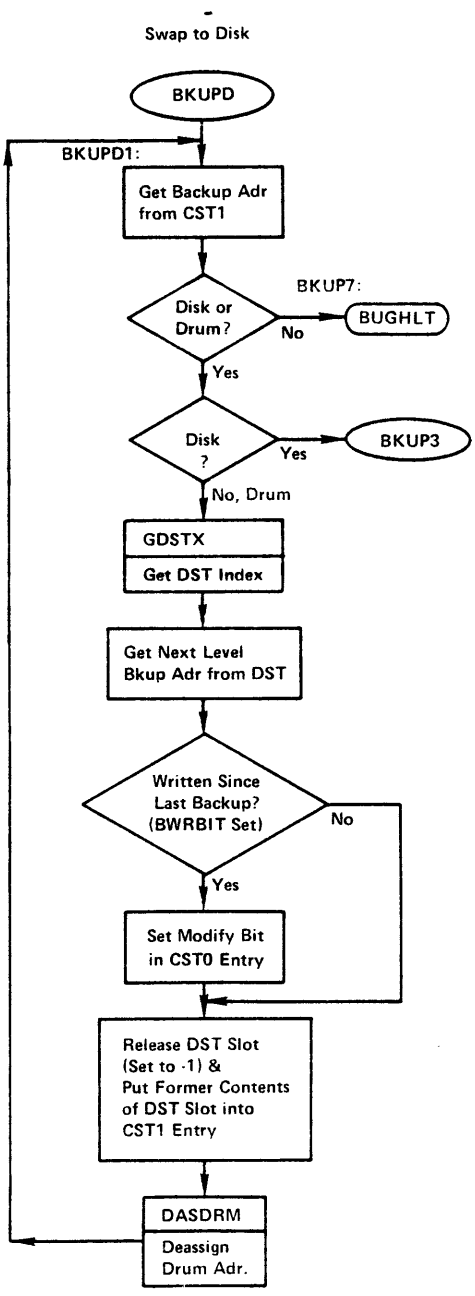
PF12

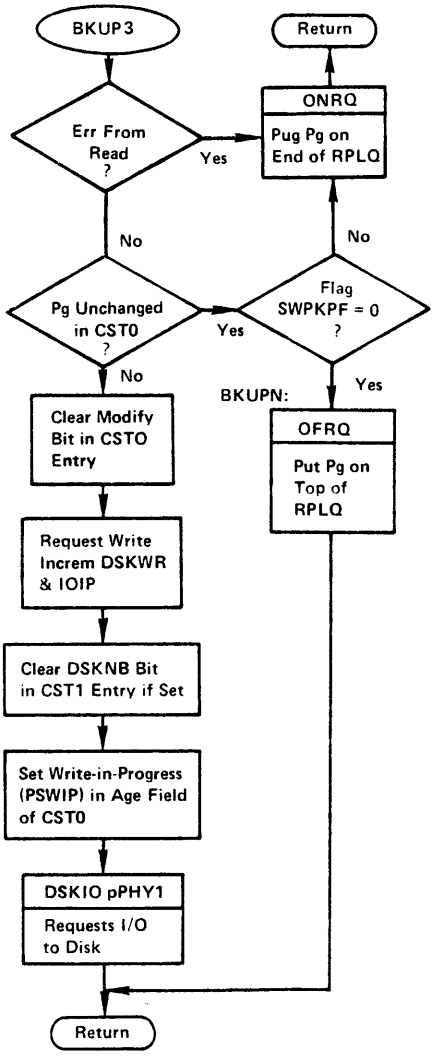

Assign New Drum Storage & Initiate I/O for All Pages on SWPOUT List
SWPOMG
Any Pages on List ?
No
Return
Yes
SWPOG2
Get List of Pgs To Do Get Count of Pgs Left on List
DRMAM
Try to Get Specified No. of Sequential Pages
Remember First Adr of Group
Remember Count of Pgs in Gp
Remember No. of Pgs Left Out of Gp
SWPOG1
At End of Swapout List?
Yes
No
Get Drum Adr in Seq.
DRMASA
Assign Adr on Drum
Store Drum Adr in CST1 Entry for Page & Former Contents of CST1 (Disk Adr) into DST Entry for the Drum Address, Marking the Modified Bit (BWRBIT) if Pg to be Swapped Was Modified in Core.
A
A
Clear Modified Bit (If Set) of Page in CST0 Entry
Store Write-in-Progress Code in Age Field in CST0
Increment Writes in Progress (IOIP) and Drum Writers (DRMWR)
DRMIAD
Bump Drum Adr to Next Page
More Pages in This Gp ?
Yes
SWPOG1
No, End of Gp
SWPOG3:
Get Ptr to Next Gp., Tie Off Current List & Set Up Next List
Get Current List & Set Write Flag
DRMIOM
Do Multipage Write (Calls DRMIO pPHY1 for Each Pg.)
Done? (No Pgs Left to Swap)
No
SWPOG2
Do Another gp
Yes
Return

PF13


SWPOUT
Clear SWPKPF Bit
Implies Top of RPLQ
SWPOUO
(7)
Backup Ptr in CST2 ?
No
Yes
OFRQ
Put Pg on Top of RPLQ
Return
Yes
Get Home Map Loc in CST2 for Pg to be Swapped
SPT the Home Map?
No
Yes
GETSHR
Get Shr Cnt
SPT Index in OFN Area?
Yes
SWPOF1
No
Shr Cnt = 0 ?
Yes
No
BKUPC:
Pg Live in File XB
Yes
Swap to DSK Req by DDMP?
Yes
BKUPD pPF15
No
BKUPS:
NO, in Fork PT Send to Drum
Bkup Adr from CST1 a DSK Adr. or Unassigned ?
Yes
Assign Drum Adr.
SWPO4
BKUPO
No
Has Pg for Swpout Been Modified ?
Yes
GDSTX
Get DST Index for Drum Adr.
SWPO5
BKUPN:
No
ONRQ
Put Pg on End of RPLQ
Return
SWPO4
Here to Assign Drum Adr
Dsk Rd Err in CST3 ?
Yes
No
Enough Room on Swap. Dev?
Yes
No
Have a Bkup Dsk Adr ?
No
Yes
Pg's Home Map Loc a Fork Pg Tb ?
Yes
No, File XB
Dsk Update Suppressed? (OFNDUD=1) ?
Yes
No
BKUPD pPF15
Swap to DSK
SWOFN:
SWPOF1
Backup Adr in CST1 on DSK ?
No
BKUPO
Yes
Room on Drum DRUMP≥ o ?
No
Return
Yes
DRMASN
Assign Drum Adr
Set Drum Adr as New Adr in CST1 & Put Former Adr. in DST
Pg to be Swapped Been Modified?
No
SWP05
Yes
Set Mod. Bit in DST Entry
SWPO2:
Clear Mod Bit & Store PSWIP Code in CST0 Entry
Increm. Writes in Progress (IOIP)
Set Write Req.
DRMIO pPHY1
Initiate Drum Write
Increm. DRMWR
Return

PF14


Swap to Disk
BKUPD
BKUPD1:
Get Backup Adr from CST1
BKUP7:
Disk or Drum?
No
BUGHLT
Yes
Disk ?
Yes
BKUP3
No, Drum
GDSTX
Get DST Index
Get Next Level Bkup Adr from DST
Written Since Last Backup? (BWRBIT Set)
No
Yes
Set Modify Bit in CST0 Entry
Release DST Slot (Set to -1) & Put Former Contents of DST Slot into CST1 Entry
DASDRM
Deassign Drum Adr.


BKUP3
Return
Err From Read ?
Yes
ONRQ
Pug Pg on End of RPLQ
No
No
Pg Unchanged in CST0 ?
Yes
Flag SWPKPF = 0 ?
No
Yes
BKUPN:
Clear Modify Bit in CSTO Entry
OFRQ
Put Pg on Top of RPLQ
Request Write Increm DSKWR & IOIP
Clear DSKNB Bit in CST1 Entry if Set
Set Write-in-Progress (PSWIP) in Age Field of CST0
DSKIO pPHY1
Requests I/O to Disk
Return

PF15

Page Fault Handling Comments

PGTACC

(1) Checks if process has accrued more than or equal to the number of age ticks of GCRATE. Currently, this is set to 50, which implies 2 sec. of process virtual time (i.e., the age stamp is incremented every 40 ms of process run time).

NICCKS

(1) GNPBAS is currently initialized at system startup to zero and is incremented/decremented only when pages are locked/unlocked. It is currently only tested by NICCKS as well.

GETTPD

(2) The age field when used to hold the age stamp, will always have a value of 100 or greater. This checks if any of the lefthand 6 bits of the age field are set.

NIC

(3) Could take the ILRD path, for example, when OPENed file for write, but PMAPed for each of a nonexistent page. A page would have to be created which would then imply a write which was not enabled under PMAP.

(4) If file page faulted does not have its own SPT slot, but has to be mapped (using indirect pointer) via the index blk slot in the OFN area, then the index blk will be locked in core. (So can't be swapped in case of reschedule.)

(5) Note in the predispatch code that AC1 was stored in BHC + 1 and AC, P, which holds a push down list pointer, was saved in TRAPAP.

SWPINW

(6) SWPINW will invoke SWPIN to swap in a page into a page from the RPLQ. However, this same code can also be entered with different flag settings and be used to swap in a page into a page from the special memory queue (SPMQ), a queue used by the memory error handling code.

SWPOUT

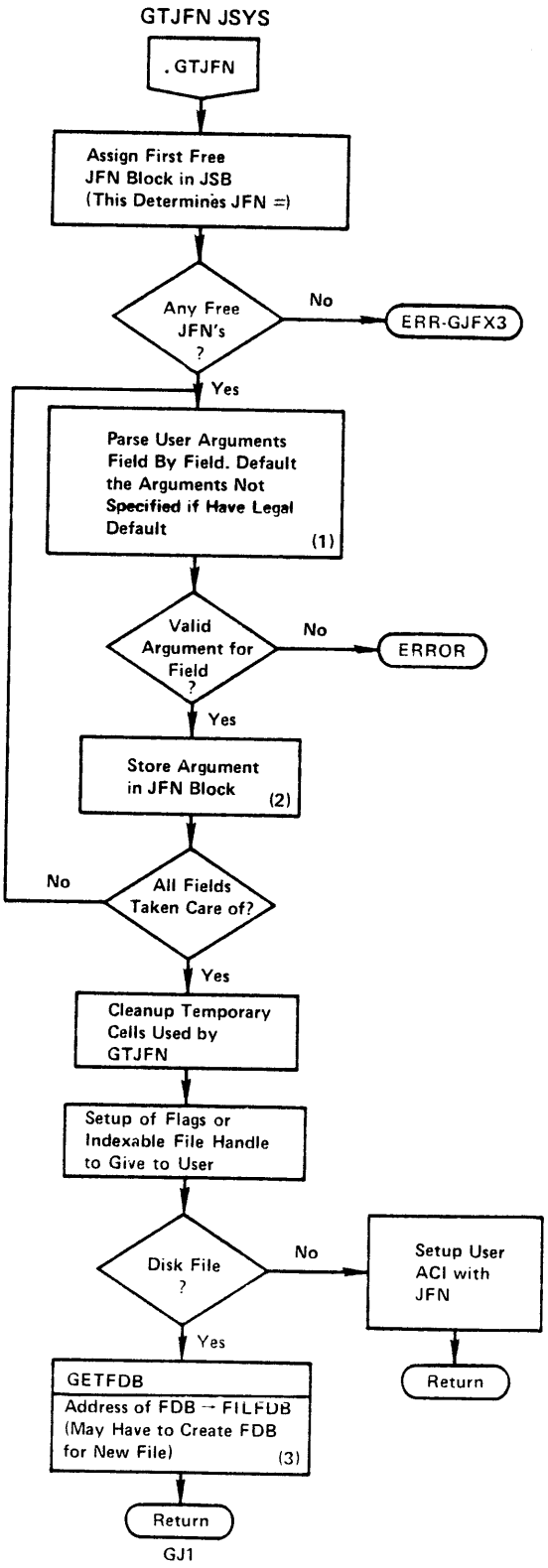
(7) SWPOUO is called from:

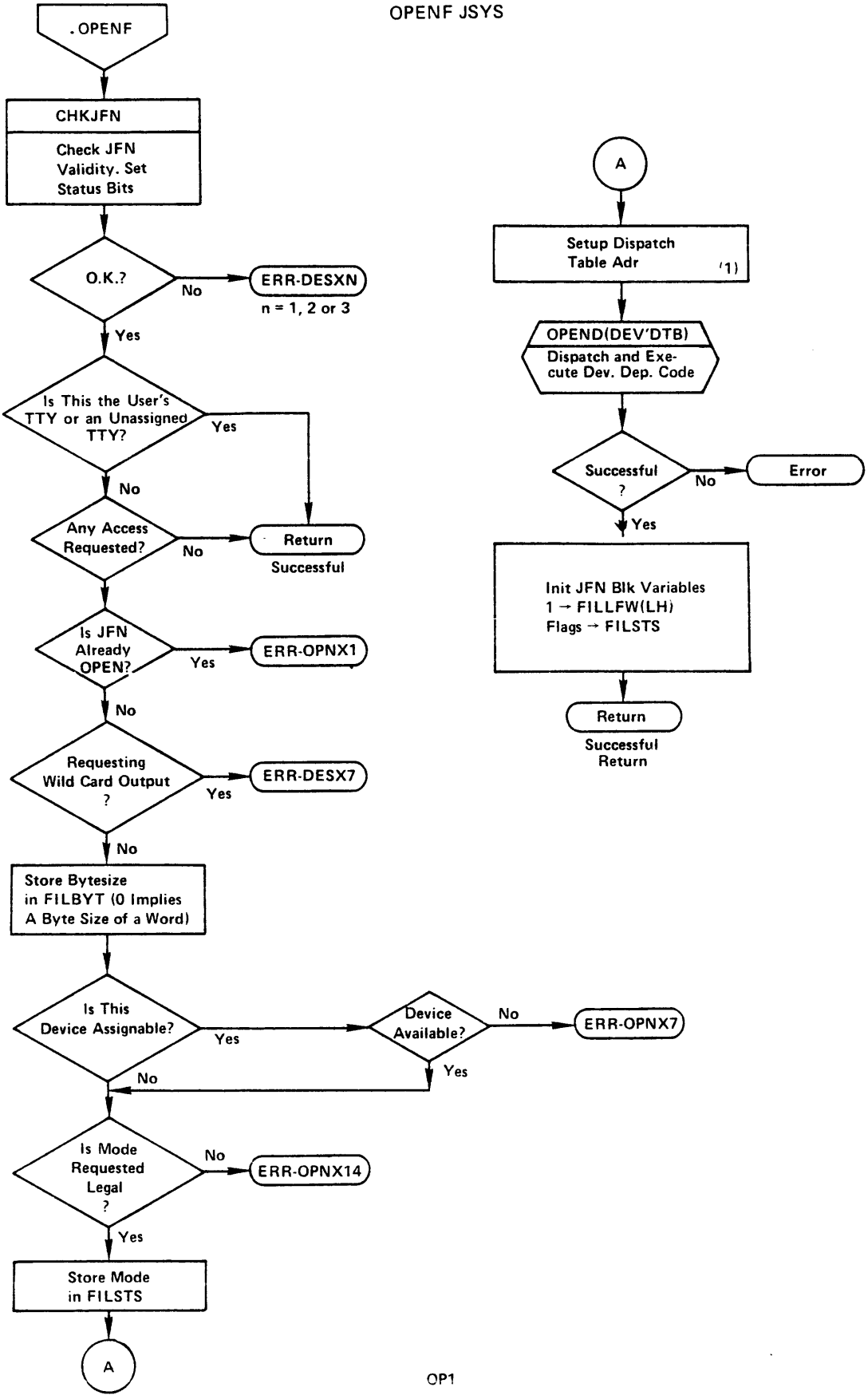
SWPOTO which clears the SWPKPF bit (for top of RPLQ) before calling SWPOUO and

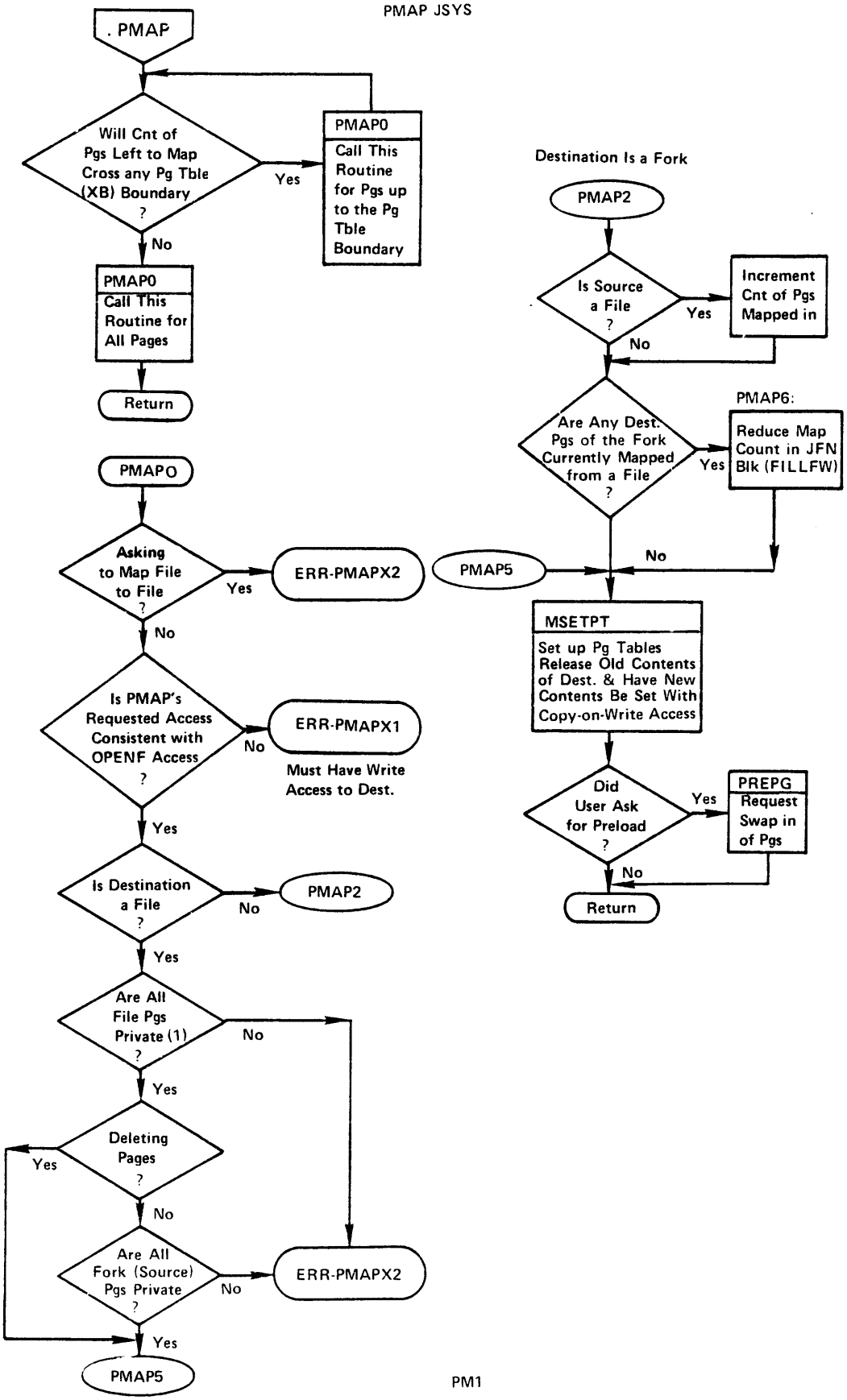
SWPOTK (called from the UPDPGS JSYS) which sets the SWPKPF bit (for end of RPLQ) before calling SWPOUO.

JSYS CALL FLOWCHARTS

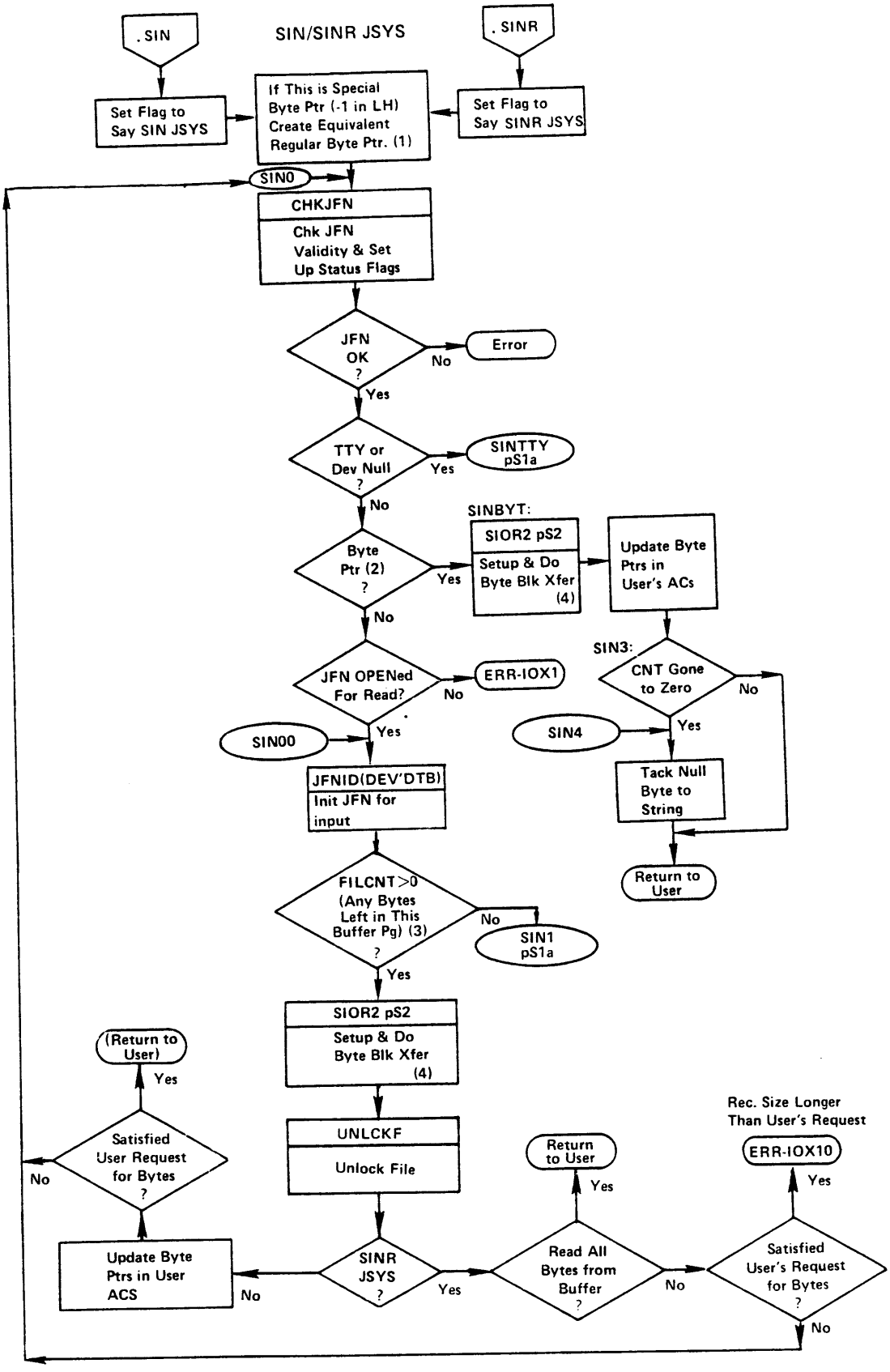
DEVICE INDEPENDENT LEVEL


GTJFN JSYS
.GTJFN
Assign First Free JFN Block in JSB (This Determines JFN =)
Any Free JFN's ?
No
ERR-GJFX3
Yes
Parse User Arguments Field By Field. Default the Arguments Not Specified if Have Legal Default
(1)
Valid Argument for Field ?
No
ERROR
Yes
Store Argument in JFN Block
(2)
All Fields Taken Care of?
No
Yes
Cleanup Temporary Cells Used by GTJFN
Setup of Flags or Indexable File Handle to Give to User
Disk File ?
No
Setup User ACI with JFN
Return
Yes
GETFDB
Address of FDB → FILFDB (May Have to Create FDB for New File)
(3)
Return
GJ1

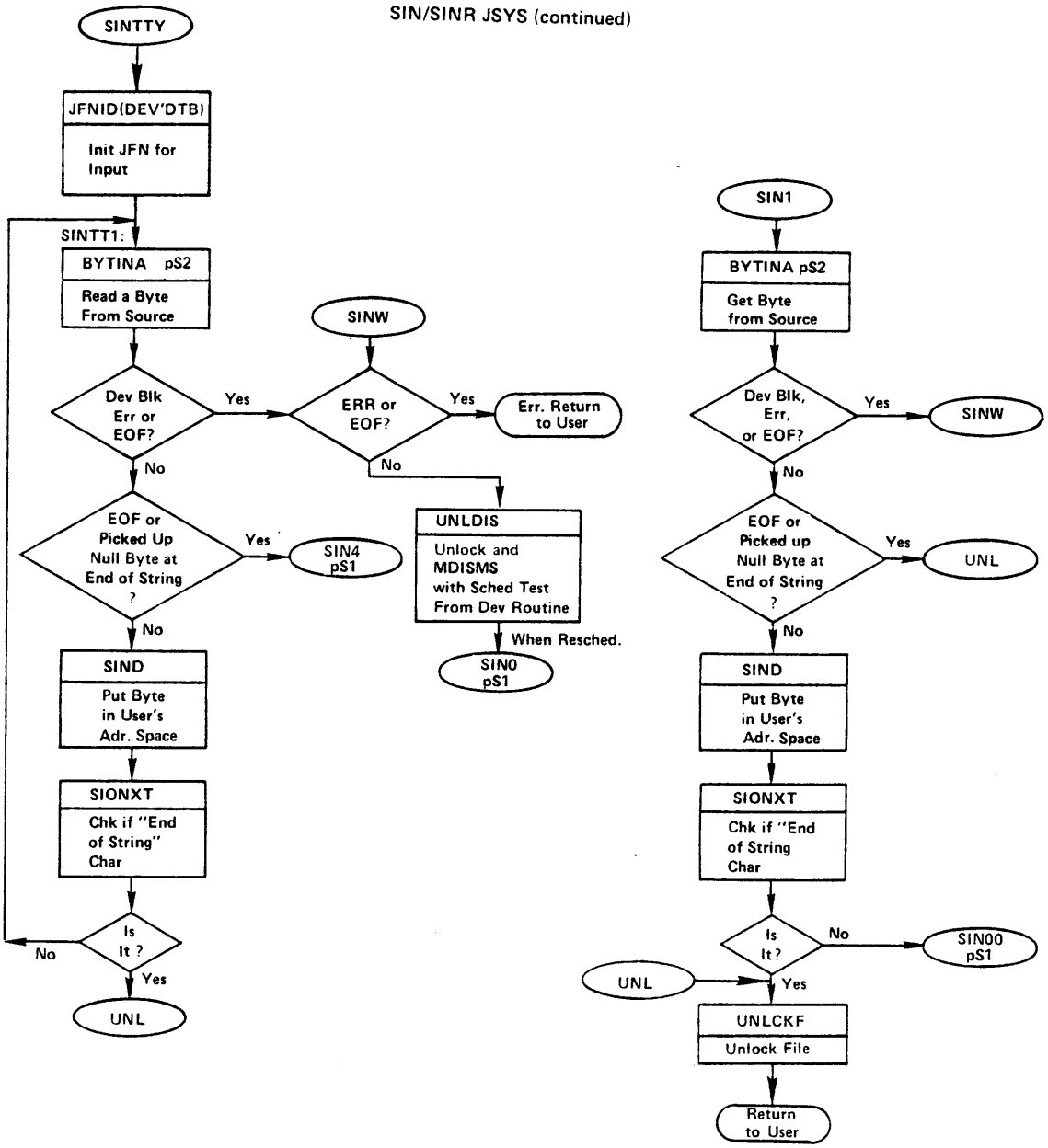

.OPENF
CHKJFN
Check JFN Validity. Set Status Bits
O.K.?
No
ERR-DESXN
n = 1, 2 or 3
Yes
Is This the User's TTY or an Unassigned TTY?
Yes
No
Any Access Requested?
No
Return
Successful
Is JFN Already OPEN?
Yes
ERR-OPNX1
No
Requesting Wild Card Output ?
Yes
ERR-DESX7
No
Store Bytesize in FILBYT (0 Implies A Byte Size of a Word)
Is This Device Assignable?
Yes
Device Available?
No
ERR-OPNX7
Yes
No
Is Mode Requested Legal ?
No
ERR-OPNX14
Yes
Store Mode in FILSTS
A
A
Setup Dispatch Table Adr (1)
OPEND(DEV'DTB)
Dispatch and Execute Dev. Dep. Code
Successful ?
No
Error
Yes
Init JFN Blk Variables
1 → FILLFW(LH)
Flags → FILSTS
Return
Successful Return


. PMAP
Will Cnt of Pgs Left to Map Cross any Pg Tble (XB) Boundary ?
Yes
PMAP0
Call This Routine for Pgs up to the Pg Tble Boundary
No
PMAP0
Call This Routine for All Pages
Return
PMAP0
Asking to Map File to File ?
Yes
ERR-PMAPX2
No
Is PMAP's Requested Access Consistent with OPENF Access ?
No
ERR-PMAPX1
Must Have Write Access to Dest.
Yes
Is Destination a File ?
No
PMAP2
Yes
Are All File Pgs Private (1) ?
No
Yes
Deleting Pages ?
Yes
No
Are All Fork (Source) Pgs Private ?
No
ERR-PMAPX2
Yes
PMAP5
Destination Is a Fork
PMAP2
Is Source a File ?
Yes
Increment Cnt of Pgs Mapped in
No
PMAP6:
Are Any Dest. Pgs of the Fork Currently Mapped from a File ?
Yes
Reduce Map Count in JFN Blk (FILLFW)
PMAP5
No
MSETPT
Set up Pg Tables Release Old Contents of Dest. & Have New Contents Be Set With Copy-on-Write Access
Did User Ask for Preload ?
Yes
PREPG
Request Swap in of Pgs
No
Return

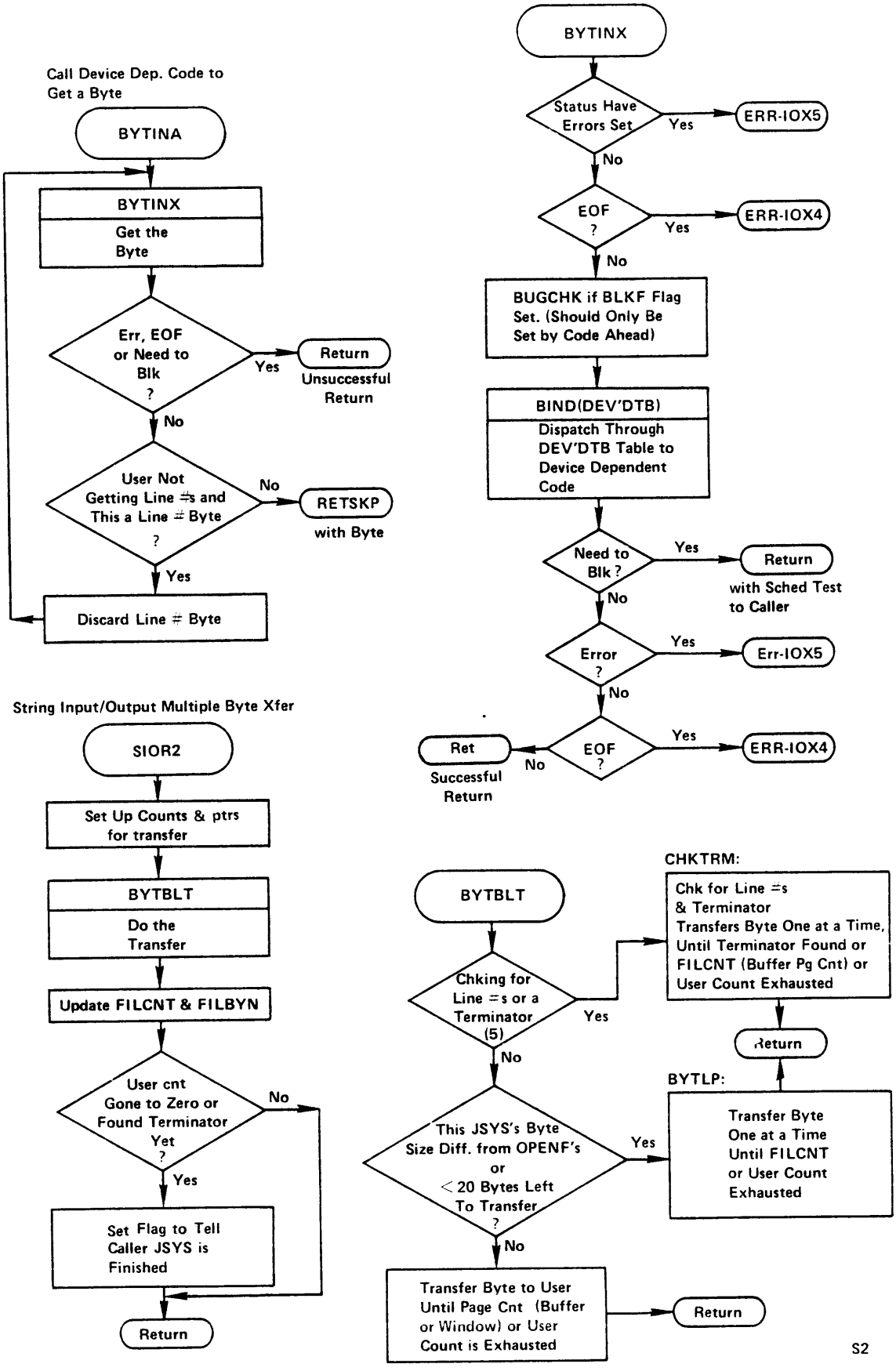
PM1


.SIN
SIN/SINR JSYS
.SINR
Set Flag to Say SIN JSYS
If This is Special Byte Ptr (-1 in LH) Create Equivalent Regular Byte Ptr. (1)
Set Flag to Say SINR JSYS
SIN0
CHKJFN
Chk JFN Validity & Set Up Status Flags
JFN OK ?
No
Error
Yes
TTY or Dev Null ?
Yes
SINTTY pS1a
No
SINBYT:
Byte Ptr (2) ?
Yes
SIOR2 pS2
Setup & Do Byte Blk Xfer (4)
Update Byte Ptrs in User's ACs
No
JFN OPENed For Read?
No
ERR-IOX1
SIN3:
CNT Gone to Zero
No
SIN00
Yes
SIN4
Yes
JFNID(DEV'DTB)
Init JFN for input
Tack Null Byte to String
Return to User
FILCNT>0 (Any Bytes Left in This Buffer Pg) (3) ?
No
SIN1 pS1a
Yes
SIOR2 pS2
Setup & Do Byte Blk Xfer (4)
(Return to User)
Yes
UNLCKF
Unlock File
Rec. Size Longer Than User's Request
ERR-IOX10
Yes
Satisfied User Request for Bytes ?
No
Return to User
Yes
Update Byte Ptrs in User ACS
No
SINR JSYS ?
Yes
Read All Bytes from Buffer ?
No
Satisfied User's Request for Bytes ?
No
S1

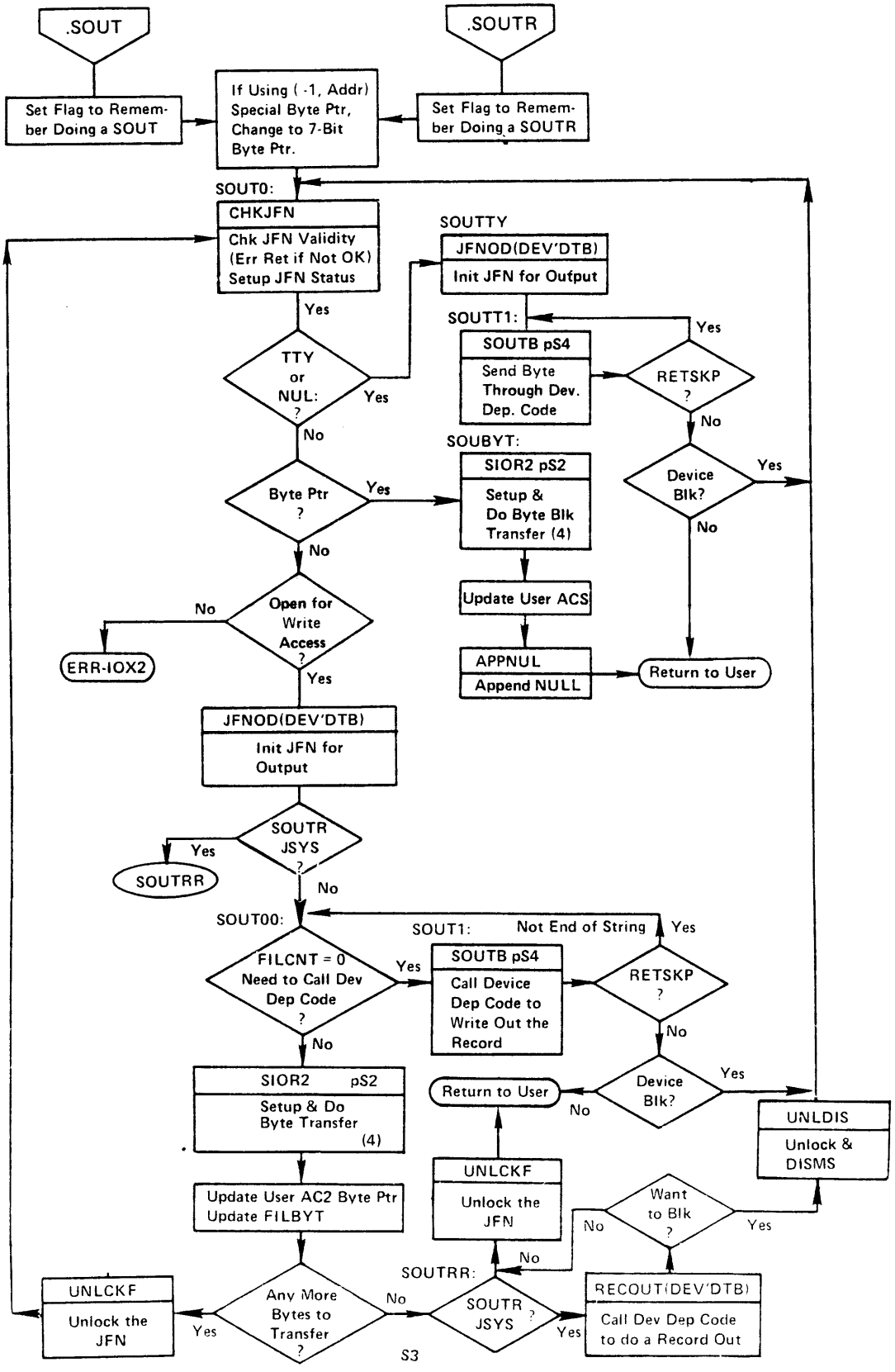
SIN/SINR JSYS (continued)

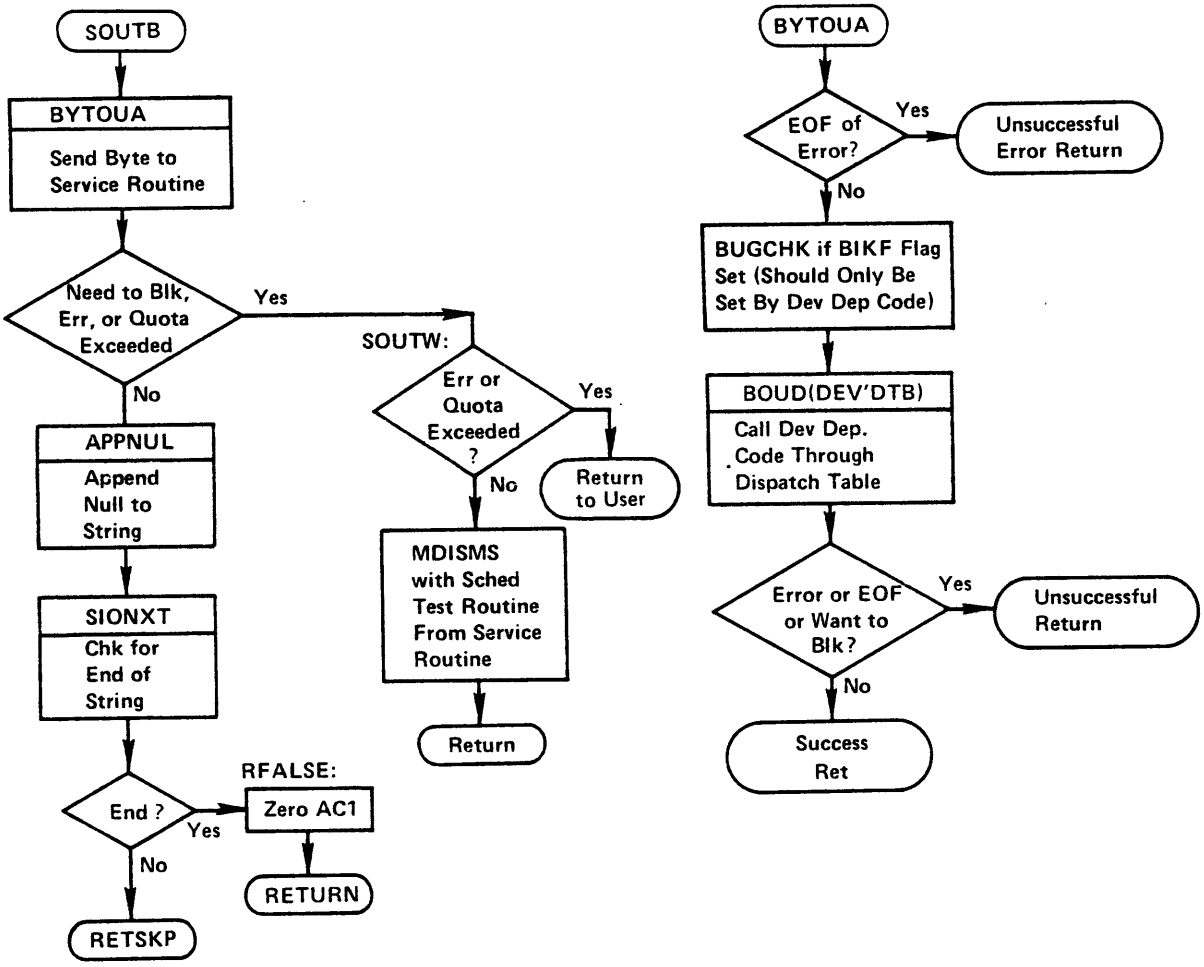


S1a

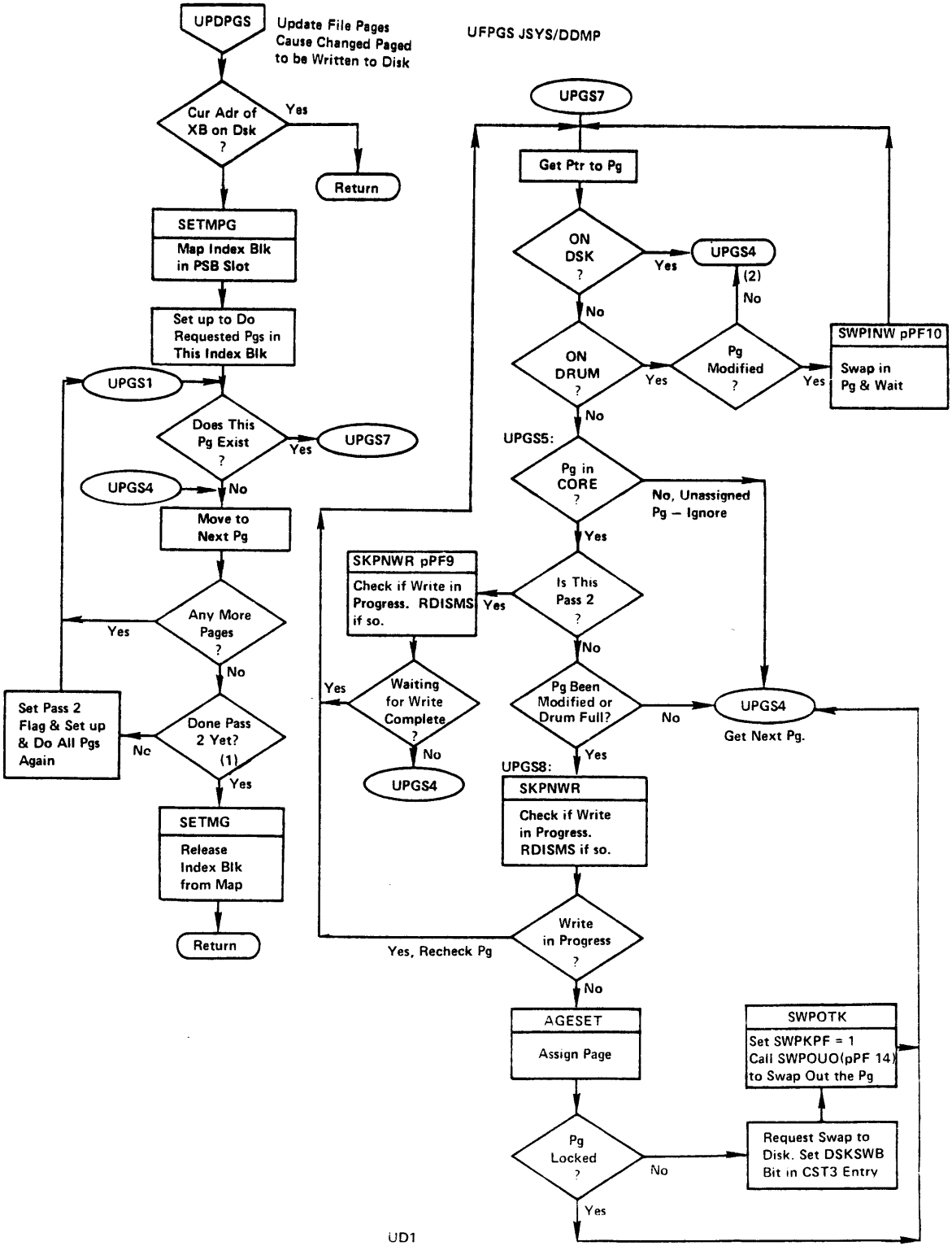

Call Device Dep. Code to Get a Byte
BYTINA
BYTINX
Get the Byte
Err, EOF or Need to Blk ?
Yes
Return
Unsuccessful Return
No
User Not Getting Line #s and This a Line # Byte ?
No
RETSKP
with Byte
Yes
Discard Line # Byte
String Input/Output Multiple Byte Xfer
SIOR2
Set Up Counts & ptrs for transfer
BYTBLT
Do the Transfer
Update FILCNT & FILBYN
User cnt Gone to Zero or Found Terminator Yet ?
No
Yes
Set Flag to Tell Caller JSYS is Finished
Return
BYTINX
Status Have Errors Set
Yes
ERR-IOX5
No
EOF ?
Yes
ERR-IOX4
No
BUGCHK if BLKF Flag Set. (Should Only Be Set by Code Ahead)
BIND(DEV'DTB)
Dispatch Through DEV'DTB Table to Device Dependent Code
Need to Blk ?
Yes
Return
with Sched Test to Caller
No
Error ?
Yes
Err-IOX5
No
EOF ?
Yes
ERR-IOX4
No
Ret
Successful Return
BYTBLT
Chking for Line #s or a Terminator (5)
Yes
CHKTRM:
Chk for Line #s & Terminator
Transfers Byte One at a Time, Until Terminator Found or FILCNT (Buffer Pg Cnt) or User Count Exhausted
Return
No
BYTLP:
This JSYS's Byte Size Diff. from OPENF's or < 20 Bytes Left To Transfer ?
Yes
Transfer Byte One at a Time Until FILCNT or User Count Exhausted
No
Transfer Byte to User Until Page Cnt (Buffer or Window) or User Count is Exhausted
Return
S2

SOUT/SOUTR JSYS

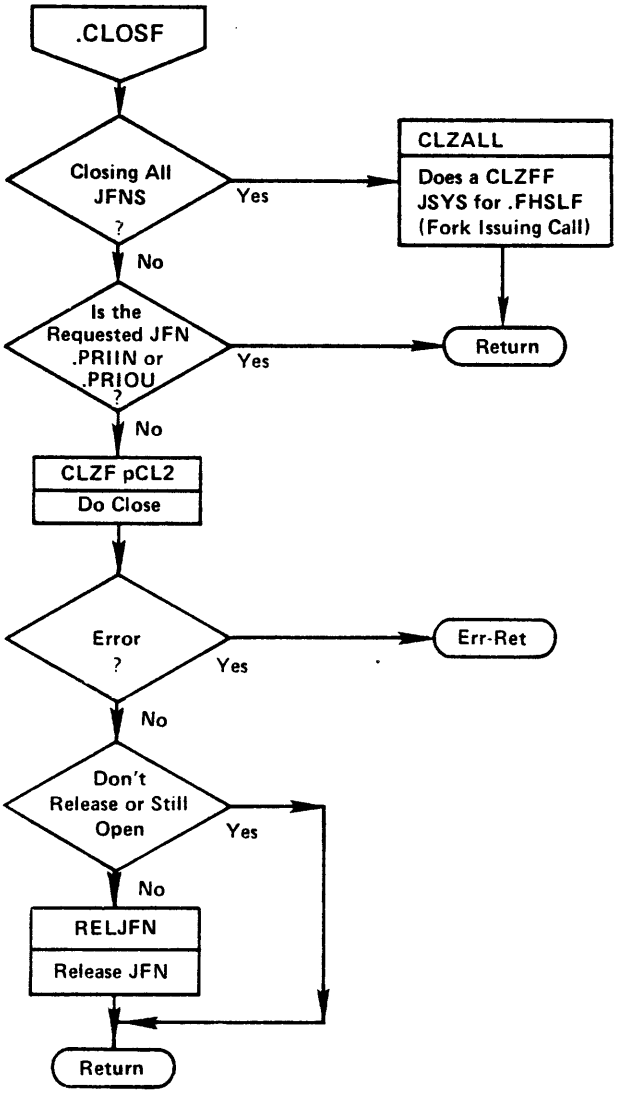



SOUTB
BYTOUA
Send Byte to Service Routine
Need to Blk, Err, or Quota Exceeded
Yes
No
SOUTW:
Err or Quota Exceeded ?
Yes
Return to User
No
MDISMS with Sched Test Routine From Service Routine
Return
APPNUL
Append Null to String
SIONXT
Chk for End of String
End ?
Yes
RFALSE:
Zero AC1
RETURN
No
RETSKP
BYTOUA
EOF of Error?
Yes
Unsuccessful Error Return
No
BUGCHK if BIKF Flag Set (Should Only Be Set By Dev Dep Code)
BOUD(DEV'DTB)
Call Dev Dep. Code Through Dispatch Table
Error or EOF or Want to Blk?
Yes
Unsuccessful Return
No
Success Ret

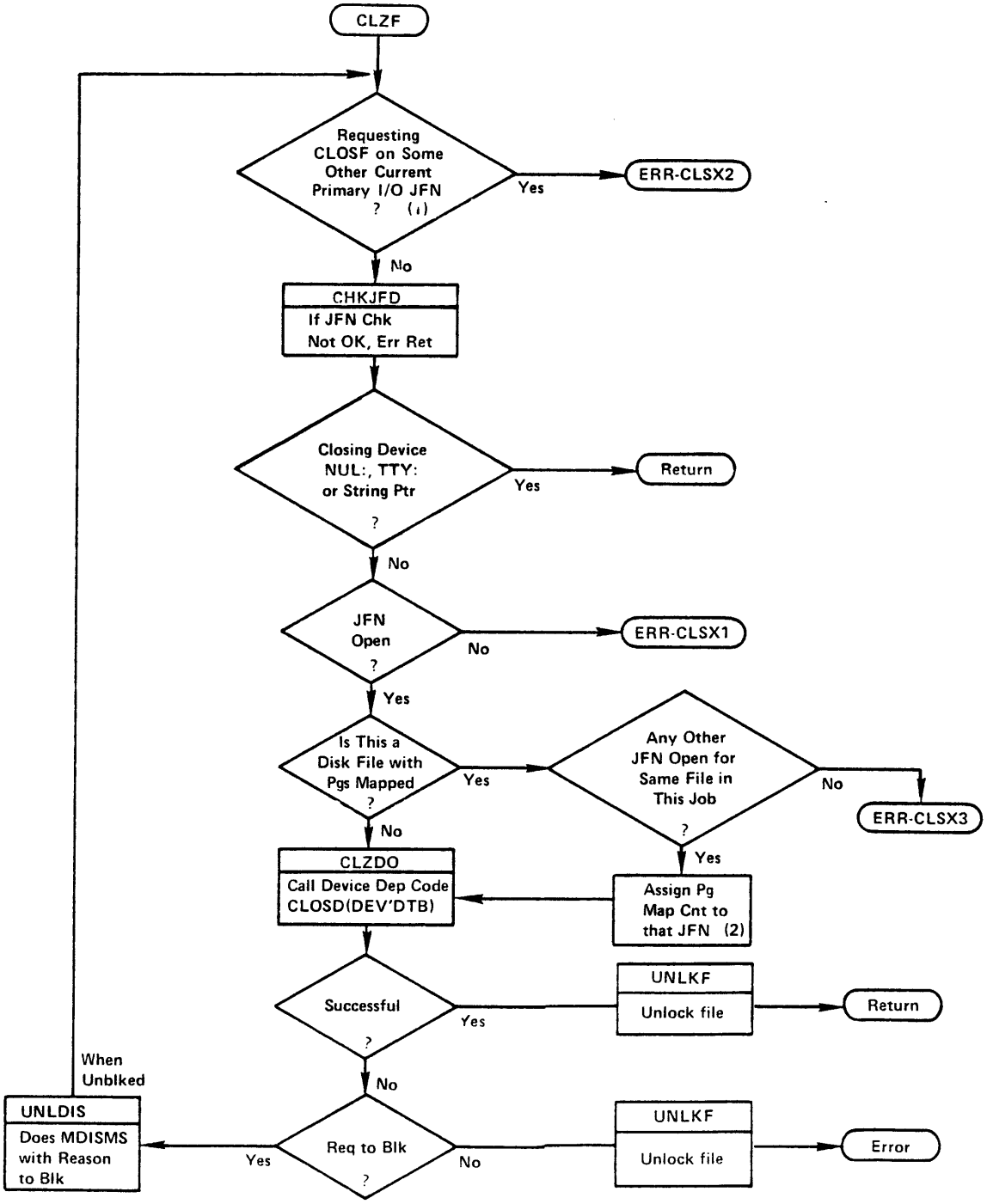
S4


UPDPGS
Update File Pages Cause Changed Paged to be Written to Disk
Cur Adr of XB on Dsk ?
Yes
Return
SETMPG
Map Index Blk in PSB Slot
Set up to Do Requested Pgs in This Index Blk
UPGS1
Does This Pg Exist ?
Yes
UPGS7
UPGS4
No
Move to Next Pg
Any More Pages ?
Yes
No
Set Pass 2 Flag & Set up & Do All Pgs Again
Done Pass 2 Yet? (1)
No
Yes
SETMG
Release Index Blk from Map
Return
UFPGS JSYS/DDMP
UPGS7
Get Ptr to Pg
ON DSK ?
Yes
UPGS4
(2)
No
No
ON DRUM ?
Yes
Pg Modified ?
Yes
SWPINW pPF10
Swap in Pg & Wait
No
UPGS5:
Pg in CORE ?
No, Unassigned Pg – Ignore
Yes
SKPNWR pPF9
Check if Write in Progress. RDISMS if so.
Yes
Is This Pass 2 ?
No
Yes
Waiting for Write Complete ?
No
UPGS4
Pg Been Modified or Drum Full?
No
UPGS4
Get Next Pg.
Yes
UPGS8:
SKPNWR
Check if Write in Progress. RDISMS if so.
Write in Progress ?
Yes, Recheck Pg
No
AGESET
Assign Page
Pg Locked ?
No
Request Swap to Disk. Set DSKSWB Bit in CST3 Entry
SWPOTK
Set SWPKPF = 1
Call SWPOUO(pPF 14) to Swap Out the Pg
Yes
UD1

CLOSF JSYS



CL1


CLZF
Requesting CLOSF on Some Other Current Primary I/O JFN ? (1)
Yes
ERR-CLSX2
No
CHKJFD
If JFN Chk Not OK, Err Ret
Closing Device NUL:, TTY: or String Ptr ?
Yes
Return
No
JFN Open ?
No
ERR-CLSX1
Yes
Is This a Disk File with Pgs Mapped ?
Yes
Any Other JFN Open for Same File in This Job ?
No
ERR-CLSX3
No
Yes
CLZDO
Call Device Dep Code CLOSD(DEV'DTB)
Assign Pg Map Cnt to that JFN (2)
Successful ?
Yes
UNLKF
Unlock file
Return
No
When Unblked
UNLDIS
Does MDISMS with Reason to Blk
Yes
Req to Blk ?
No
UNLKF
Unlock file
Error

CL2

GTJFN Comments

(1) This code is looking for a file specification of the form:

Dev:Directory Name,type,gen;T(temporary);P(protection);A(account)

One or more fields can be defined by logical names. If any fields are omitted from the specification, the system will default the values as follows:

Device	DSK:
Directory	Connected directory
Name	No default for disk Null for other devices
Generation	Highest existing for input Next highest for output
Protection	As specified for directory or protection of next lower generation
Account	Current user account

(2) The internal GTJFN code uses several locations in the JFN block as temporary cells. These locations have two names in the JFN block table descriptions. The JFN block storage locations set up or used by GTJFN are:

FILLCK*	FILDDN	FILNEN
FILTMP*	FILPRT	FILVER
FILACT*	FILSTS1	FILCOD (LH)
FILOPT*	FILLNM*	
FILDEV	FILDNM	

*Used internally only by the GTJFN JSYS.

(3) The creation process of the FDB simply asks for space in the directory for the FDB.

.OPENF Comments

(1) Cell FILDEV in the JFN blk has the device dispatch table address. For example, for disk, GTJFN sets the dispatch table address to DSKDTB. If spooling to disk, GTJFN sets the dispatch table address to SPLDTB, but the OPENF code changes the dispatch table address to DSKDTB and sets up a file specification in the JFN block.

(1) TOPS-20 allows a user to specify a special byte pointer of -1,, Address which is interpreted as a 7-bit byte size beginning on the word boundary, Address.

(2) A user can do I/O from one place to another in core by specifying byte pointers for both source and destination. This differs from BLT in that the use can transfer on non-word boundaries.

(3) For disk files, FILCNT will be the number of bytes remaining in the window page. For magtape and other devices it will be the number of bytes remaining in the current page of the buffer.

(4) The routine BYTBLT only moves data up to the page boundary of the current buffer page.

(5) If the user has not specified OF%PLN in the OPENF, line numbers are stripped off the beginning of each line. (See SIN JSYS in Monitor Calls manual for definition of terminator.)

.PMAP Comments

(1) A page is private if it is not shared between a file and a fork.

UPDPGS Comments

(1) Routine scans page table twice: first time to request writes on all changed pages. Second time to wait for completion of writes. (This is faster than waiting for each write to complete as it is requested.)

(2) If page has not been modified, a check is made to see if the drum is full and if so, to release this page back to the drum. The map pointer to the page will be changed to its disk address.

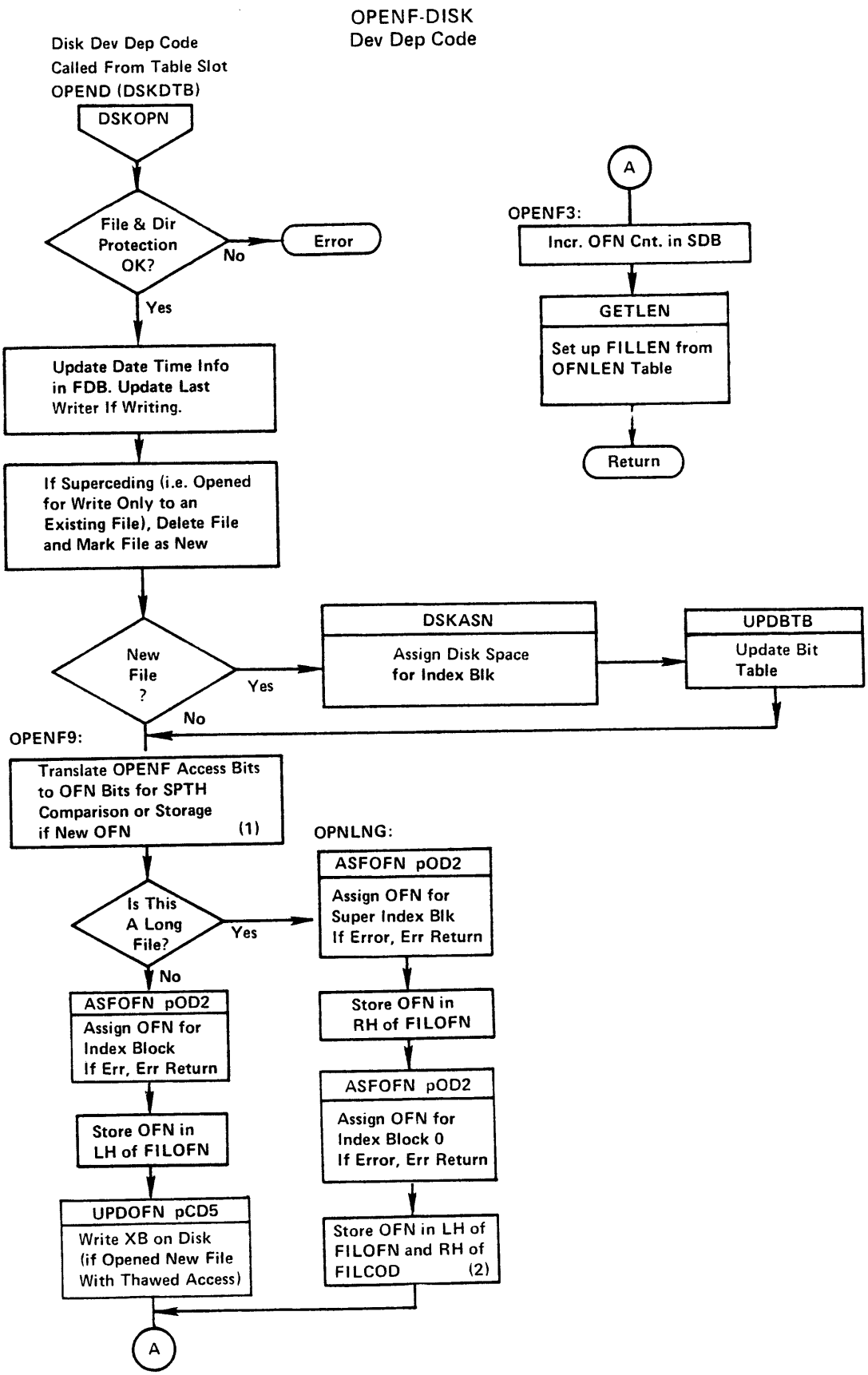
CLOSF Comments

(1) If user has switched primary I/O to some other JFN and attempts to close it, an error results.

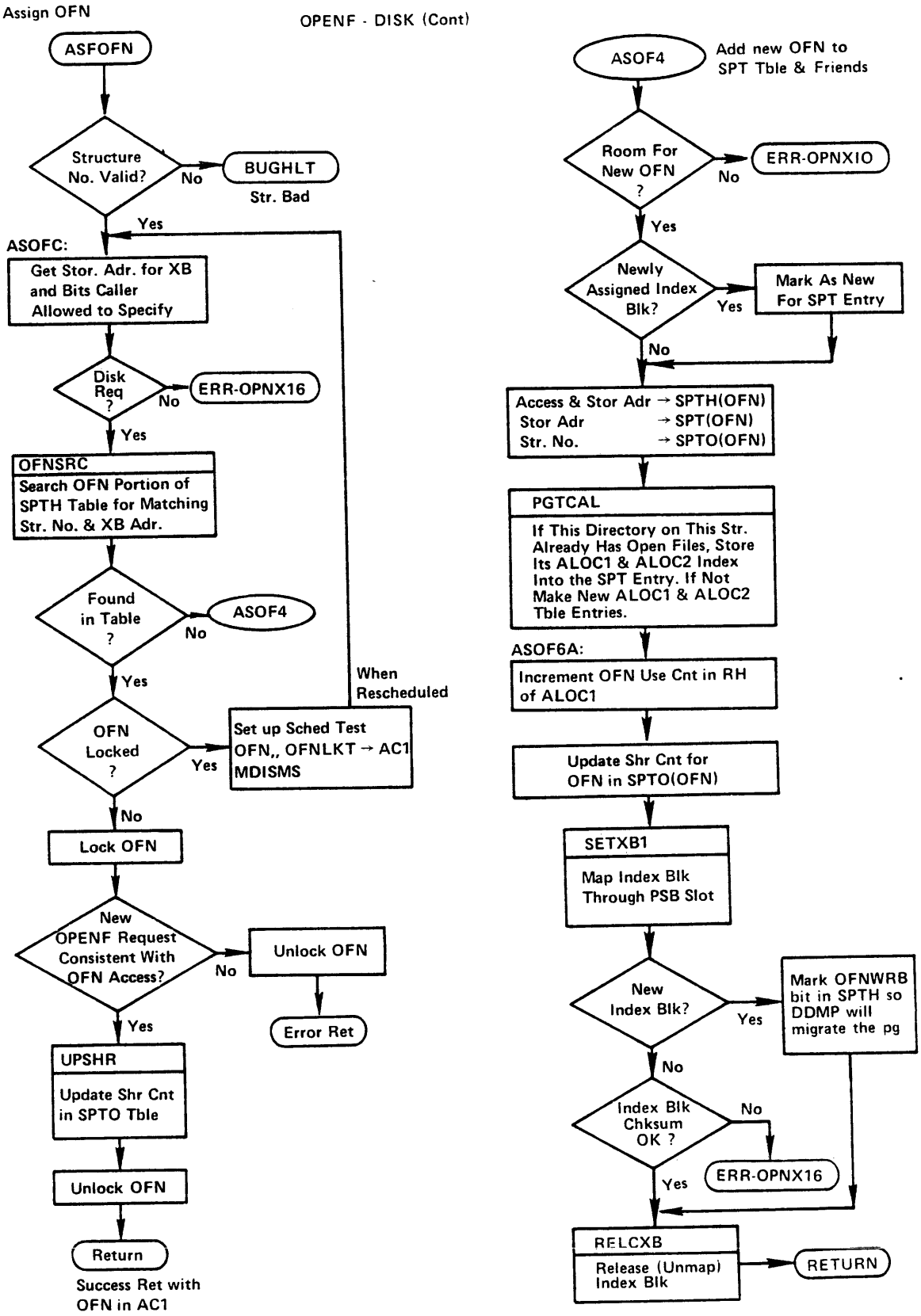
(2) The page map count in FILFW reflects the number of pages mapped and a CLOSF can't be done on a file if this count is greater than Ø .

JSYS CALL FLOWCHARTS

DSK DEPENDENT LEVEL

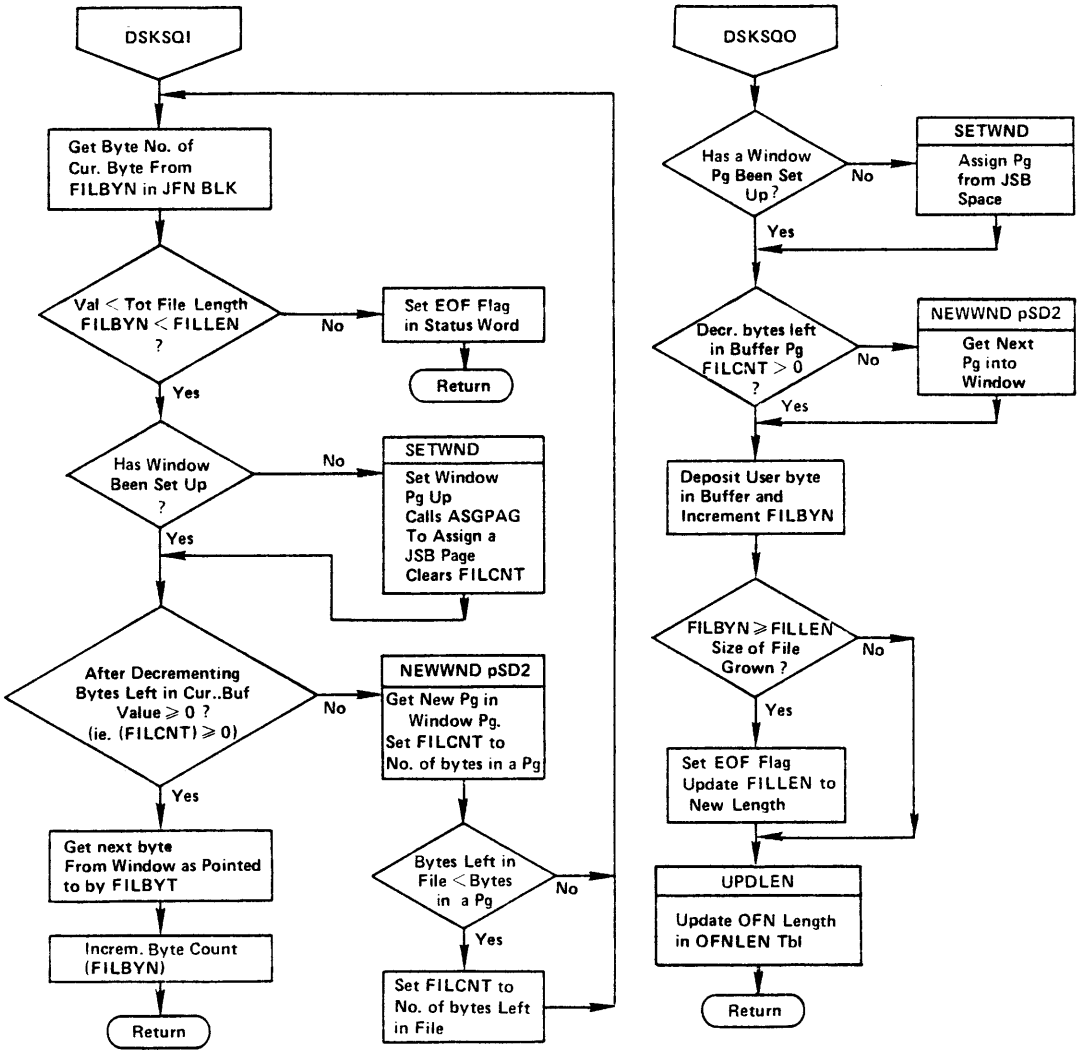

OPENF-DISK
Dev Dep Code
Disk Dev Dep Code
Called From Table Slot
OPEND (DSKDTB)
DSKOPN
File & Dir Protection OK?
No
Error
Yes
Update Date Time Info in FDB. Update Last Writer If Writing.
If Superceding (i.e. Opened for Write Only to an Existing File), Delete File and Mark File as New
New File ?
Yes
No
DSKASN
Assign Disk Space for Index Blk
UPDBTB
Update Bit Table
OPENF9:
Translate OPENF Access Bits to OFN Bits for SPTH Comparison or Storage if New OFN (1)
Is This A Long File?
Yes
No
OPNLNG:
ASFOFN pOD2
Assign OFN for Super Index Blk If Error, Err Return
Store OFN in RH of FILOFN
ASFOFN pOD2
Assign OFN for Index Block 0 If Error, Err Return
Store OFN in LH of FILOFN and RH of FILCOD (2)
ASFOFN pOD2
Assign OFN for Index Block If Err, Err Return
Store OFN in LH of FILOFN
UPDOFN pCD5
Write XB on Disk (if Opened New File With Thawed Access)
A
A
OPENF3:
Incr. OFN Cnt. in SDB
GETLEN
Set up FILLEN from OFNLEN Table
Return

OD1

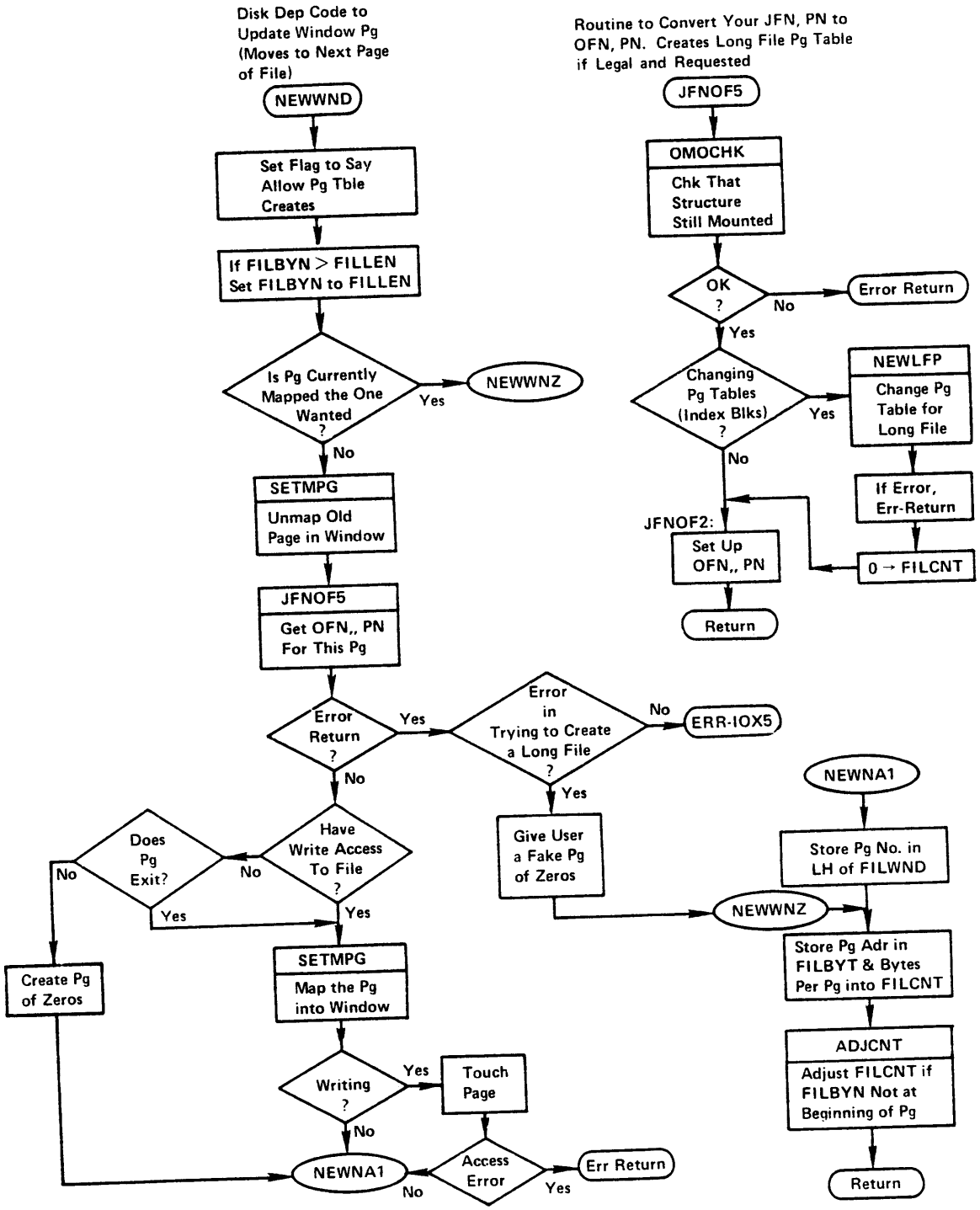

Assign OFN
OPENF - DISK (Cont)
ASFOFN
Structure No. Valid?
No
BUGHLT
Str. Bad
Yes
ASOFC:
Get Stor. Adr. for XB and Bits Caller Allowed to Specify
Disk Req ?
No
ERR-OPNX16
Yes
OFNSRC
Search OFN Portion of SPTH Table for Matching Str. No. & XB Adr.
Found in Table ?
No
ASOF4
Yes
When Rescheduled
OFN Locked ?
Yes
Set up Sched Test OFN,, OFNLKT → AC1 MDISMS
No
Lock OFN
New OPENF Request Consistent With OFN Access?
No
Unlock OFN
Error Ret
Yes
UPSHR
Update Shr Cnt in SPTO Tble
Unlock OFN
Return
Success Ret with OFN in AC1
ASOF4
Add new OFN to SPT Tble & Friends
Room For New OFN ?
No
ERR-OPNXIO
Yes
Newly Assigned Index Blk?
Yes
Mark As New For SPT Entry
No
Access & Stor Adr → SPTH(OFN)
Stor Adr → SPT(OFN)
Str. No. → SPTO(OFN)
PGTCAL
If This Directory on This Str. Already Has Open Files, Store Its ALOC1 & ALOC2 Index Into the SPT Entry. If Not Make New ALOC1 & ALOC2 Tble Entries.
ASOF6A:
Increment OFN Use Cnt in RH of ALOC1
Update Shr Cnt for OFN in SPTO(OFN)
SETXB1
Map Index Blk Through PSB Slot
New Index Blk?
Yes
Mark OFNWRB bit in SPTH so DDMP will migrate the pg
No
Index Blk Chksum OK ?
No
ERR-OPNX16
Yes
RELCXB
Release (Unmap) Index Blk
RETURN

OD2

SEQUENTIAL I/O-DSK
(String & Byte Dev Dep Code)

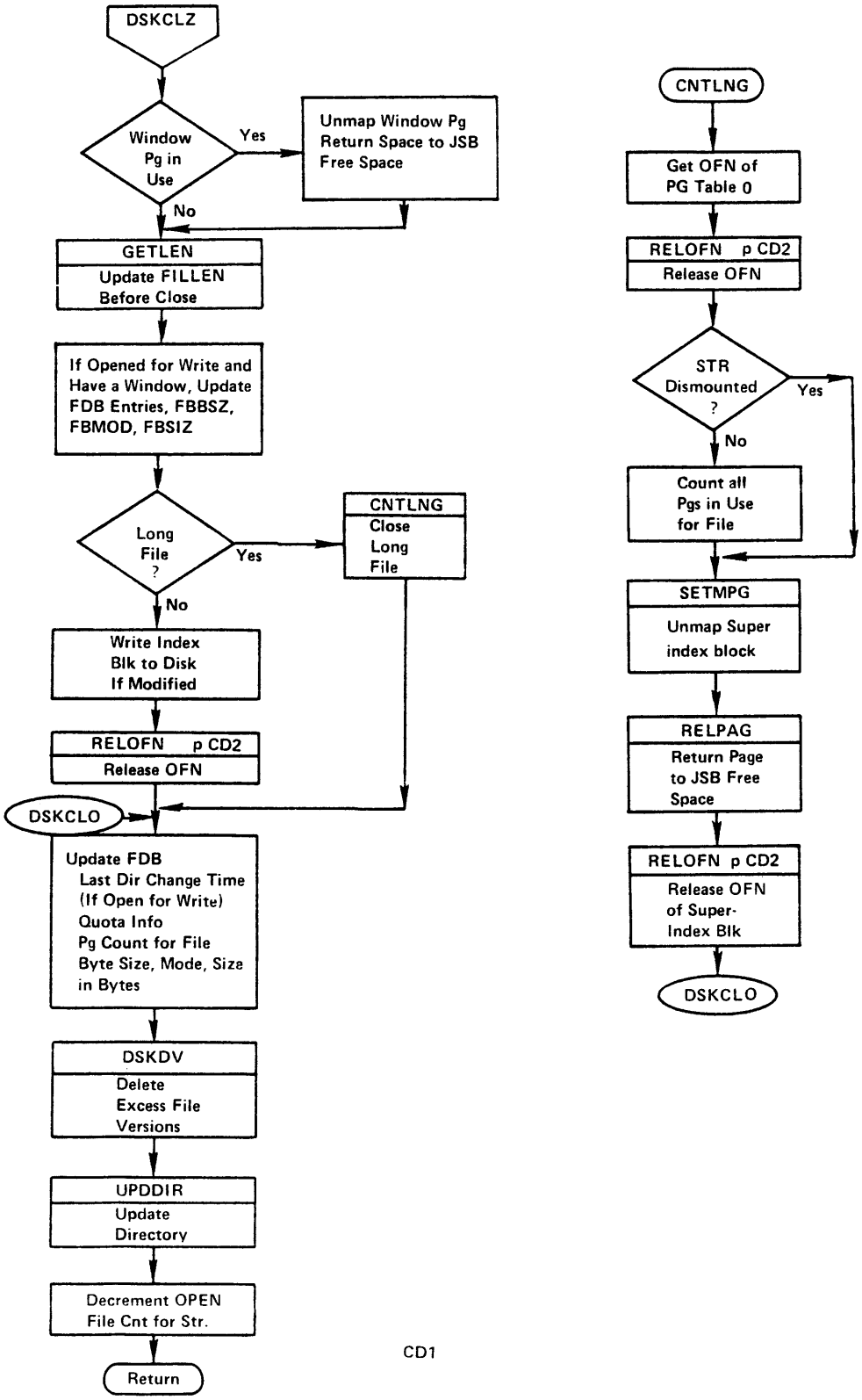


SD1


Disk Dep Code to Update Window Pg (Moves to Next Page of File)
NEWWND
Set Flag to Say Allow Pg Tble Creates
If FILBYN > FILLEN Set FILBYN to FILLEN
Is Pg Currently Mapped the One Wanted ?
Yes
NEWWNZ
No
SETMPG
Unmap Old Page in Window
JFNOF5
Get OFN,, PN For This Pg
Error Return ?
Yes
Error in Trying to Create a Long File ?
No
ERR-IOX5
Yes
No
Have Write Access To File ?
No
Does Pg Exit?
No
Yes
Yes
Create Pg of Zeros
SETMPG
Map the Pg into Window
Writing ?
Yes
Touch Page
No
NEWNA1
Access Error
Yes
Err Return
No
Routine to Convert Your JFN, PN to OFN, PN. Creates Long File Pg Table if Legal and Requested
JFNOF5
OMOCHK
Chk That Structure Still Mounted
OK ?
No
Error Return
Yes
Changing Pg Tables (Index Blks) ?
Yes
NEWLFP
Change Pg Table for Long File
If Error, Err-Return
No
JFNOF2:
Set Up OFN,, PN
0 → FILCNT
Return
Give User a Fake Pg of Zeros
NEWNA1
Store Pg No. in LH of FILWND
NEWWNZ
Store Pg Adr in FILBYT & Bytes Per Pg into FILCNT
ADJCNT
Adjust FILCNT if FILBYN Not at Beginning of Pg
Return

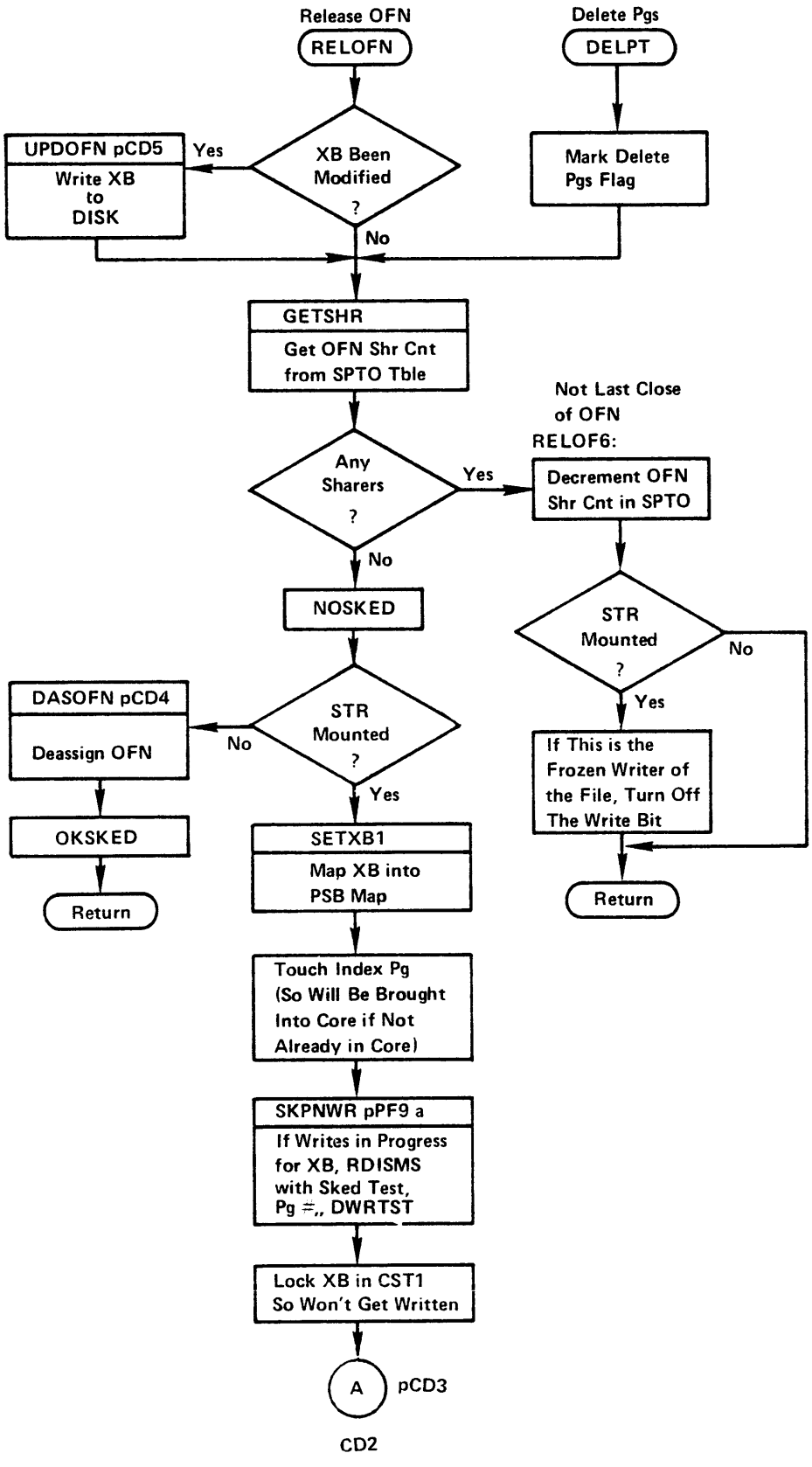
SD2

CLOSF – DSK
Dev. Dep. Code

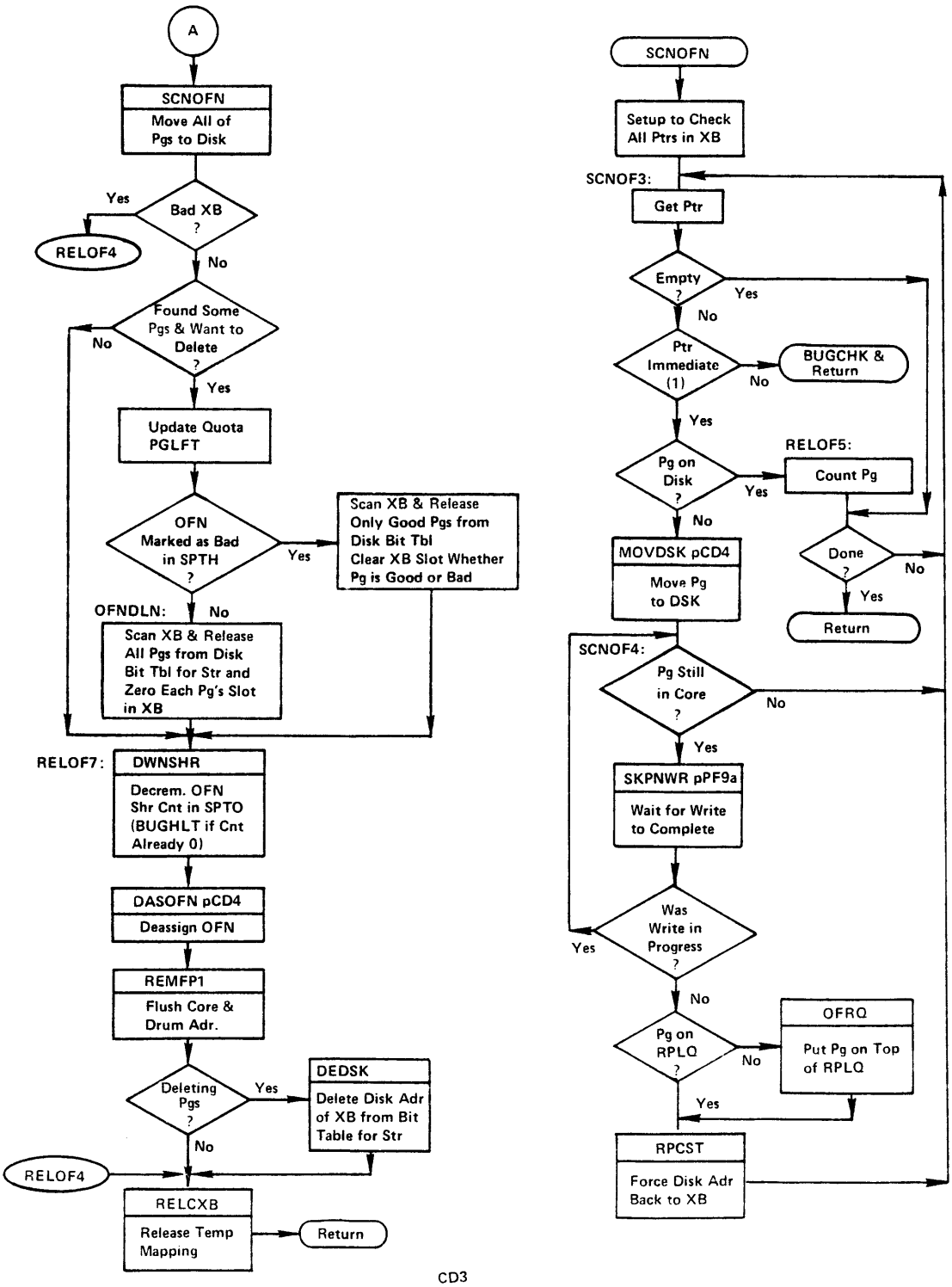


CD1

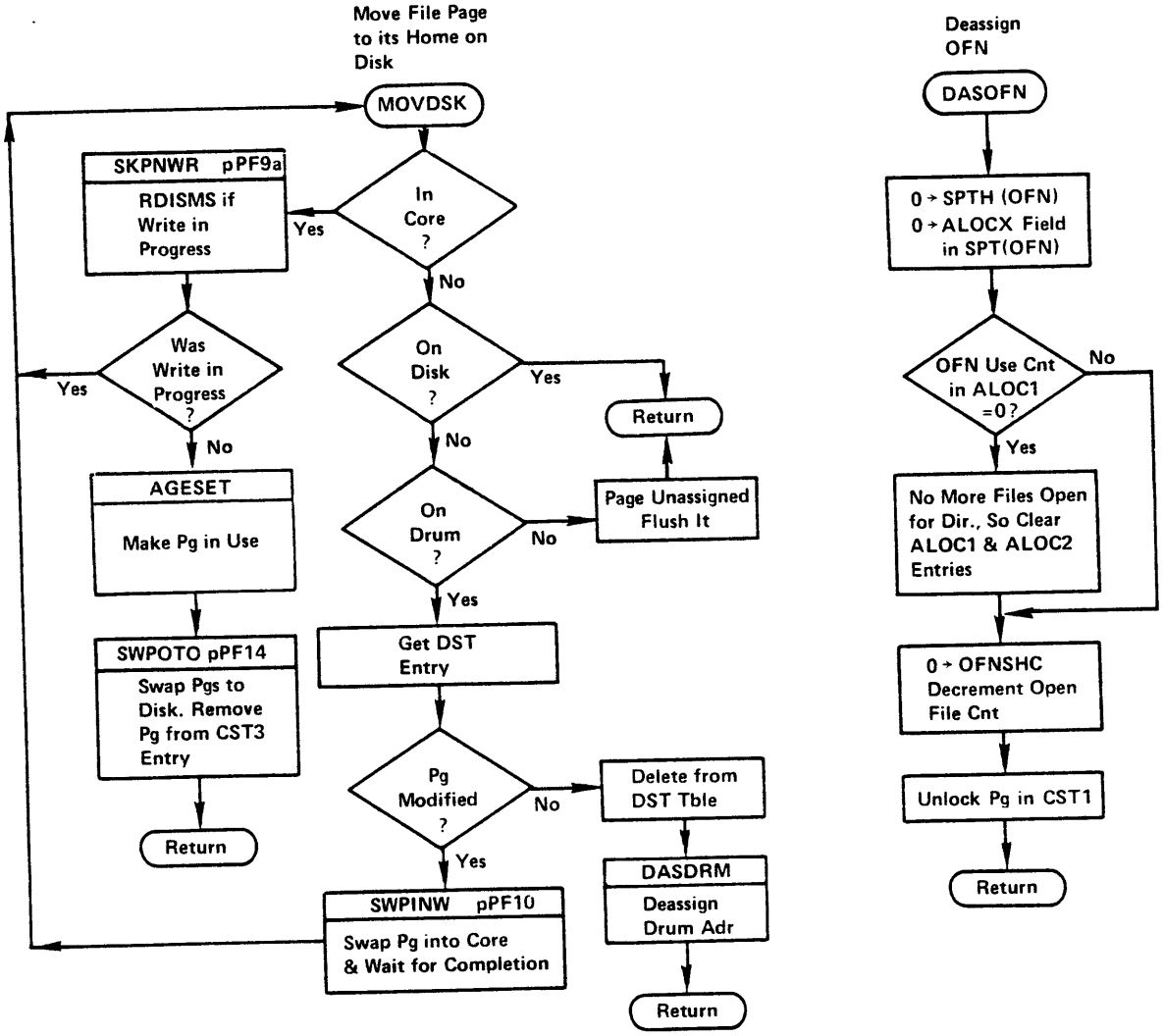
CLOSF-DSK (Continued)

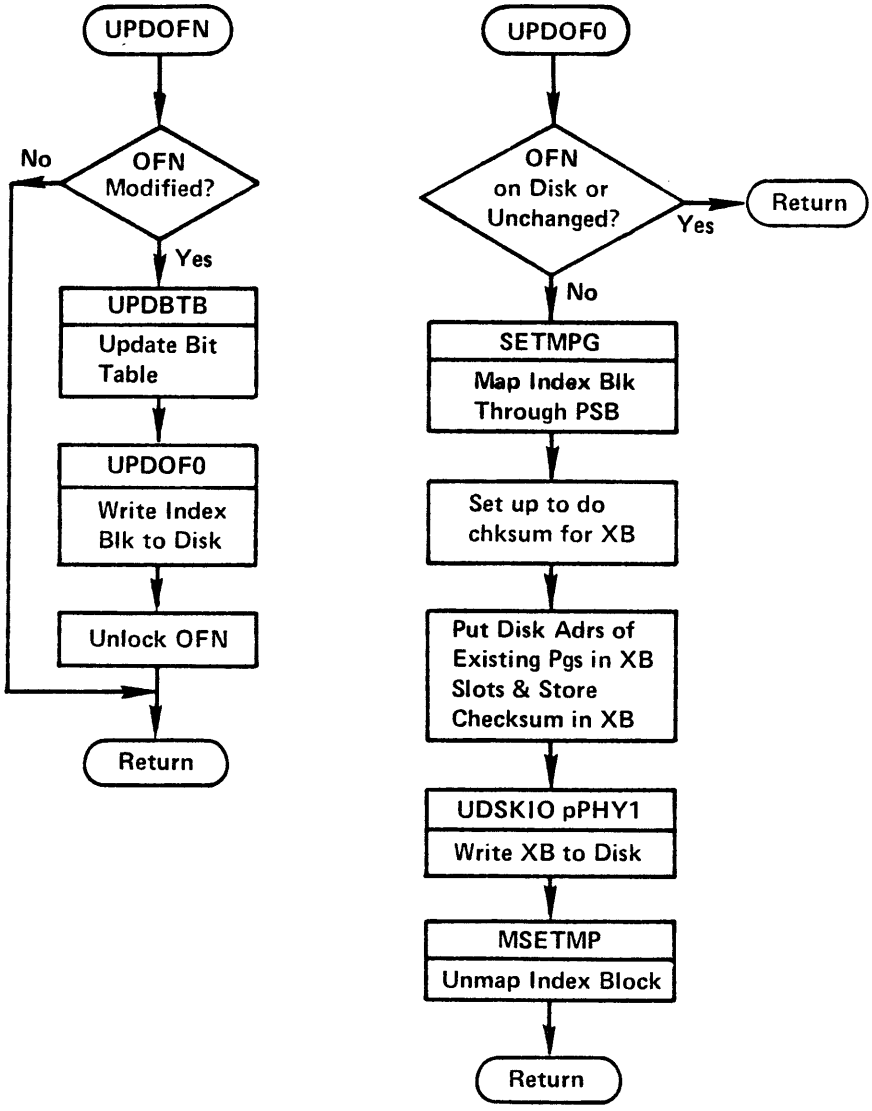


CLOSF-DSK (Continued)



CD3


Move File Page to its Home on Disk
MOVDSK
In Core ?
Yes
SKPNWR pPF9a
RDISMS if Write in Progress
No
On Disk ?
Yes
Return
No
On Drum ?
No
Page Unassigned Flush It
Yes
Get DST Entry
Pg Modified ?
No
Delete from DST Tble
Yes
DASDRM
Deassign Drum Adr
Return
SWPINW pPF10
Swap Pg into Core & Wait for Completion
Was Write in Progress ?
Yes
No
AGESET
Make Pg in Use
SWPOTO pPF14
Swap Pgs to Disk. Remove Pg from CST3 Entry
Return
Deassign OFN
DASOFN
0 → SPTH (OFN)
0 → ALOCX Field in SPT(OFN)
OFN Use Cnt in ALOC1 =0?
No
Yes
No More Files Open for Dir., So Clear ALOC1 & ALOC2 Entries
0 → OFNSHC
Decrement Open File Cnt
Unlock Pg in CST1
Return


UPDOFN
No
OFN Modified?
Yes
UPDBTB
Update Bit Table
UPDOF0
Write Index Blk to Disk
Unlock OFN
Return
UPDOF0
OFN on Disk or Unchanged?
Yes
Return
No
SETMPG
Map Index Blk Through PSB
Set up to do chksum for XB
Put Disk Adrs of Existing Pgs in XB Slots & Store Checksum in XB
UDSKIO pPHY1
Write XB to Disk
MSETMP
Unmap Index Block
Return

OPENF-DISK Comments

(1) OFN bits: 0=read, 10=write, 11=thawed, 01=restricted

(2) For a long file, the OFN of index block Ø is remembered in the JFN blk and used as the identity of the file by the ENQ/DEQ facility.

CLOSF-DISK Comments

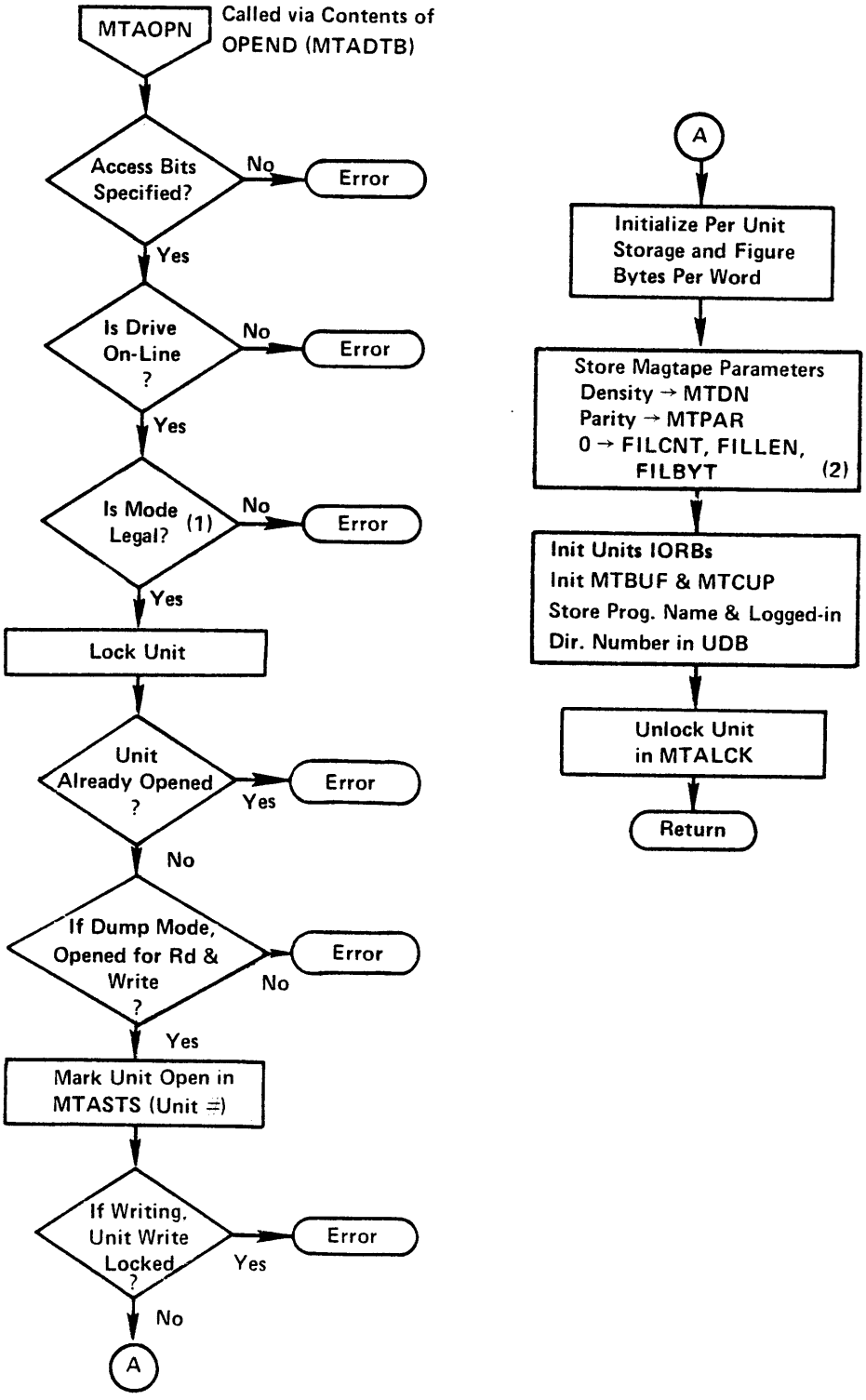
(1) All storage addresses placed in an index blk have the pointer type field set to immediate.

JSYS's CALLS

MTA DEPENDENT LEVEL

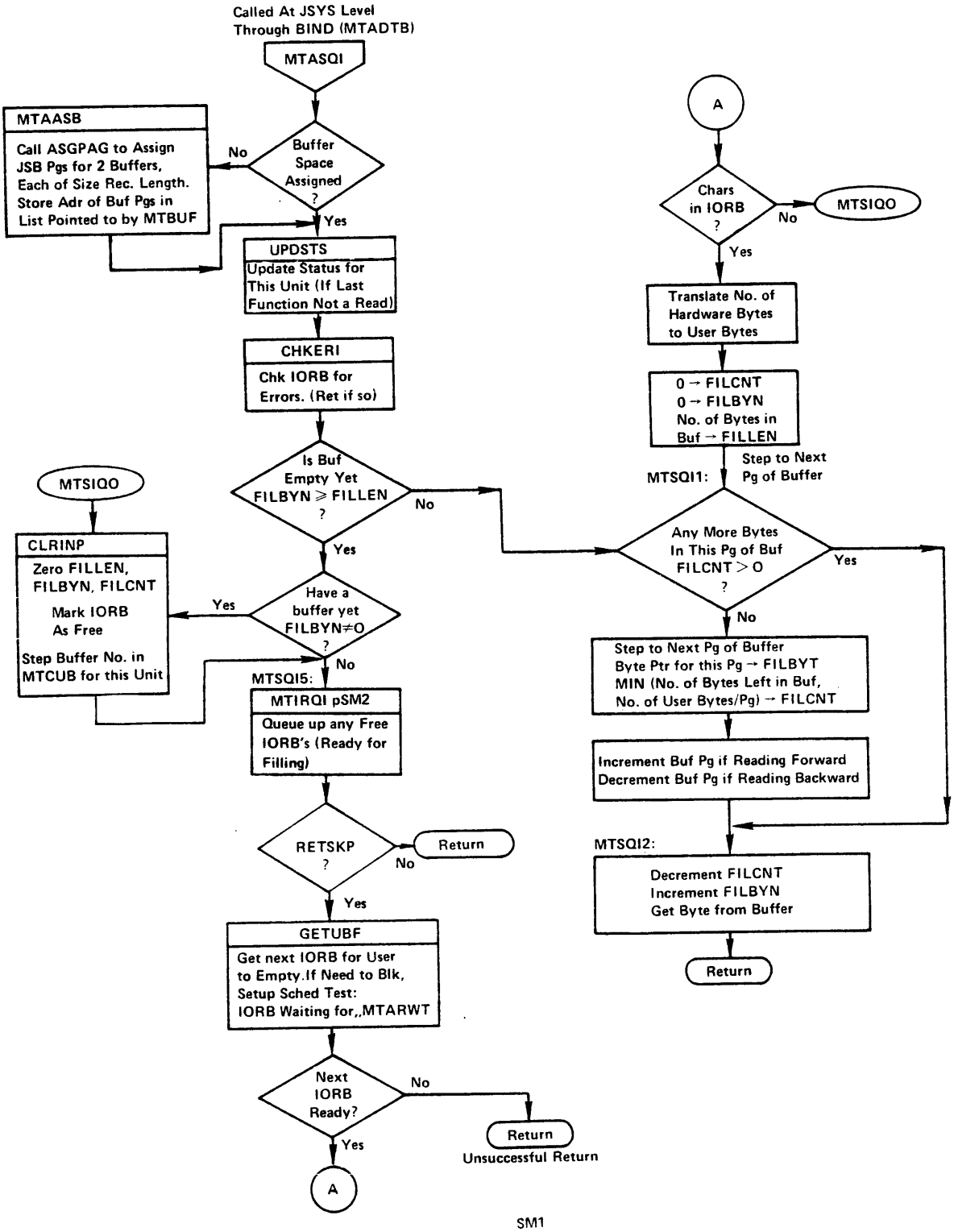
MTAOPN -	Magtape Opening of a File	OM1
MTASQI -	Magtape Sequential Input	SM1
	MTAIRQ - Queue Up Specified IORB	SM2
MTASQO -	Magtape Sequential Output	SM3
MTACLZ -	Magtape Closing of a File	CM1

OPENF – MAGTAPE
DEVICE DEPENDENT CODE

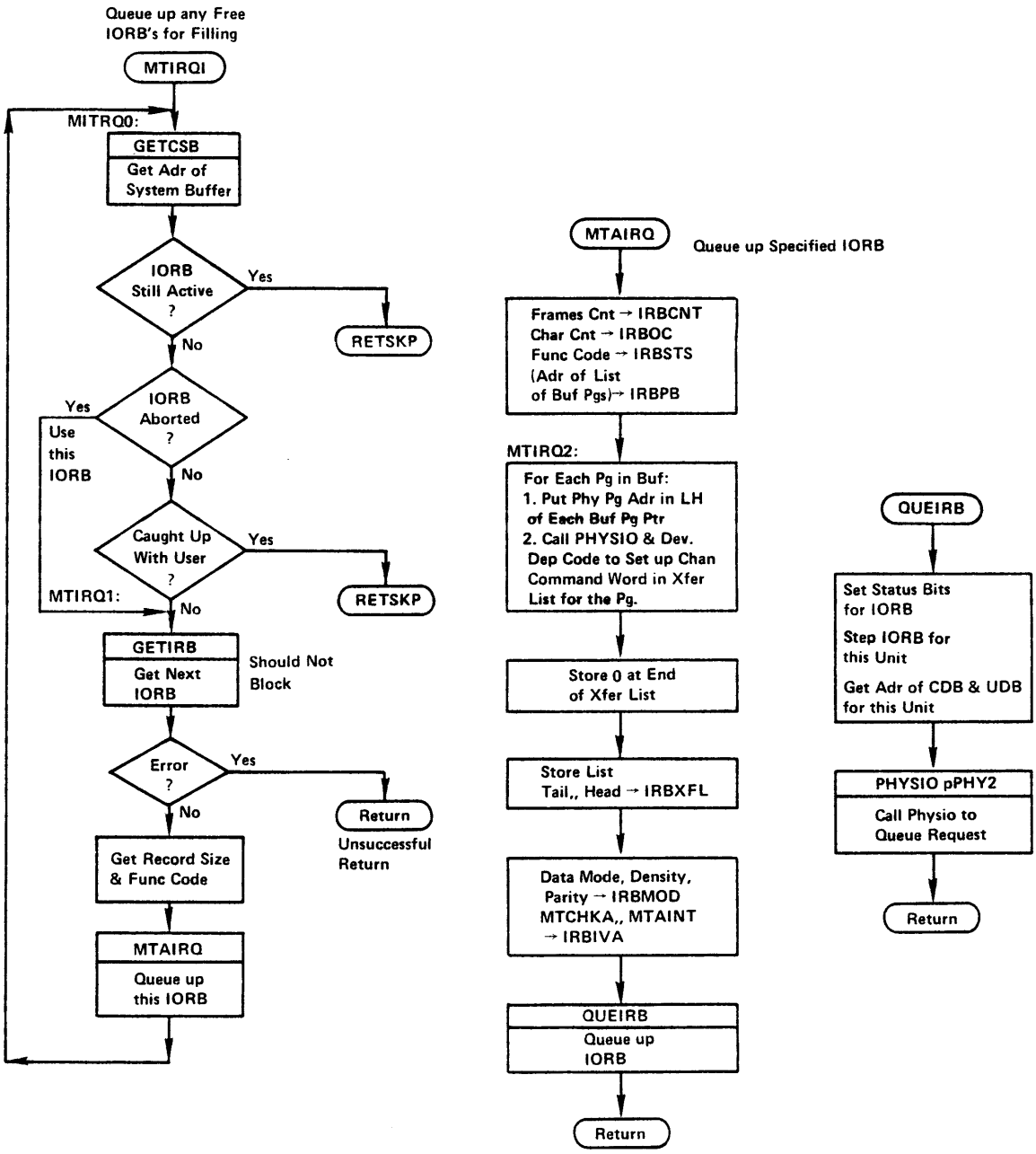


OM1

SEQUENTIAL INPUT – MTA
(STRING & BYTE DEV. DEP. CODE)

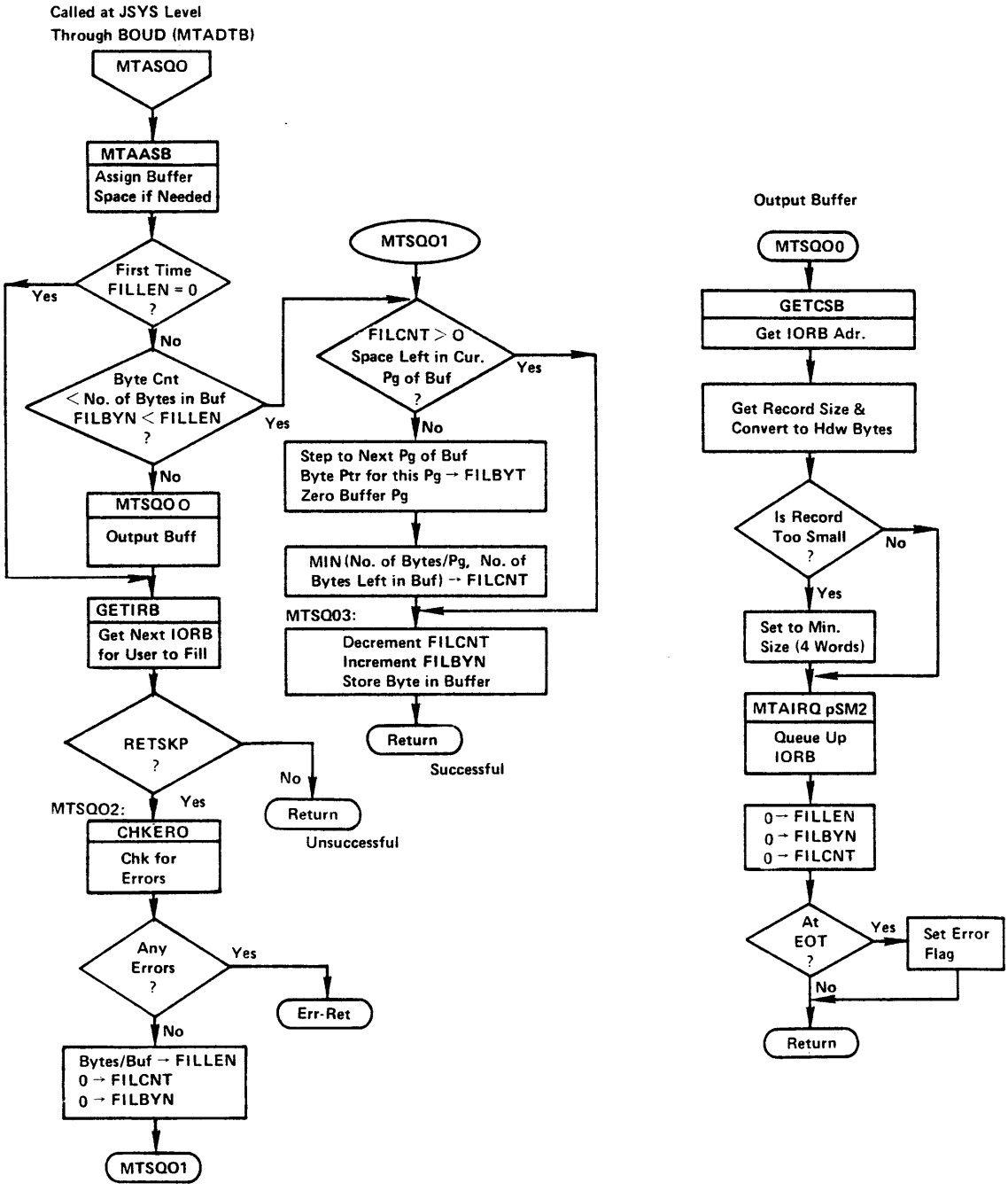


SM1

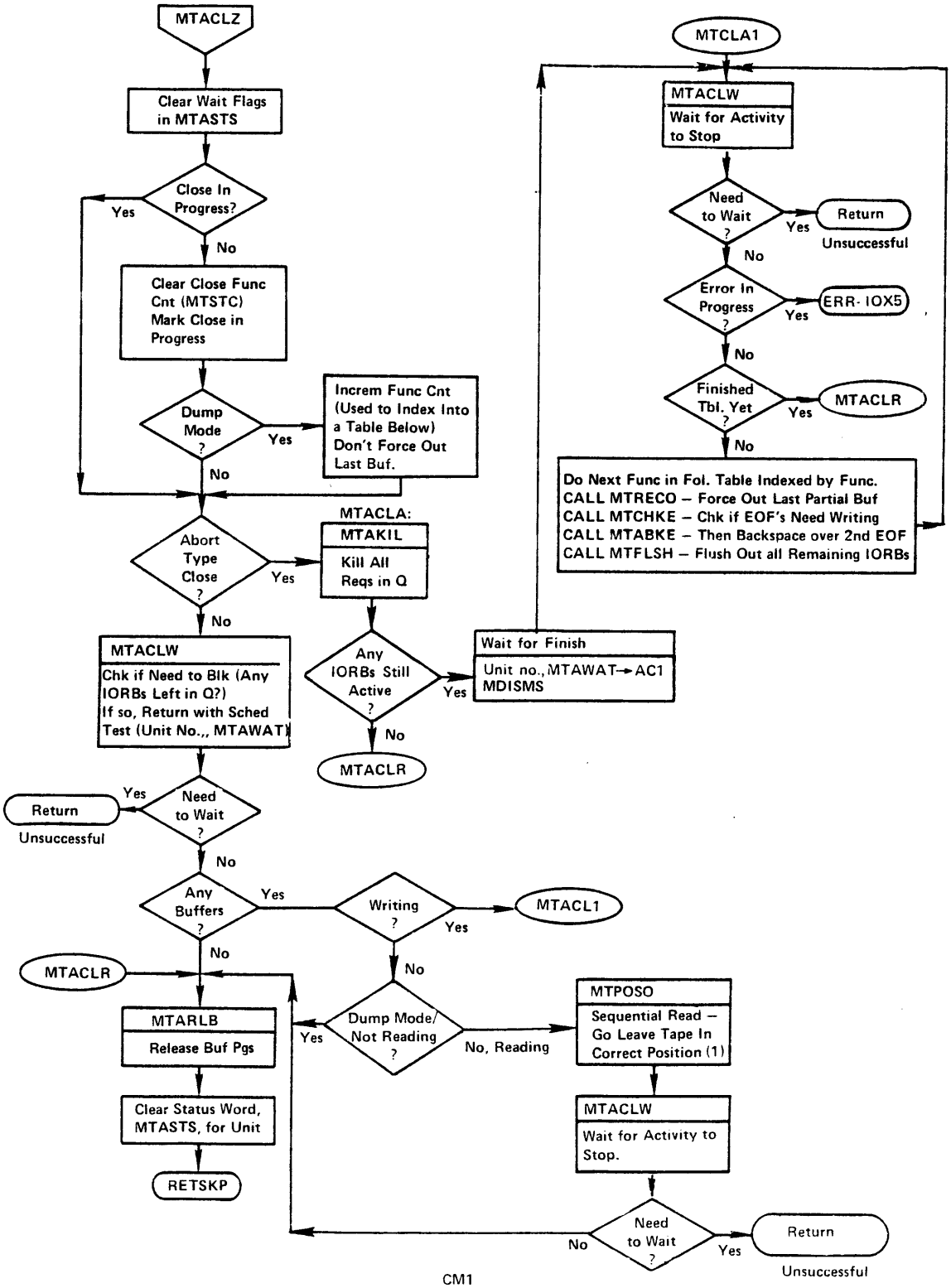

Queue up any Free IORB's for Filling
MTIRQI
MITRQ0:
GETCSB
Get Adr of System Buffer
IORB Still Active ?
Yes
No
RETSKP
IORB Aborted ?
Yes
Use this IORB
No
Caught Up With User ?
Yes
RETSKP
MTIRQ1:
No
GETIRB
Get Next IORB
Should Not Block
Error ?
Yes
No
Return
Unsuccessful Return
Get Record Size & Func Code
MTAIRQ
Queue up this IORB
MTAIRQ
Queue up Specified IORB
Frames Cnt → IRBCNT
Char Cnt → IRBOC
Func Code → IRBSTS
(Adr of List of Buf Pgs)→ IRBPB
MTIRQ2:
For Each Pg in Buf:
1. Put Phy Pg Adr in LH of Each Buf Pg Ptr
2. Call PHYSIO & Dev. Dep Code to Set up Chan Command Word in Xfer List for the Pg.
Store 0 at End of Xfer List
Store List Tail,, Head → IRBXFL
Data Mode, Density, Parity → IRBMOD
MTCHKA,, MTAINT → IRBIVA
QUEIRB
Queue up IORB
Return
QUEIRB
Set Status Bits for IORB
Step IORB for this Unit
Get Adr of CDB & UDB for this Unit
PHYSIO pPHY2
Call Physio to Queue Request
Return

SM2

SEQUENTIAL OUTPUT – MTA
STRING & BYTE DEV. DEP. CODE



CLOSF – MAGTAPE



CM1

OPENF-MAGTAPE Comments

(1) One can open for read and write only in dump mode.

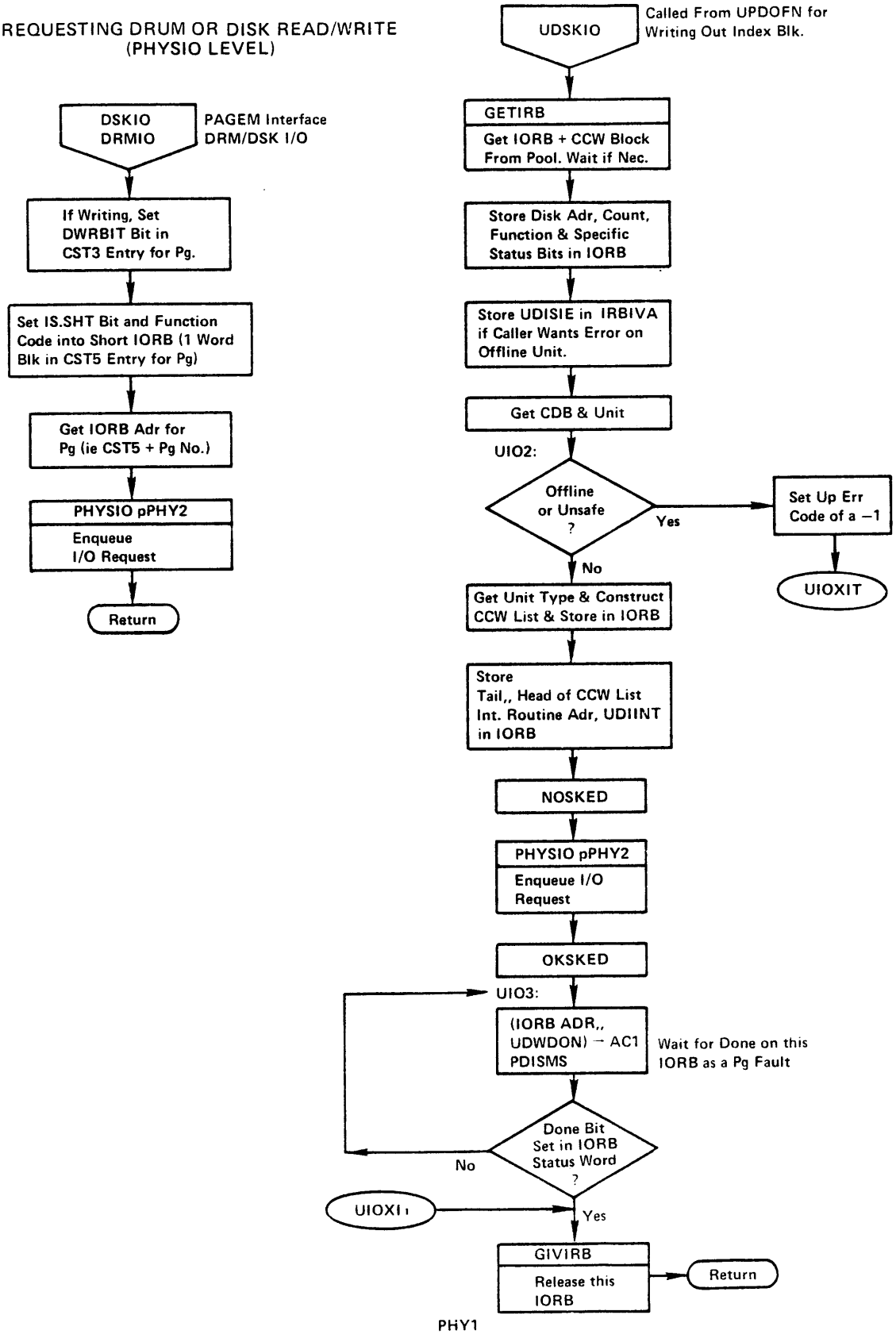
(2) FILCNT/Count of bytes left to use in current page of buffer.
FILLEN/Count of bytes in buffer.
FILBYN/Buffer byte number user is referencing.

CLOSF - MAGTAPE Comments

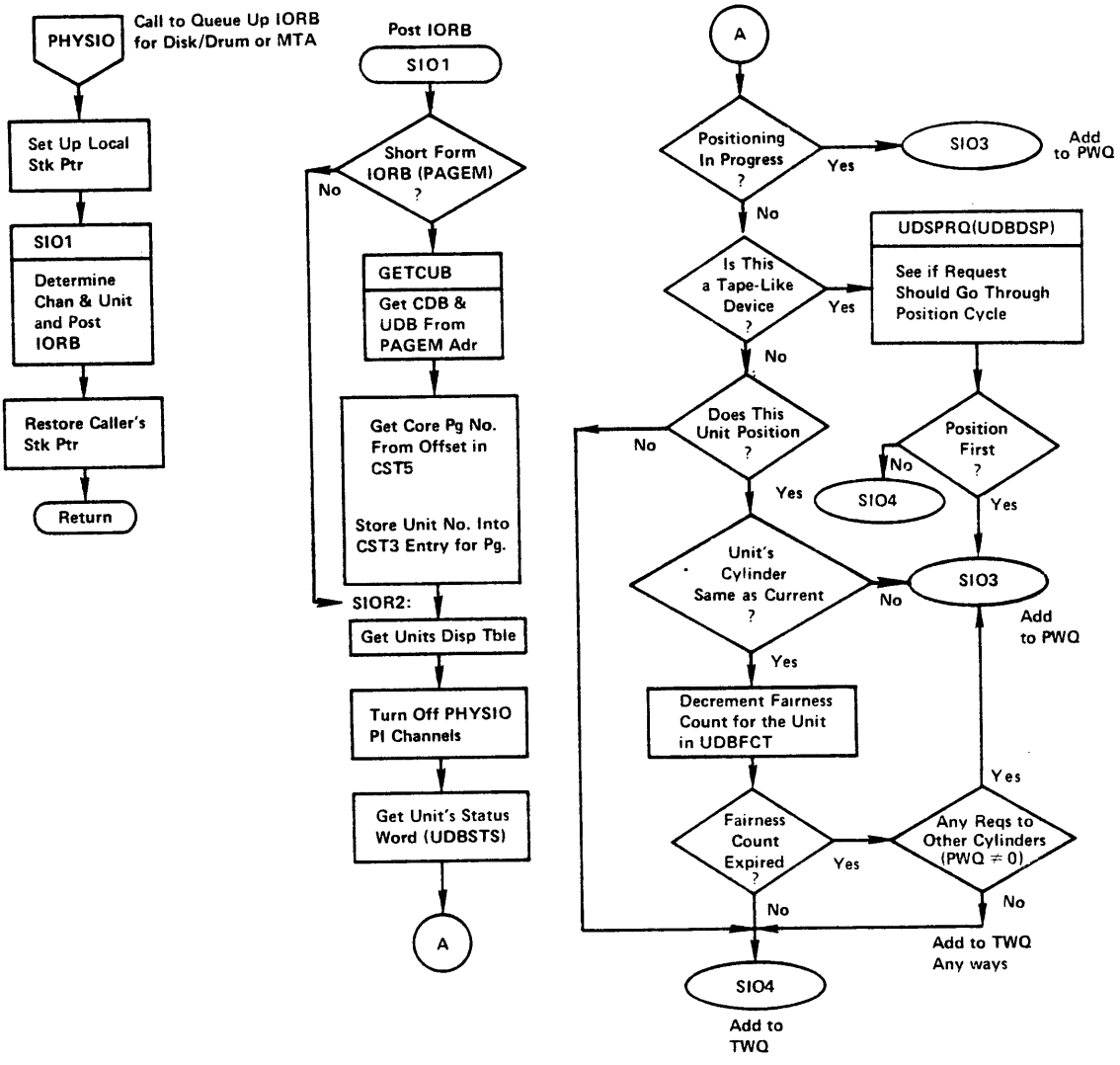
(1) Since the monitor reads ahead, backspacing to just after last user record read may be necessary.

REQUESTING DISK/MTA I/O & INTERRUPT HANDLING FLOWCHARTS
(PHYSIO LEVEL)

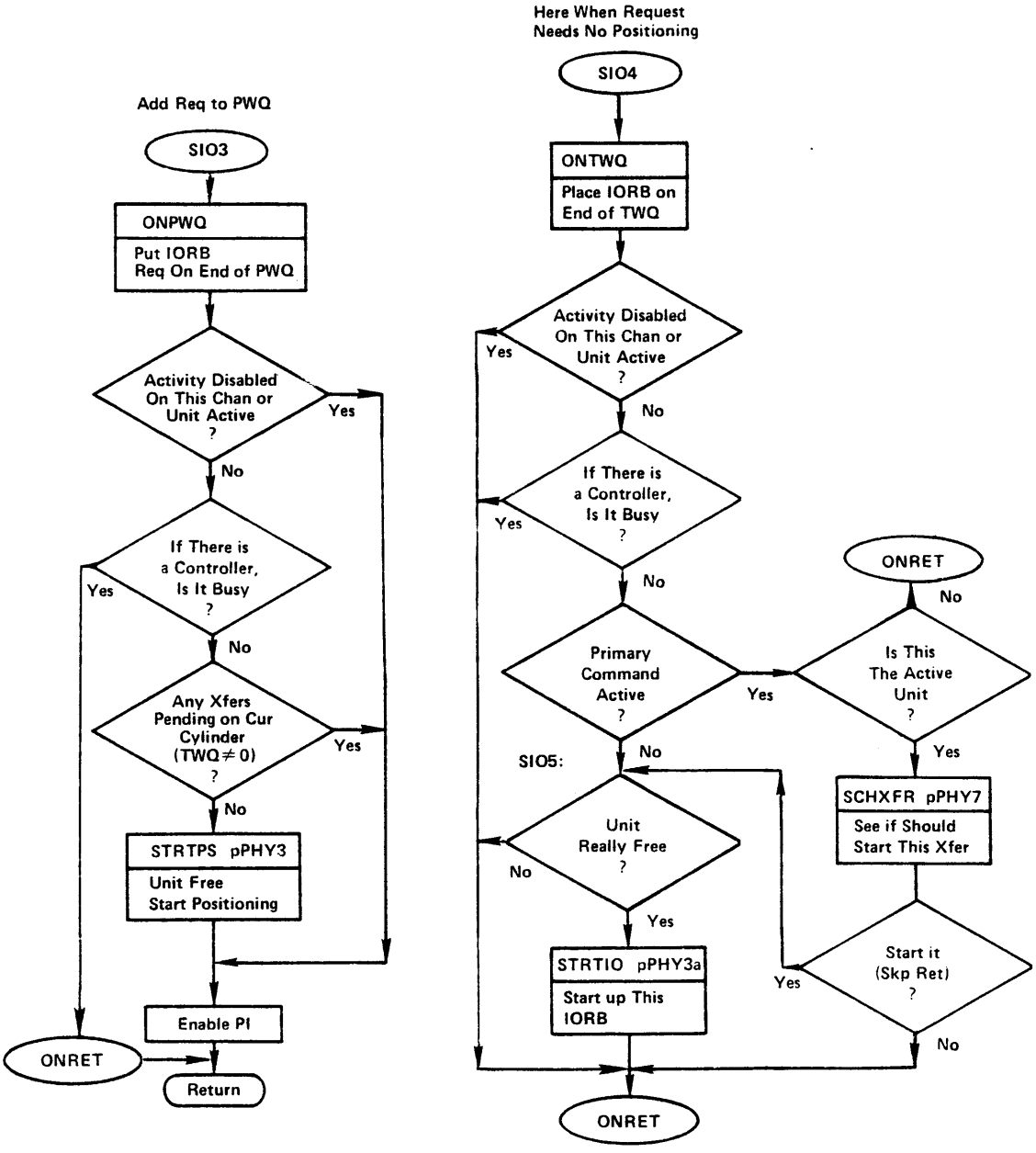
REQUESTING DRUM OR DISK READ/WRITE
(PHYSIO LEVEL)



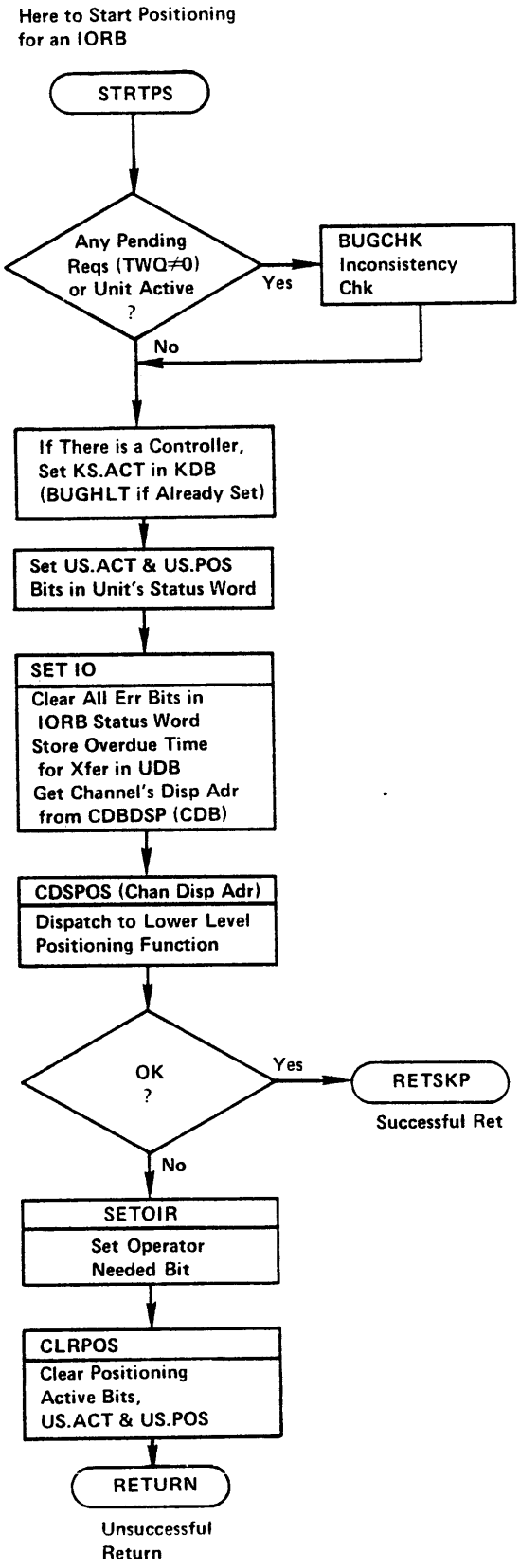
PHY1


PHYSIO
Call to Queue Up IORB for Disk/Drum or MTA
Set Up Local Stk Ptr
SIO1
Determine Chan & Unit and Post IORB
Restore Caller's Stk Ptr
Return
Post IORB
SIO1
Short Form IORB (PAGEM) ?
No
GETCUB
Get CDB & UDB From PAGEM Adr
Get Core Pg No. From Offset in CST5
Store Unit No. Into CST3 Entry for Pg.
SIOR2:
Get Units Disp Tble
Turn Off PHYSIO PI Channels
Get Unit's Status Word (UDBSTS)
A
A
Positioning In Progress ?
Yes
SIO3
Add to PWQ
No
Is This a Tape-Like Device ?
Yes
UDSPRQ(UDBDSP)
See if Request Should Go Through Position Cycle
No
Does This Unit Position ?
No
Yes
Position First ?
No
SIO4
Yes
Unit's Cylinder Same as Current ?
No
SIO3
Add to PWQ
Yes
Decrement Fairness Count for the Unit in UDBFCT
Fairness Count Expired ?
Yes
Any Reqs to Other Cylinders (PWQ ≠ 0)
Yes
No
No
Add to TWQ Any ways
SIO4
Add to TWQ

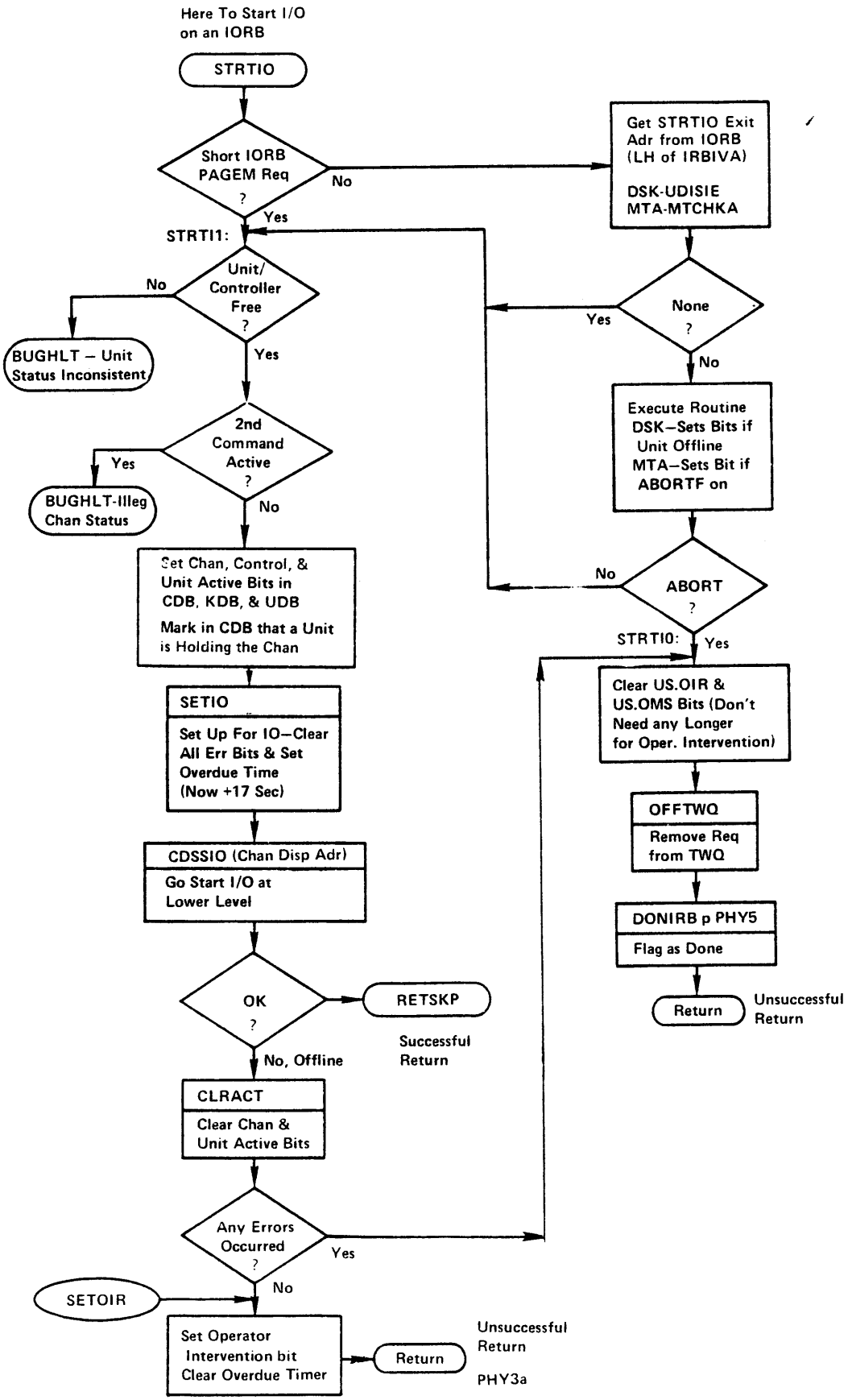
PHY2

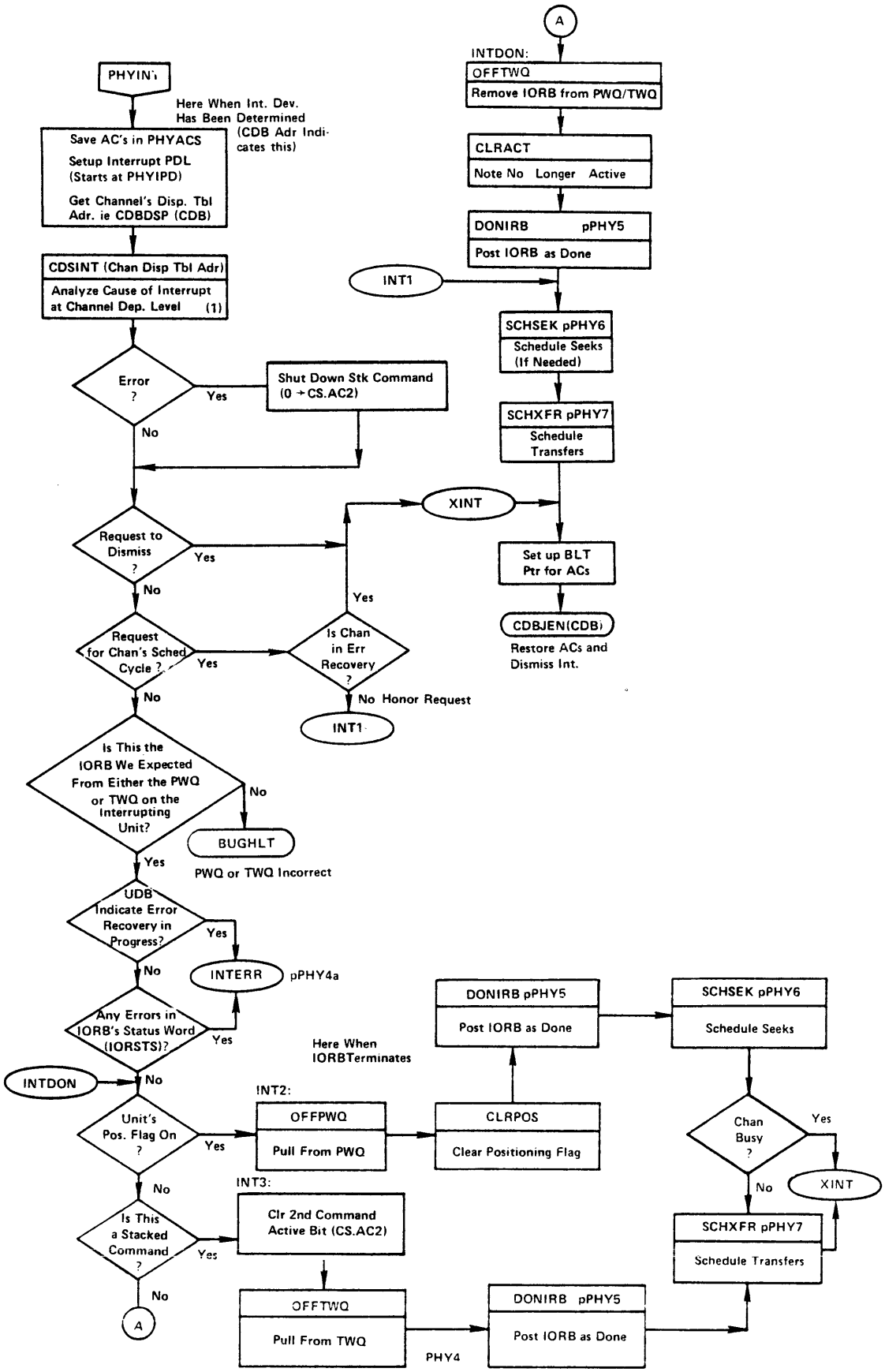

Add Req to PWQ
SIO3
ONPWQ
Put IORB
Req On End of PWQ
Activity Disabled
On This Chan or
Unit Active
?
Yes
No
If There is
a Controller,
Is It Busy
?
Yes
No
Any Xfers
Pending on Cur
Cylinder
(TWQ≠0)
?
Yes
No
STRTPS pPHY3
Unit Free
Start Positioning
Enable PI
ONRET
Return
Here When Request
Needs No Positioning
SIO4
ONTWQ
Place IORB on
End of TWQ
Activity Disabled
On This Chan or
Unit Active
?
Yes
No
If There is
a Controller,
Is It Busy
?
Yes
No
Primary
Command
Active
?
Yes
No
ONRET
No
Is This
The Active
Unit
?
Yes
SCHXFR pPHY7
See if Should
Start This Xfer
Start it
(Skp Ret)
?
Yes
No
SIO5:
Unit
Really Free
?
No
Yes
STRTIO pPHY3a
Start up This
IORB
ONRET

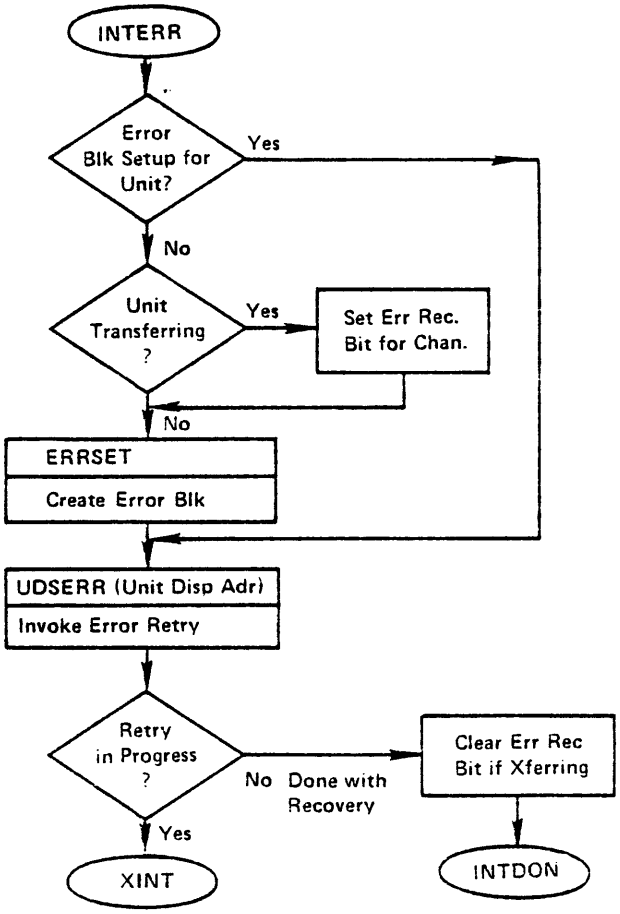
PHY2a


Here to Start Positioning for an IORB
STRTPS
Any Pending Reqs (TWQ≠0) or Unit Active ?
Yes
BUGCHK Inconsistency Chk
No
If There is a Controller, Set KS.ACT in KDB (BUGHLT if Already Set)
Set US.ACT & US.POS Bits in Unit's Status Word
SET IO
Clear All Err Bits in IORB Status Word
Store Overdue Time for Xfer in UDB
Get Channel's Disp Adr from CDBDSP (CDB)
CDSPOS (Chan Disp Adr)
Dispatch to Lower Level Positioning Function
OK ?
Yes
RETSKP
Successful Ret
No
SETOIR
Set Operator Needed Bit
CLRPOS
Clear Positioning Active Bits, US.ACT & US.POS
RETURN
Unsuccessful Return

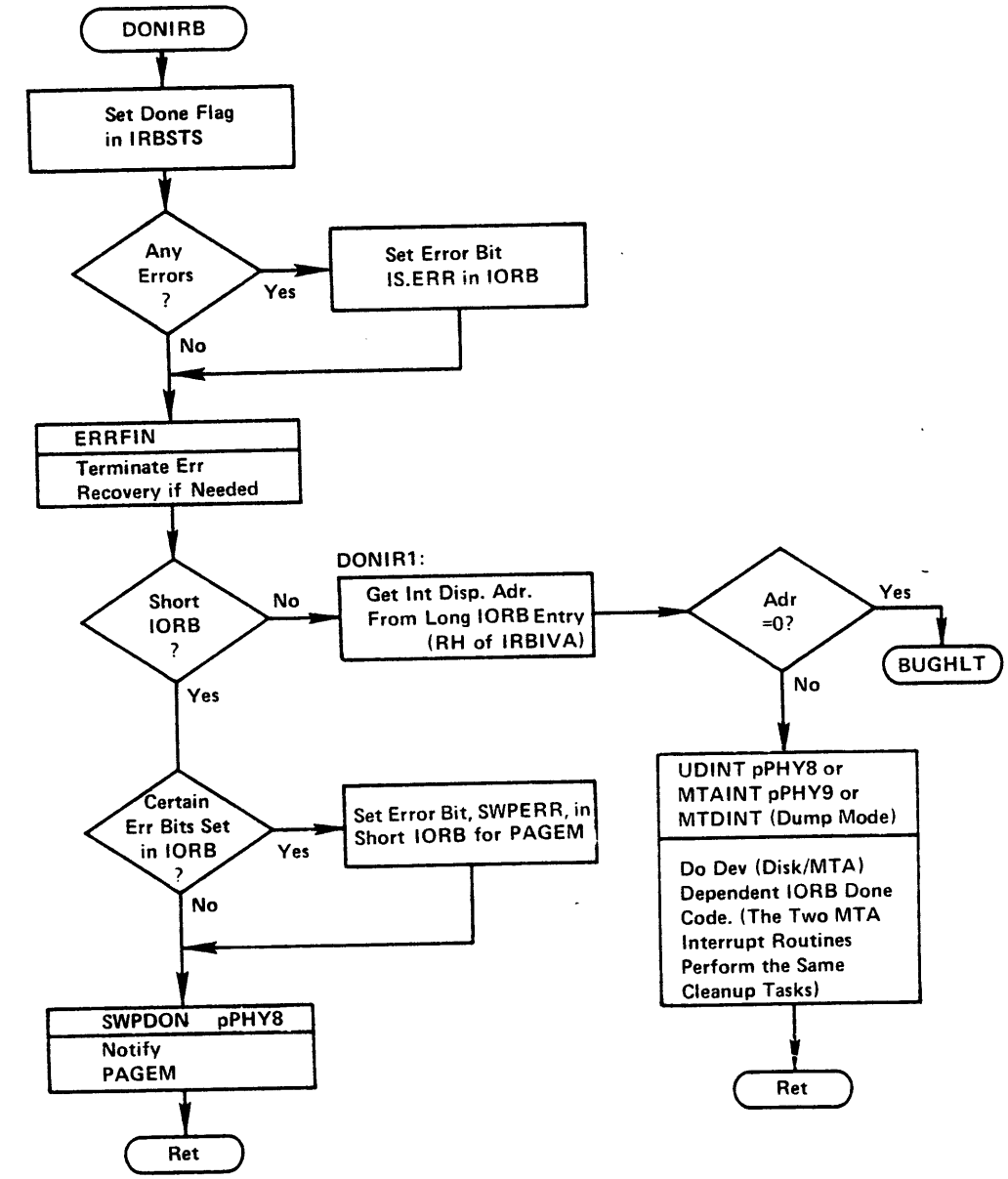
PHY3


Here To Start I/O on an IORB
STRTIO
Short IORB PAGEM Req ?
No
Yes
Get STRTIO Exit Adr from IORB (LH of IRBIVA)
DSK-UDISIE
MTA-MTCHKA
STRTI1:
Unit/ Controller Free ?
No
Yes
BUGHLT – Unit Status Inconsistent
None ?
Yes
No
Execute Routine
DSK–Sets Bits if Unit Offline
MTA–Sets Bit if ABORTF on
2nd Command Active ?
Yes
No
BUGHLT-Illeg Chan Status
ABORT ?
No
Yes
STRTI0:
Set Chan, Control, & Unit Active Bits in CDB, KDB, & UDB
Mark in CDB that a Unit is Holding the Chan
Clear US.OIR & US.OMS Bits (Don't Need any Longer for Oper. Intervention)
SETIO
Set Up For IO–Clear All Err Bits & Set Overdue Time (Now +17 Sec)
OFFTWQ
Remove Req from TWQ
CDSSIO (Chan Disp Adr)
Go Start I/O at Lower Level
DONIRB p PHY5
Flag as Done
OK ?
RETSKP
Successful Return
Return
Unsuccessful Return
No, Offline
CLRACT
Clear Chan & Unit Active Bits
Any Errors Occurred ?
Yes
No
SETOIR
Set Operator Intervention bit
Clear Overdue Timer
Return
Unsuccessful Return
PHY3a

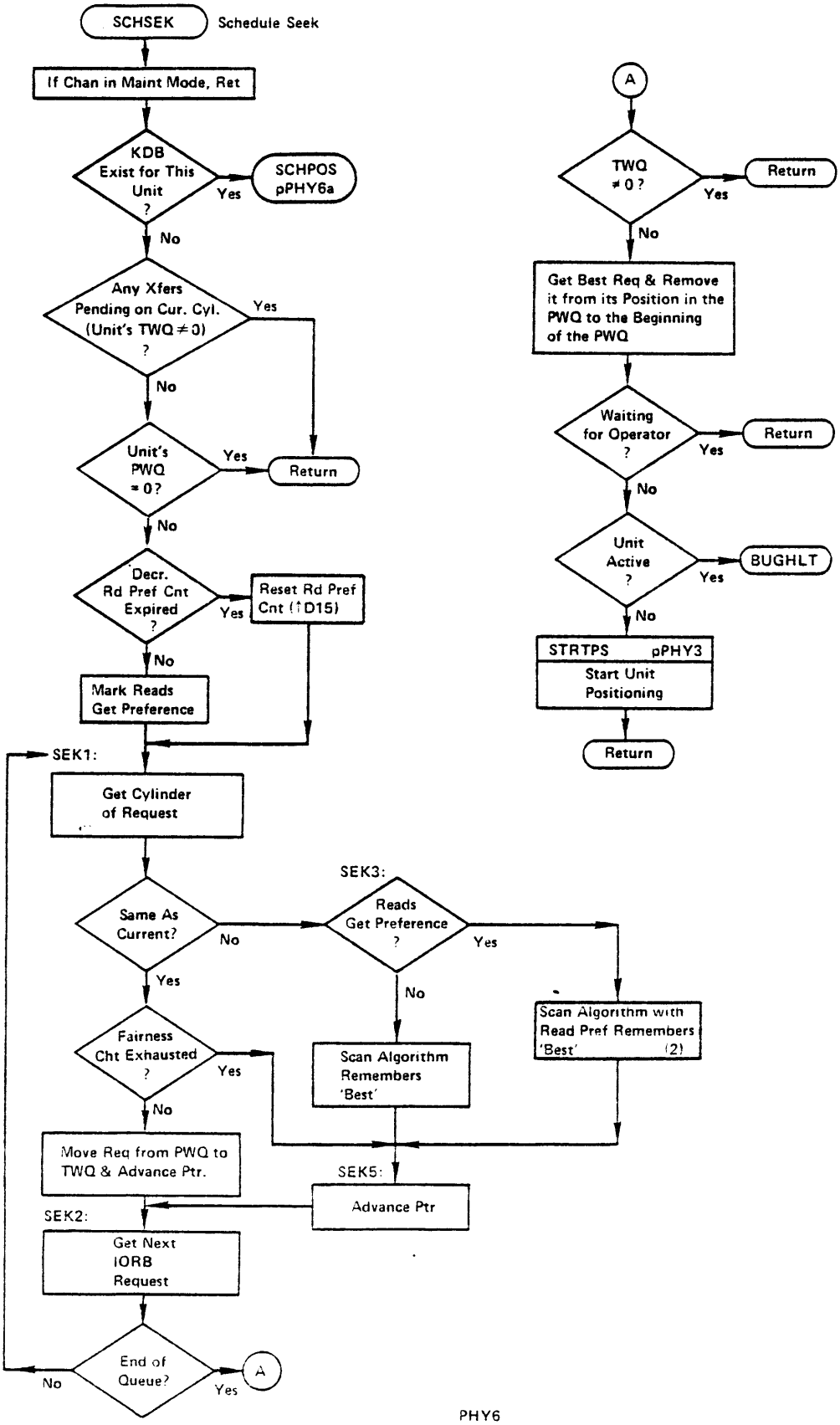

A
INTDON:
OFFTWQ
Remove IORB from PWQ/TWQ
CLRACT
Note No Longer Active
DONIRB pPHY5
Post IORB as Done
INT1
SCHSEK pPHY6
Schedule Seeks (If Needed)
SCHXFR pPHY7
Schedule Transfers
XINT
Set up BLT Ptr for ACs
CDBJEN(CDB)
Restore ACs and Dismiss Int.
PHYIN
Here When Int. Dev. Has Been Determined (CDB Adr Indicates this)
Save AC's in PHYACS
Setup Interrupt PDL (Starts at PHYIPD)
Get Channel's Disp. Tbl Adr. ie CDBDSP (CDB)
CDSINT (Chan Disp Tbl Adr)
Analyze Cause of Interrupt at Channel Dep. Level (1)
Error ?
Yes
Shut Down Stk Command (0 → CS.AC2)
No
Request to Dismiss ?
Yes
No
Request for Chan's Sched Cycle ?
Yes
Is Chan in Err Recovery ?
Yes
No Honor Request
INT1
No
Is This the IORB We Expected From Either the PWQ or TWQ on the Interrupting Unit?
No
BUGHLT
PWQ or TWQ Incorrect
Yes
UDB Indicate Error Recovery in Progress?
Yes
INTERR
pPHY4a
No
Any Errors in IORB's Status Word (IORSTS)?
Yes
INTDON
No
Here When IORBTerminates
DONIRB pPHY5
Post IORB as Done
SCHSEK pPHY6
Schedule Seeks
INT2:
Unit's Pos. Flag On ?
Yes
OFFPWQ
Pull From PWQ
CLRPOS
Clear Positioning Flag
Chan Busy ?
Yes
No
XINT
SCHXFR pPHY7
Schedule Transfers
No
INT3:
Is This a Stacked Command ?
Yes
Clr 2nd Command Active Bit (CS.AC2)
No
A
OFFTWQ
Pull From TWQ
PHY4
DONIRB pPHY5
Post IORB as Done


INTERR
Error Blk Setup for Unit?
Yes
No
Unit Transferring ?
Yes
Set Err Rec. Bit for Chan.
No
ERRSET
Create Error Blk
UDSERR (Unit Disp Adr)
Invoke Error Retry
Retry in Progress ?
No Done with Recovery
Clear Err Rec Bit if Xferring
Yes
XINT
INTDON

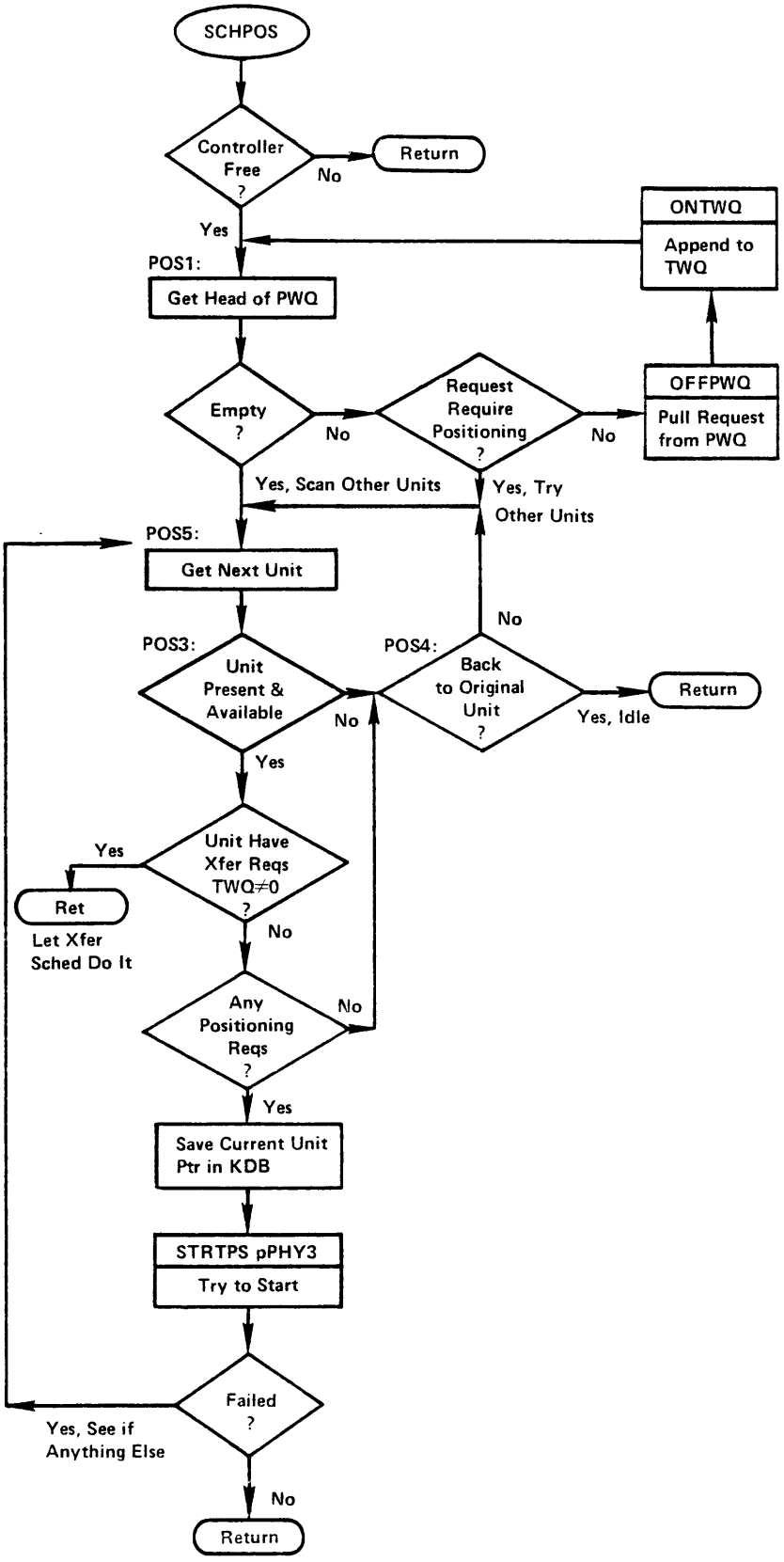
Here to Post an IORB Compiete



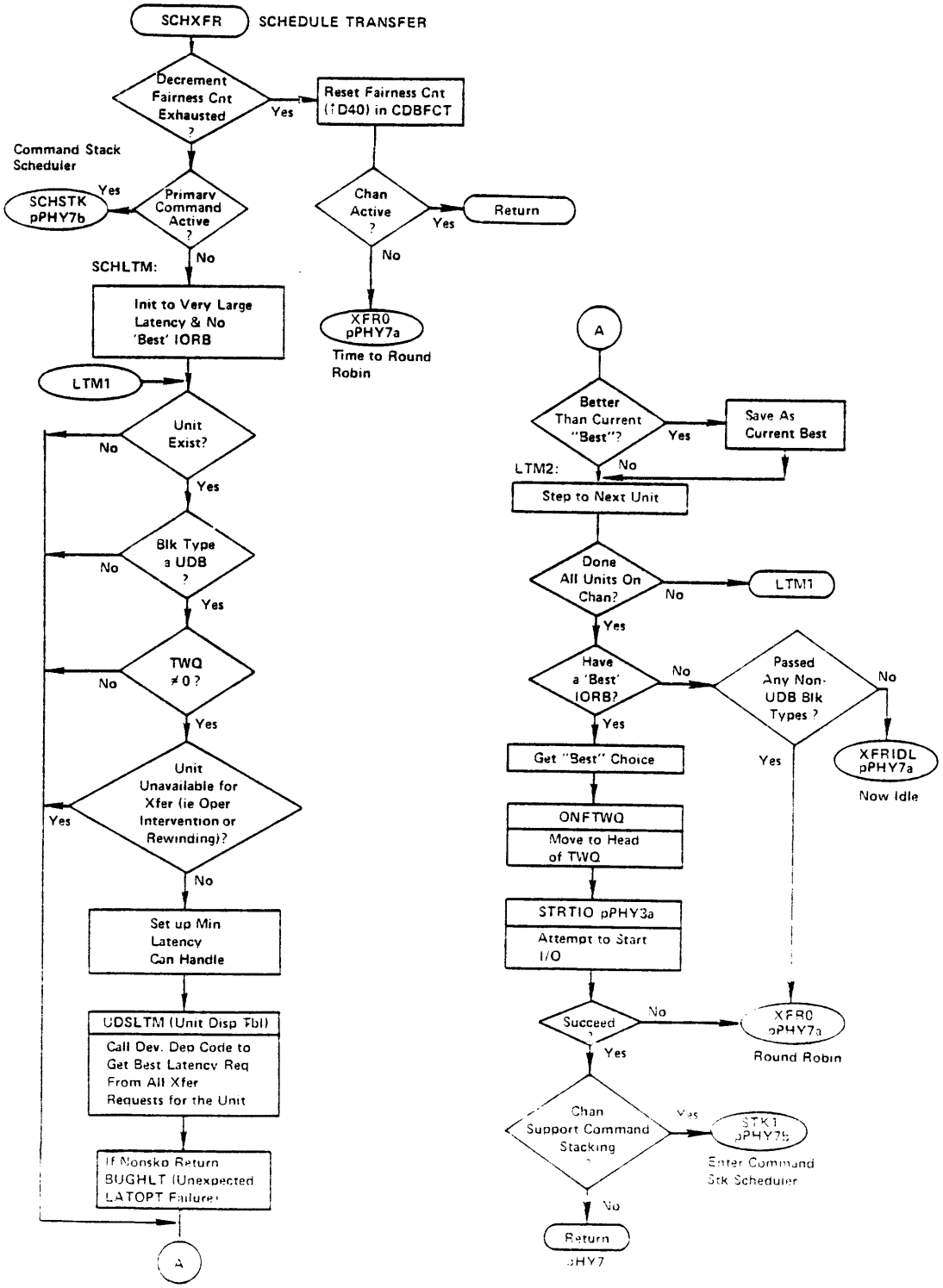
PHY5

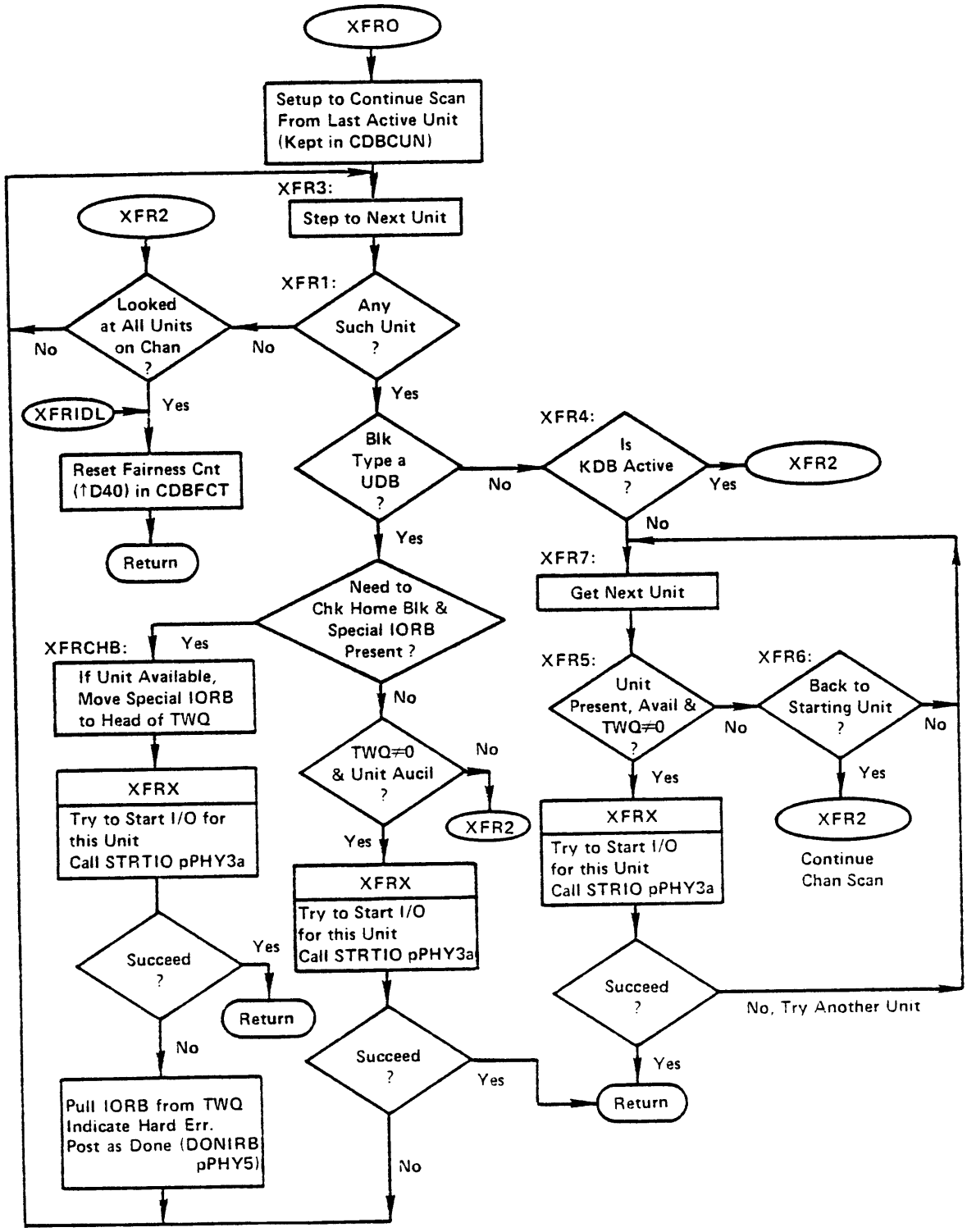

SCHSEK
Schedule Seek
If Chan in Maint Mode, Ret
KDB Exist for This Unit ?
Yes
SCHPOS pPHY6a
No
Any Xfers Pending on Cur. Cyl. (Unit's TWQ ≠ 0) ?
Yes
No
Unit's PWQ = 0?
Yes
Return
No
Decr. Rd Pref Cnt Expired ?
Yes
Reset Rd Pref Cnt (↑D15)
No
Mark Reads Get Preference
SEK1:
Get Cylinder of Request
SEK3:
Same As Current?
No
Reads Get Preference ?
Yes
Yes
No
Fairness Cht Exhausted ?
Yes
No
Scan Algorithm Remembers 'Best'
Scan Algorithm with Read Pref Remembers 'Best' (2)
Move Req from PWQ to TWQ & Advance Ptr.
SEK5:
Advance Ptr
SEK2:
Get Next IORB Request
End of Queue?
No
Yes
A
A
TWQ ≠ 0 ?
Yes
Return
No
Get Best Req & Remove it from its Position in the PWQ to the Beginning of the PWQ
Waiting for Operator ?
Yes
Return
No
Unit Active ?
Yes
BUGHLT
No
STRTPS pPHY3
Start Unit Positioning
Return

PHY6

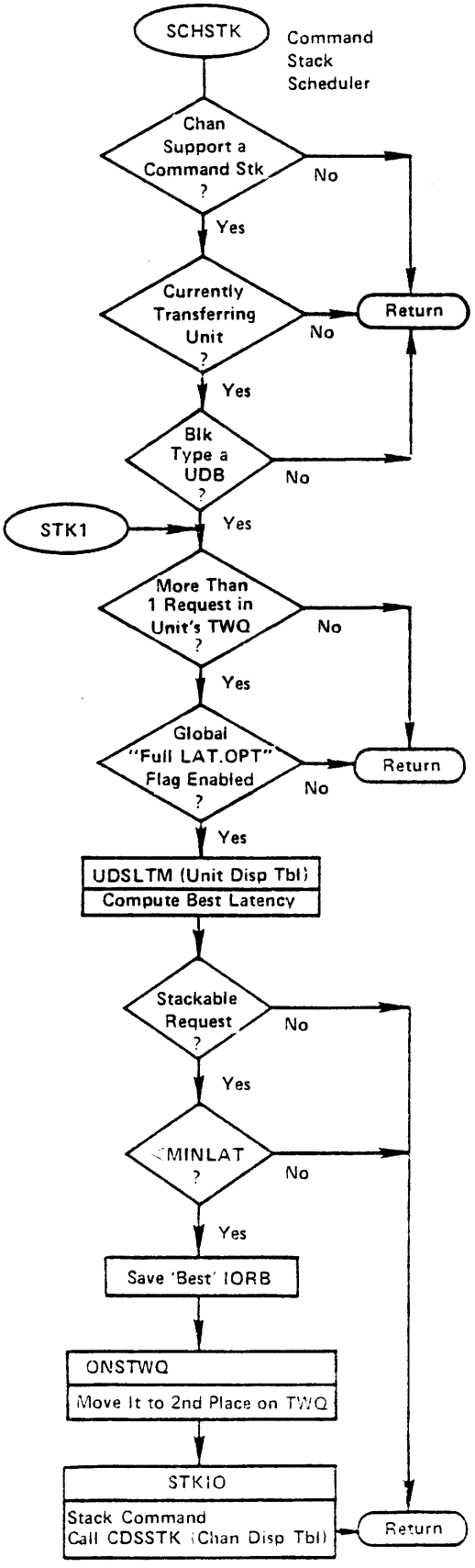

SCHPOS
Controller Free ?
No
Return
Yes
ONTWQ
Append to TWQ
POS1:
Get Head of PWQ
Empty ?
No
Request Require Positioning ?
No
OFFPWQ
Pull Request from PWQ
Yes, Scan Other Units
Yes, Try Other Units
POS5:
Get Next Unit
No
POS3:
Unit Present & Available
POS4:
Back to Original Unit ?
Return
No
Yes, Idle
Yes
Unit Have Xfer Reqs TWQ≠0 ?
Yes
Ret
Let Xfer Sched Do It
No
Any Positioning Reqs ?
No
Yes
Save Current Unit Ptr in KDB
STRTPS pPHY3
Try to Start
Failed ?
Yes, See if Anything Else
No
Return

PHY6a


SCHXFR
SCHEDULE TRANSFER
Decrement Fairness Cnt Exhausted ?
Yes
Reset Fairness Cnt (↑D40) in CDBFCT
Command Stack Scheduler
Yes
SCHSTK pPHY7b
Primary Command Active ?
No
SCHLTM:
Init to Very Large Latency & No 'Best' IORB
LTM1
Unit Exist?
No
Yes
Blk Type a UDB ?
No
Yes
TWQ ≠ 0 ?
No
Yes
Unit Unavailable for Xfer (ie Oper Intervention or Rewinding)?
Yes
No
Set up Min Latency Can Handle
UDSLTM (Unit Disp Tbl)
Call Dev. Dep Code to Get Best Latency Req From All Xfer Requests for the Unit
If Nonsko Return BUGHLT (Unexpected LATOPT Failure)
A
Chan Active ?
Yes
Return
No
XFR0 pPHY7a
Time to Round Robin
A
Better Than Current "Best"?
Yes
Save As Current Best
No
LTM2:
Step to Next Unit
Done All Units On Chan?
No
LTM1
Yes
Have a 'Best' IORB?
No
Passed Any Non-UDB Blk Types ?
No
Yes
XFRIDL pPHY7a
Now Idle
Yes
Get "Best" Choice
ONFTWQ
Move to Head of TWQ
STRTIO pPHY3a
Attempt to Start I/O
Succeed ?
No
XFR0 pPHY7a
Round Robin
Yes
Chan Support Command Stacking
Yes
STK1 pPHY7b
Enter Command Stk Scheduler
No
Return
pHY7

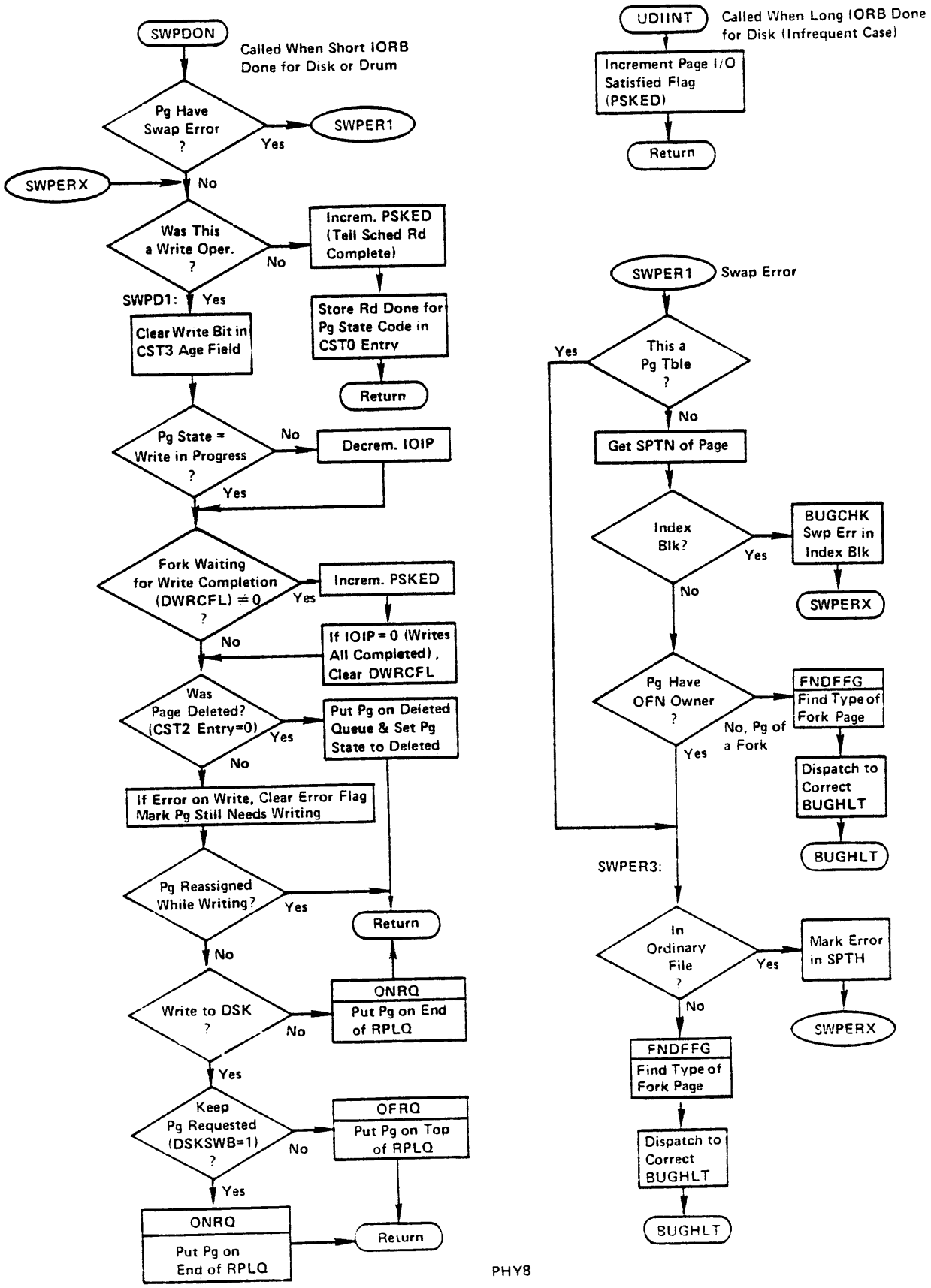

XFRO
Setup to Continue Scan From Last Active Unit (Kept in CDBCUN)
XFR3:
Step to Next Unit
XFR2
XFR1:
Any Such Unit ?
No
Looked at All Units on Chan ?
No
Yes
XFRIDL
Yes
Reset Fairness Cnt (↑D40) in CDBFCT
Return
Blk Type a UDB ?
No
Yes
XFR4:
Is KDB Active ?
Yes
XFR2
No
XFR7:
Get Next Unit
Need to Chk Home Blk & Special IORB Present ?
XFRCHB:
Yes
If Unit Available, Move Special IORB to Head of TWQ
XFRX
Try to Start I/O for this Unit
Call STRTIO pPHY3a
Succeed ?
Yes
Return
No
Pull IORB from TWQ
Indicate Hard Err.
Post as Done (DONIRB pPHY5)
No
TWQ≠0 & Unit Aucil ?
No
XFR2
Yes
XFRX
Try to Start I/O for this Unit
Call STRTIO pPHY3a
Succeed ?
Yes
Return
No
XFR5:
Unit Present, Avail & TWQ≠0 ?
No
Yes
XFRX
Try to Start I/O for this Unit
Call STRIO pPHY3a
Succeed ?
No, Try Another Unit
Yes
Return
XFR6:
Back to Starting Unit ?
No
Yes
XFR2
Continue Chan Scan

PHY7a


SCHSTK
Command Stack Scheduler
Chan Support a Command Stk ?
No
Yes
Currently Transferring Unit ?
No
Return
Yes
Blk Type a UDB ?
No
STK1
Yes
More Than 1 Request in Unit's TWQ ?
No
Yes
Global "Full LAT.OPT" Flag Enabled ?
Return
No
Yes
UDSLTM (Unit Disp Tbl)
Compute Best Latency
Stackable Request ?
No
Yes
<MINLAT ?
No
Yes
Save 'Best' IORB
ONSTWQ
Move It to 2nd Place on TWQ
STK1O
Stack Command
Call CDSSTK (Chan Disp Tbl)
Return

PHY7b

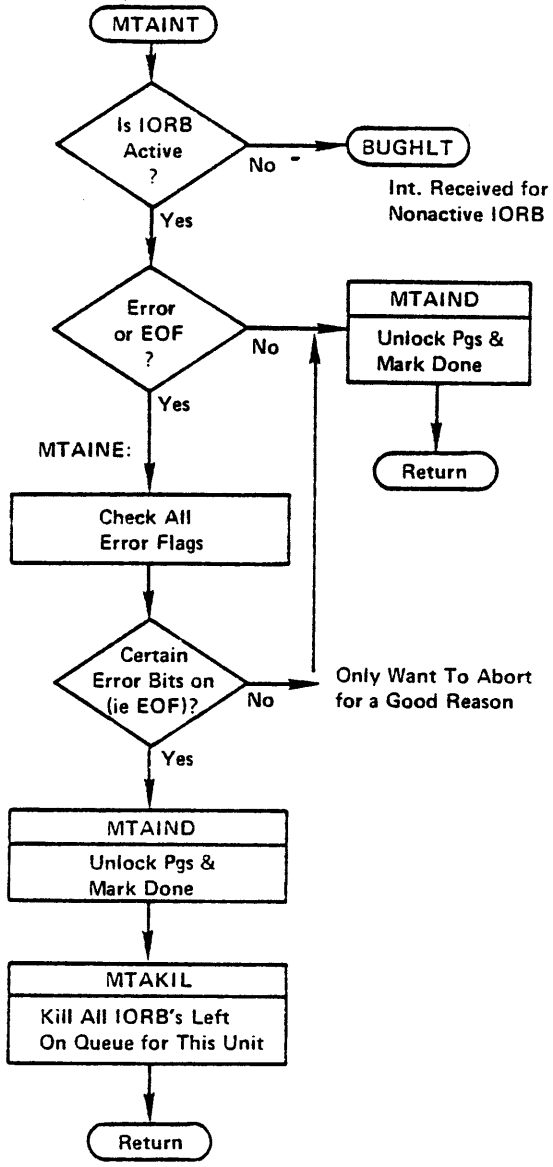
"INTERRUPT DONE" DSK/DRUM DEPENDENT CODE

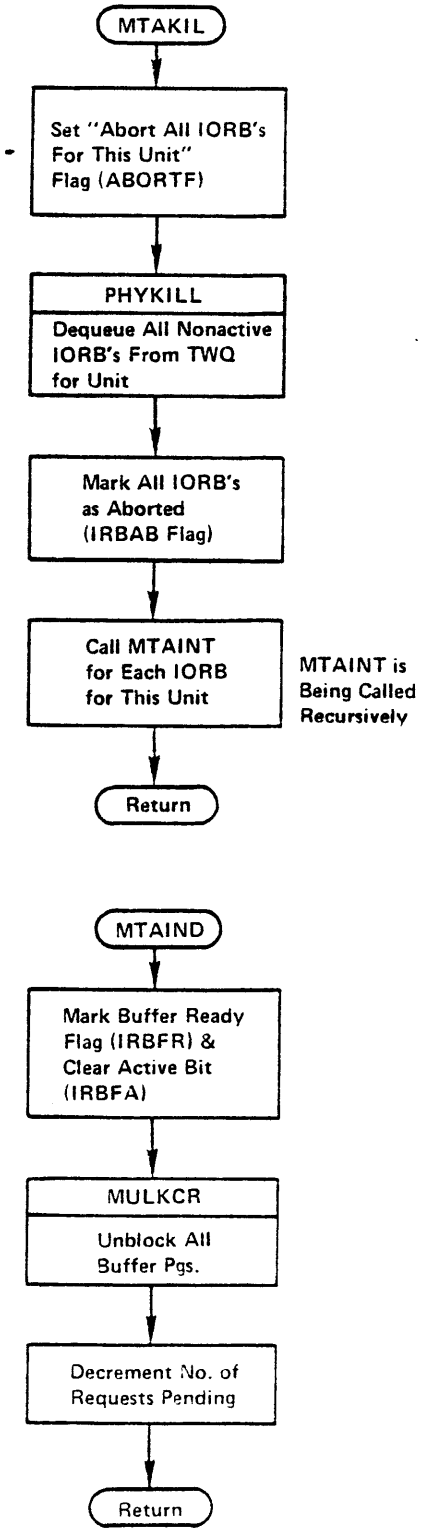


PHY8

"INTERRUPT DONE" MAGTAPE DEPENDENT CODE

Called When Non-Dump Xfer Done For MTA At Interrupt Level





Requesting DISK/MTA I/O Comments

SIO1

(1) The algorithm for queuing up a MTA request is:

If the request requires positioning, append the request to the PWQ.

If the request requires no positioning (i.e., Read/Write Forward or Read Reverse) append the request to the TWQ only if the PWQ is empty. Otherwise, append it to the PWQ.

DSK/MTA Interrupt Handling Comments

PHYINT

(1) The channel dependent routine (RH2INT for RH2Øs) is called to analyze the interrupt. Lower level routines called by RH2INT (i.e., Unit dependent routines) return an argument in AC, P4, to PHYINT to indicate whether to dismiss the interrupt (P4 = Ø), to schedule another channel cycle right away (P4 < Ø) or to housekeep the current request (P4 > Ø) before scheduling another channel cycle. The channel dependent routine also records error information so that PHYINT can see if error recovery is in progress or should be started.

The request to dismiss (P4 = Ø) is invoked for example when the done flag is on and the channel is not occupied. The request for an immediate channel cycle (P4 < Ø) is made when a positioning done interrupt has occurred and there is no transfer in progress. Transfer Done requests will require further housekeeping (P4 > Ø) by PHYINT before scheduling another channel cycle.

SCHSEK

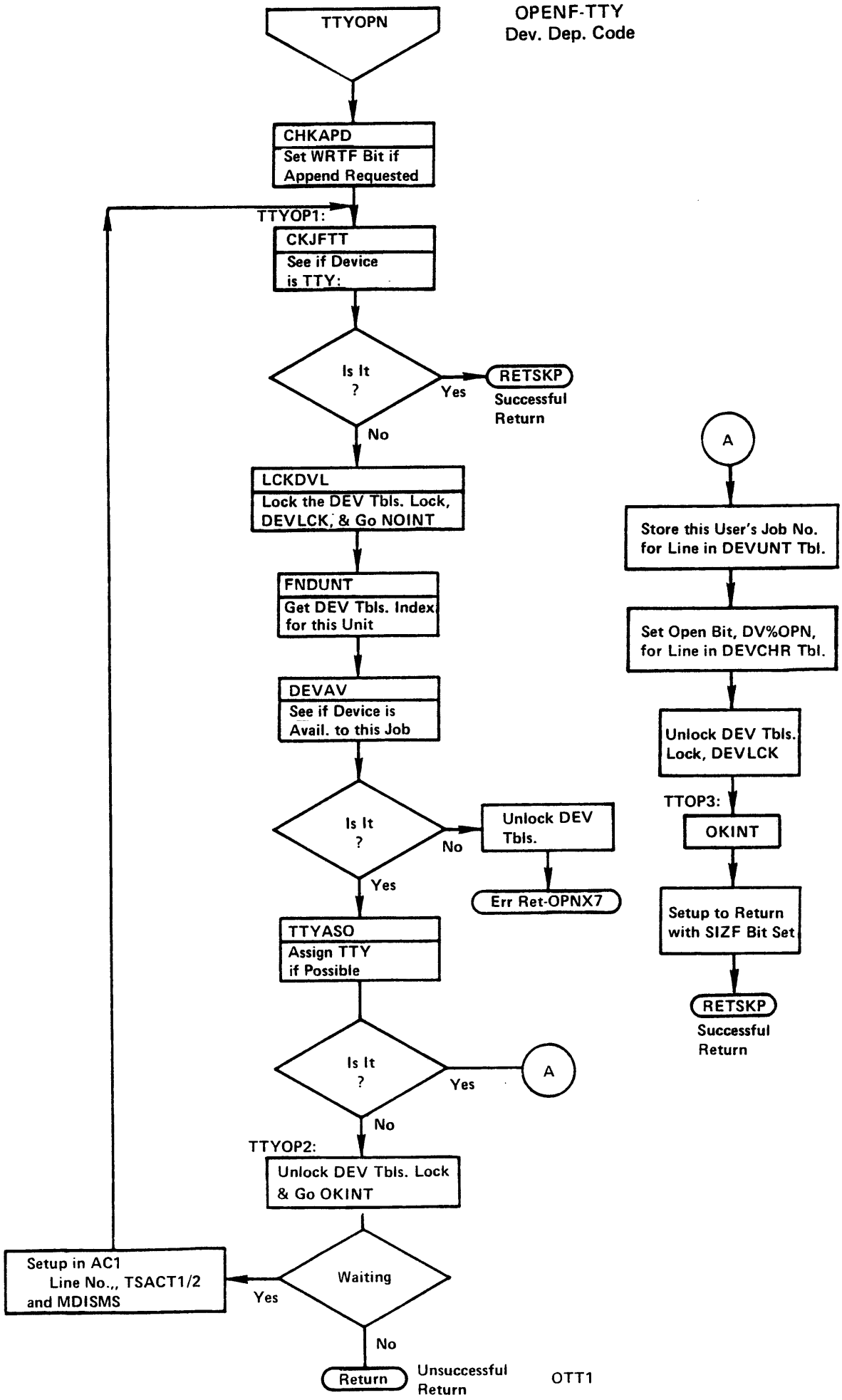
(2) The scan algorithm with read preference in effect performs as follows:

Take the next higher-numbered cylinder read request from the current cylinder. If none, take the next higher-numbered cylinder (write) request from the current cylinder.

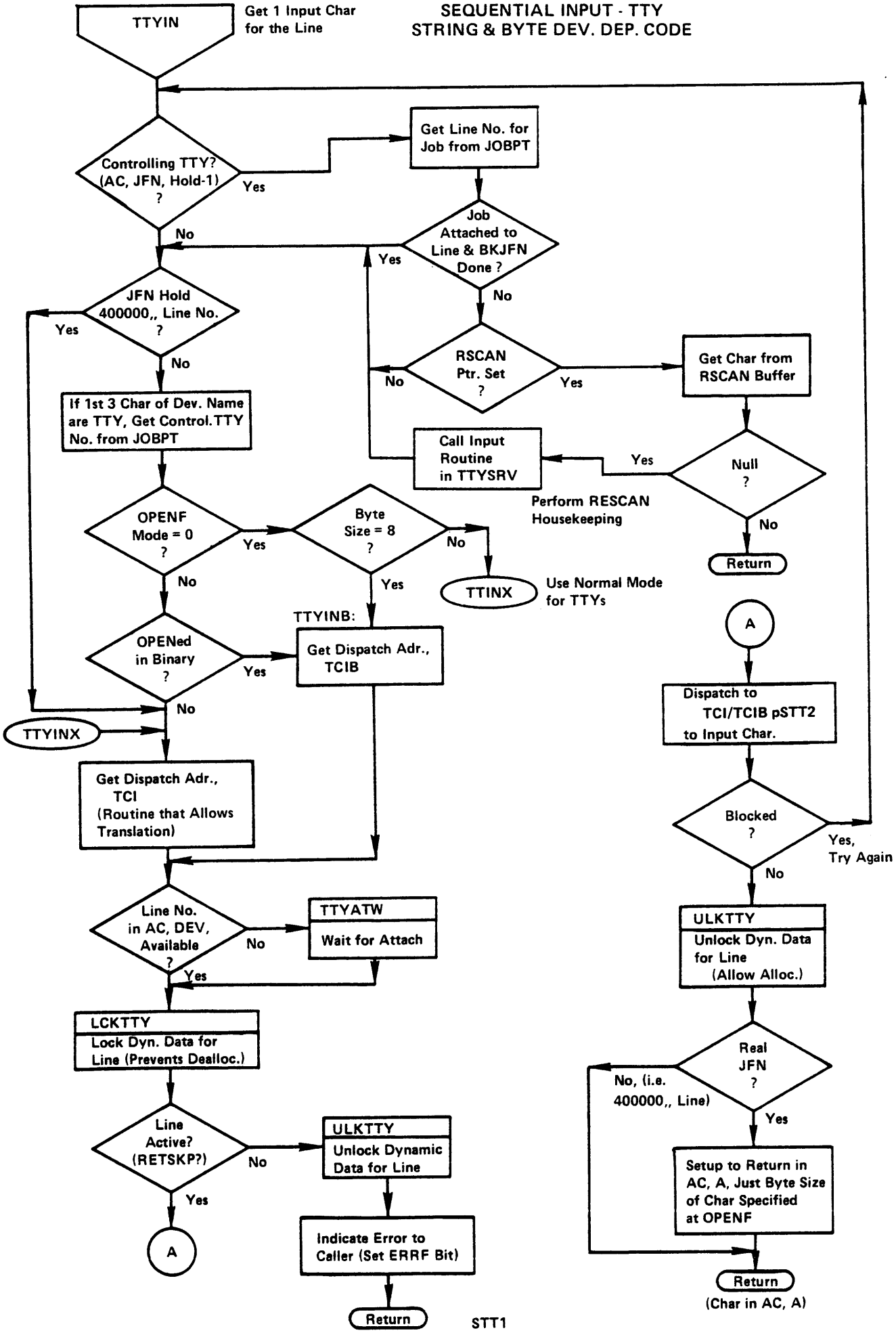
If none, take the lowest numbered cylinder read request from the current cylinder. If none, take the lowest numbered cylinder (write) request from the current cylinder.

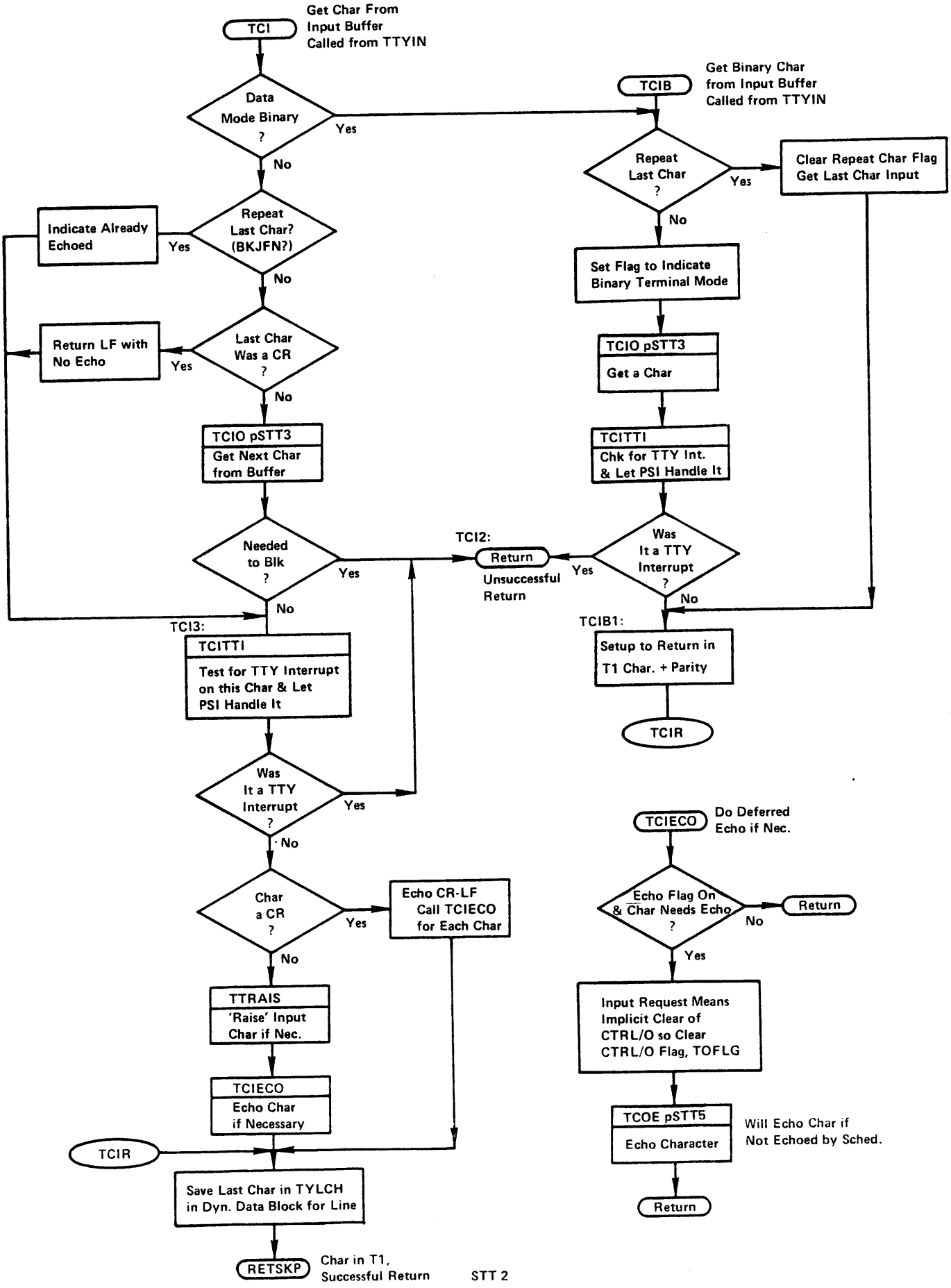
JSYS CALL FLOWCHARTS

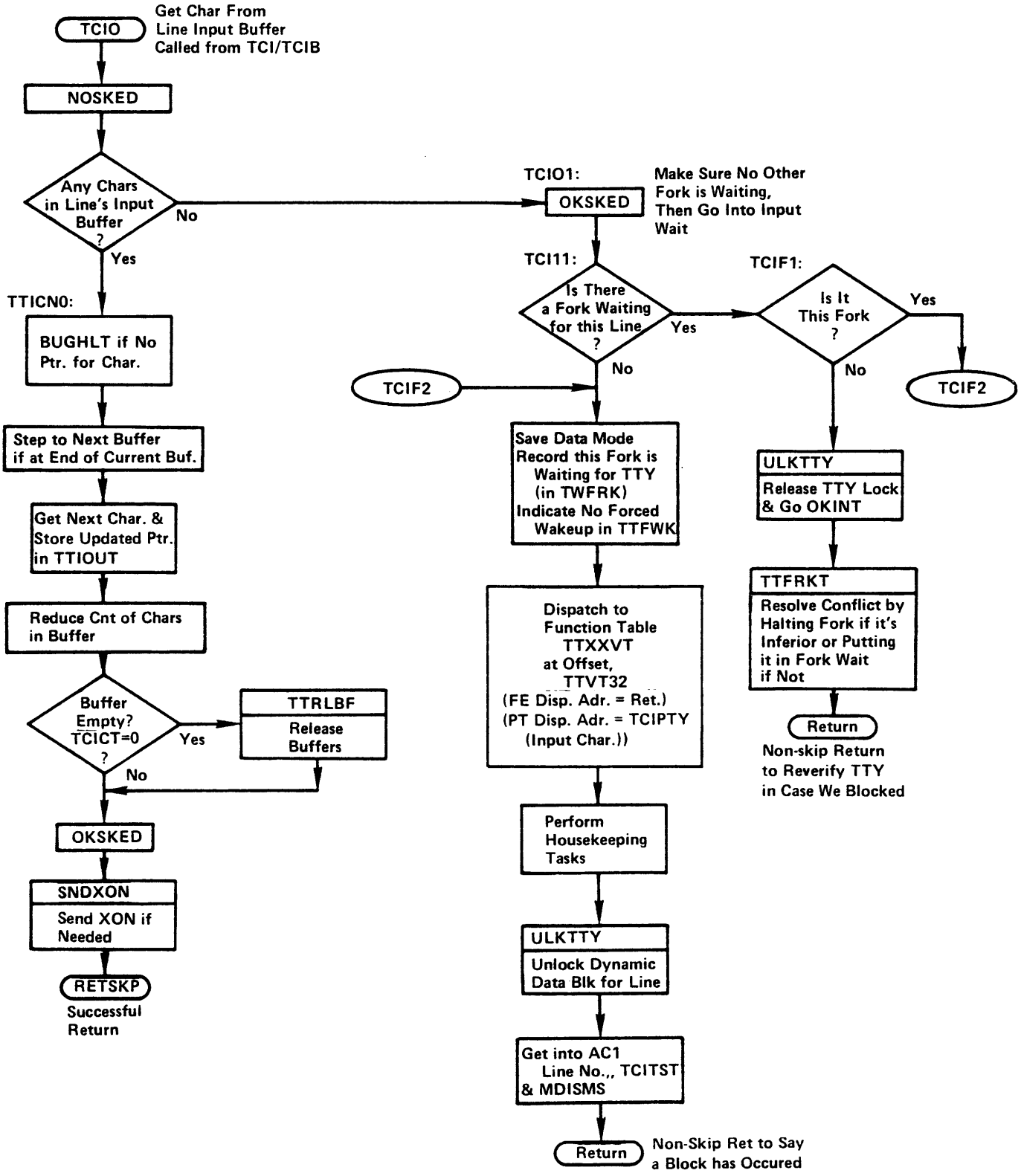
TTY DEPENDENT LEVEL

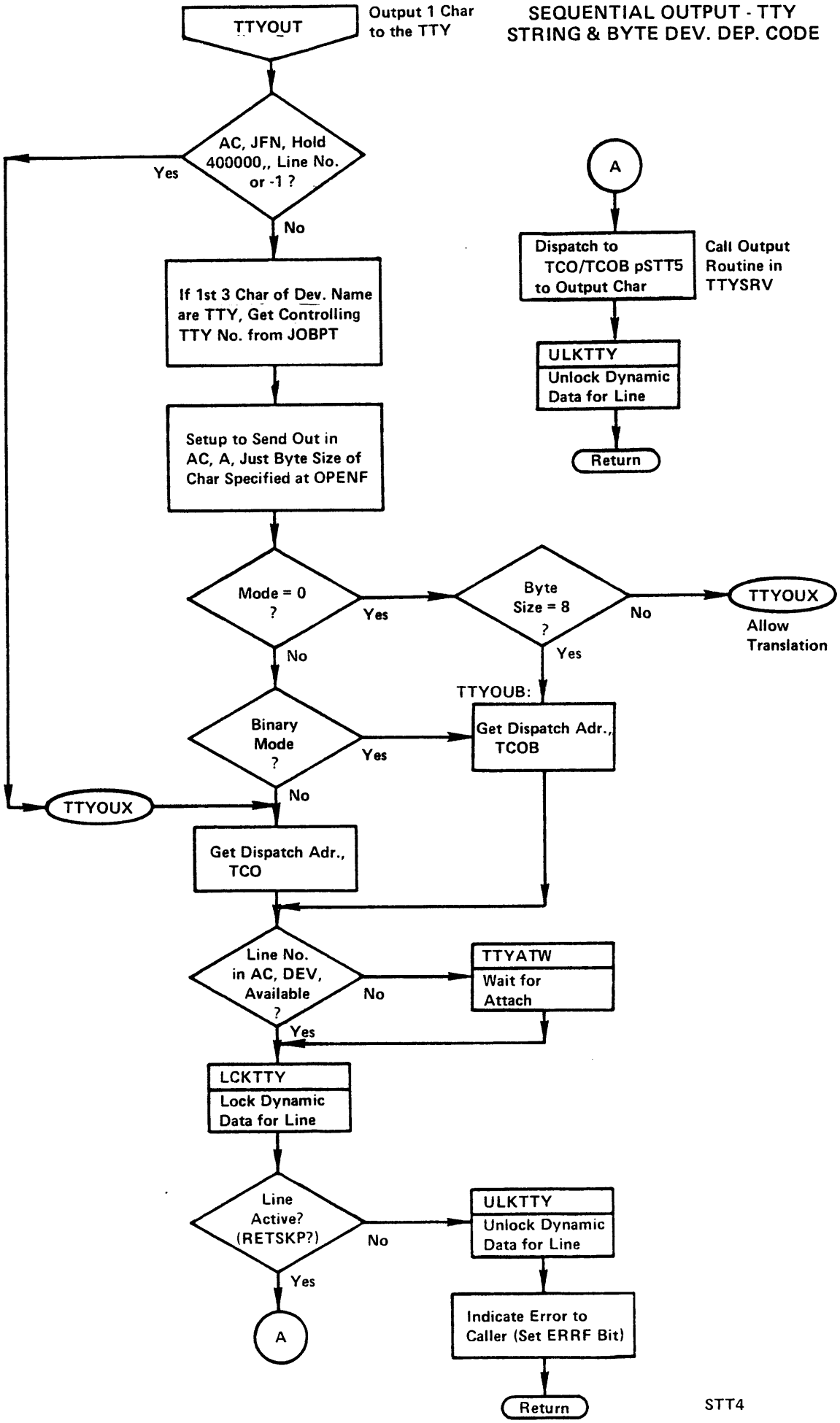

TTYOPN
OPENF-TTY
Dev. Dep. Code
CHKAPD
Set WRTF Bit if
Append Requested
TTYOP1:
CKJFTT
See if Device
is TTY:
Is It
?
Yes
RETSKP
Successful
Return
No
LCKDVL
Lock the DEV Tbls. Lock,
DEVLCK, & Go NOINT
FNDUNT
Get DEV Tbls. Index
for this Unit
DEVAV
See if Device is
Avail. to this Job
Is It
?
No
Unlock DEV
Tbls.
Err Ret-OPNX7
Yes
TTYASO
Assign TTY
if Possible
Is It
?
Yes
A
No
TTYOP2:
Unlock DEV Tbls. Lock
& Go OKINT
Waiting
Yes
Setup in AC1
Line No.,, TSACT1/2
and MDISMS
No
Return
Unsuccessful
Return
A
Store this User's Job No.
for Line in DEVUNT Tbl.
Set Open Bit, DV%OPN,
for Line in DEVCHR Tbl.
Unlock DEV Tbls.
Lock, DEVLCK
TTOP3:
OKINT
Setup to Return
with SIZF Bit Set
RETSKP
Successful
Return

OTT1

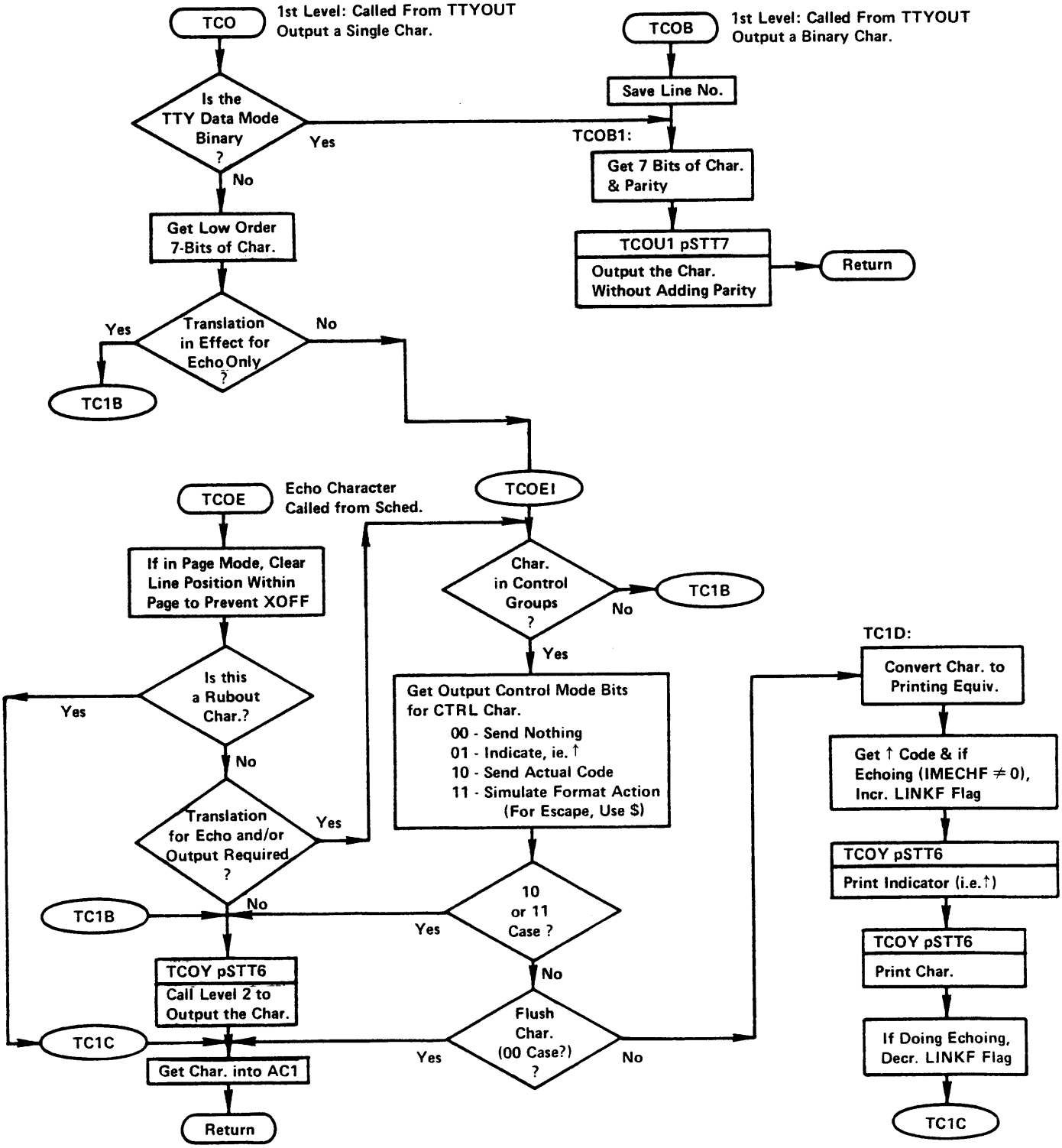

TTYIN
Get 1 Input Char for the Line
SEQUENTIAL INPUT - TTY
STRING & BYTE DEV. DEP. CODE
Controlling TTY? (AC, JFN, Hold-1) ?
Yes
Get Line No. for Job from JOBPT
No
Job Attached to Line & BKJFN Done ?
Yes
No
JFN Hold 400000,, Line No. ?
Yes
No
RSCAN Ptr. Set ?
No
Yes
Get Char from RSCAN Buffer
If 1st 3 Char of Dev. Name are TTY, Get Control.TTY No. from JOBPT
Call Input Routine in TTYSRV
Yes
Null ?
Perform RESCAN Housekeeping
No
Return
OPENF Mode = 0 ?
Yes
Byte Size = 8 ?
No
TTINX
Use Normal Mode for TTYs
No
Yes
TTYINB:
OPENed in Binary ?
Yes
Get Dispatch Adr., TCIB
No
TTYINX
Get Dispatch Adr., TCI (Routine that Allows Translation)
A
Dispatch to TCI/TCIB pSTT2 to Input Char.
Blocked ?
Yes, Try Again
No
Line No. in AC, DEV, Available ?
No
TTYATW
Wait for Attach
Yes
ULKTTY
Unlock Dyn. Data for Line (Allow Alloc.)
LCKTTY
Lock Dyn. Data for Line (Prevents Dealloc.)
Real JFN ?
No, (i.e. 400000,, Line)
Yes
Line Active? (RETSKP?)
No
ULKTTY
Unlock Dynamic Data for Line
Setup to Return in AC, A, Just Byte Size of Char Specified at OPENF
Yes
A
Indicate Error to Caller (Set ERRF Bit)
Return
Return
(Char in AC, A)
STT1


TCI
Get Char From Input Buffer Called from TTYIN
Data Mode Binary ?
Yes
No
TCIB
Get Binary Char from Input Buffer Called from TTYIN
Repeat Last Char ?
Yes
Clear Repeat Char Flag Get Last Char Input
No
Set Flag to Indicate Binary Terminal Mode
TCIO pSTT3
Get a Char
TCITTI
Chk for TTY Int. & Let PSI Handle It
Was It a TTY Interrupt ?
Yes
No
TCI2:
Return
Unsuccessful Return
TCIB1:
Setup to Return in T1 Char. + Parity
TCIR
Repeat Last Char? (BKJFN?)
Indicate Already Echoed
Yes
No
Last Char Was a CR ?
Return LF with No Echo
Yes
No
TCIO pSTT3
Get Next Char from Buffer
Needed to Blk ?
Yes
No
TCI3:
TCITTI
Test for TTY Interrupt on this Char & Let PSI Handle It
Was It a TTY Interrupt ?
Yes
No
Char a CR ?
Yes
Echo CR-LF Call TCIECO for Each Char
No
TTRAIS
'Raise' Input Char if Nec.
TCIECO
Echo Char if Necessary
TCIR
Save Last Char in TYLCH in Dyn. Data Block for Line
RETSKP
Char in T1, Successful Return
STT 2
TCIECO
Do Deferred Echo if Nec.
Echo Flag On & Char Needs Echo ?
No
Return
Yes
Input Request Means Implicit Clear of CTRL/O so Clear CTRL/O Flag, TOFLG
TCOE pSTT5
Echo Character
Will Echo Char if Not Echoed by Sched.
Return

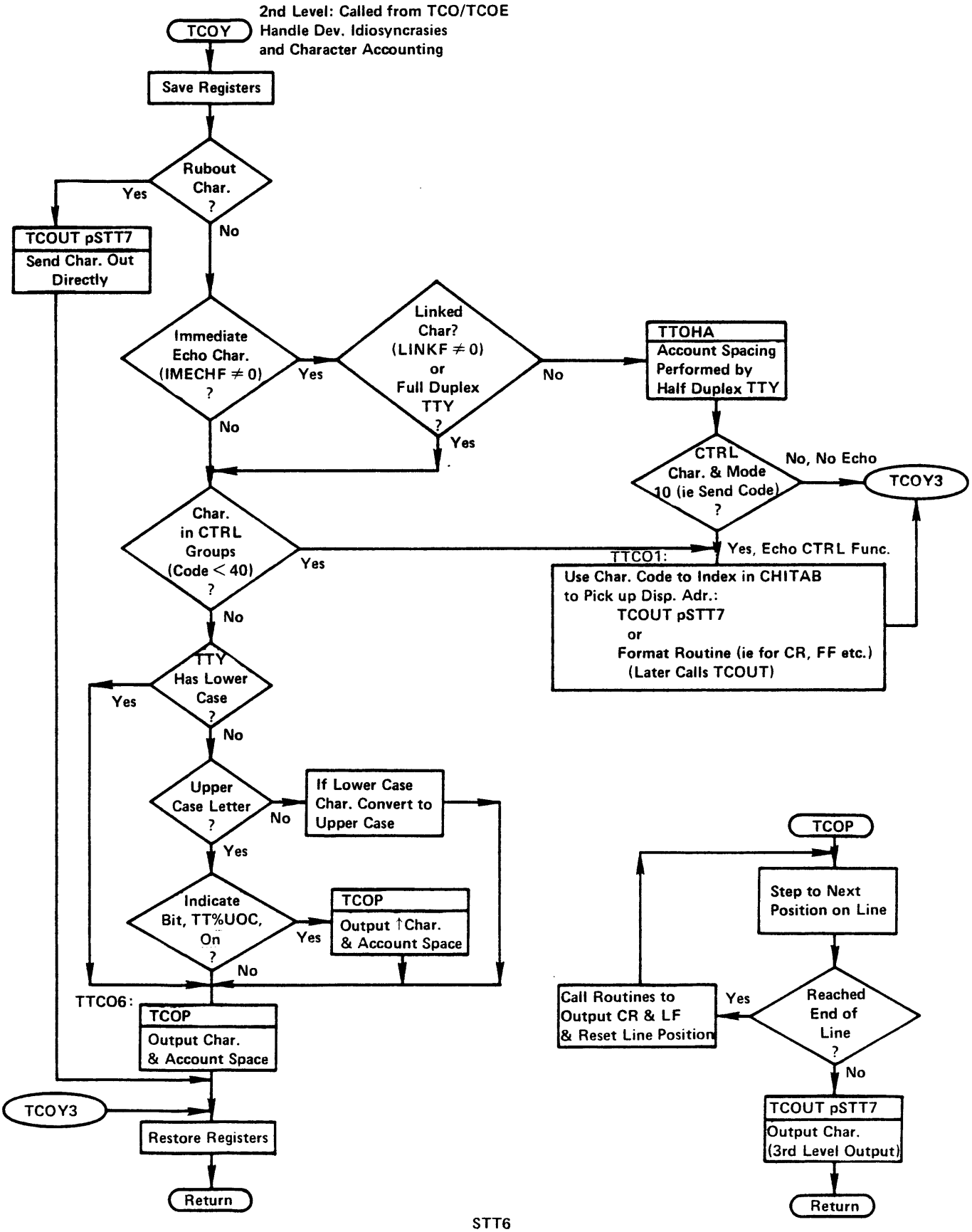

TCIO
Get Char From Line Input Buffer Called from TCI/TCIB
NOSKED
Any Chars in Line's Input Buffer ?
No
Yes
TTICN0:
BUGHLT if No Ptr. for Char.
Step to Next Buffer if at End of Current Buf.
Get Next Char. & Store Updated Ptr. in TTIOUT
Reduce Cnt of Chars in Buffer
Buffer Empty? TCICT=0 ?
Yes
TTRLBF
Release Buffers
No
OKSKED
SNDXON
Send XON if Needed
RETSKP
Successful Return
TCIO1:
OKSKED
Make Sure No Other Fork is Waiting, Then Go Into Input Wait
TCI11:
Is There a Fork Waiting for this Line ?
Yes
No
TCIF1:
Is It This Fork ?
Yes
No
TCIF2
TCIF2
Save Data Mode
Record this Fork is Waiting for TTY (in TWFRK)
Indicate No Forced Wakeup in TTFWK
Dispatch to Function Table TTXXVT at Offset, TTVT32
(FE Disp. Adr. = Ret.)
(PT Disp. Adr. = TCIPTY (Input Char.))
Perform Housekeeping Tasks
ULKTTY
Unlock Dynamic Data Blk for Line
Get into AC1 Line No.,, TCITST & MDISMS
Return
Non-Skip Ret to Say a Block has Occured
ULKTTY
Release TTY Lock & Go OKINT
TTFRKT
Resolve Conflict by Halting Fork if it's Inferior or Putting it in Fork Wait if Not
Return
Non-skip Return to Reverify TTY in Case We Blocked


TTYOUT
Output 1 Char to the TTY
SEQUENTIAL OUTPUT - TTY
STRING & BYTE DEV. DEP. CODE
AC, JFN, Hold 400000,, Line No. or -1 ?
Yes
No
If 1st 3 Char of Dev. Name are TTY, Get Controlling TTY No. from JOBPT
Setup to Send Out in AC, A, Just Byte Size of Char Specified at OPENF
Mode = 0 ?
Yes
No
Byte Size = 8 ?
No
Yes
TTYOUX
Allow Translation
TTYOUB:
Binary Mode ?
Yes
No
Get Dispatch Adr., TCOB
TTYOUX
Get Dispatch Adr., TCO
Line No. in AC, DEV, Available ?
No
Yes
TTYATW
Wait for Attach
LCKTTY
Lock Dynamic Data for Line
Line Active? (RETSKP?)
No
Yes
ULKTTY
Unlock Dynamic Data for Line
Indicate Error to Caller (Set ERRF Bit)
Return
A
A
Dispatch to TCO/TCOB pSTT5 to Output Char
Call Output Routine in TTYSRV
ULKTTY
Unlock Dynamic Data for Line
Return
STT4

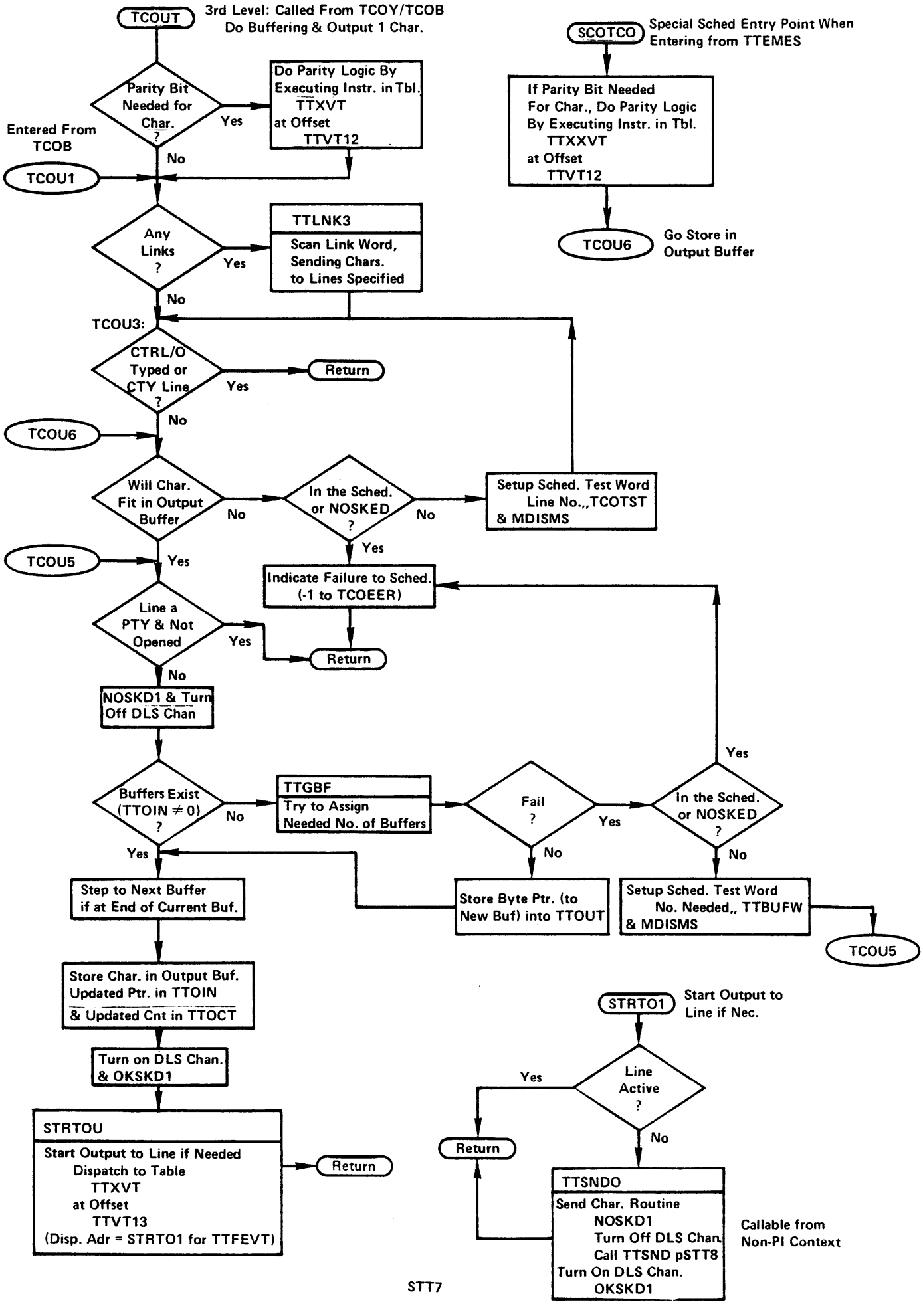
TCO - 1st Level - Translate According to Program's Desires
TCOY - 2nd Level - Do Links & Format for a Particular Device
TCOUT - 3rd Level - Do Buffering, etc.



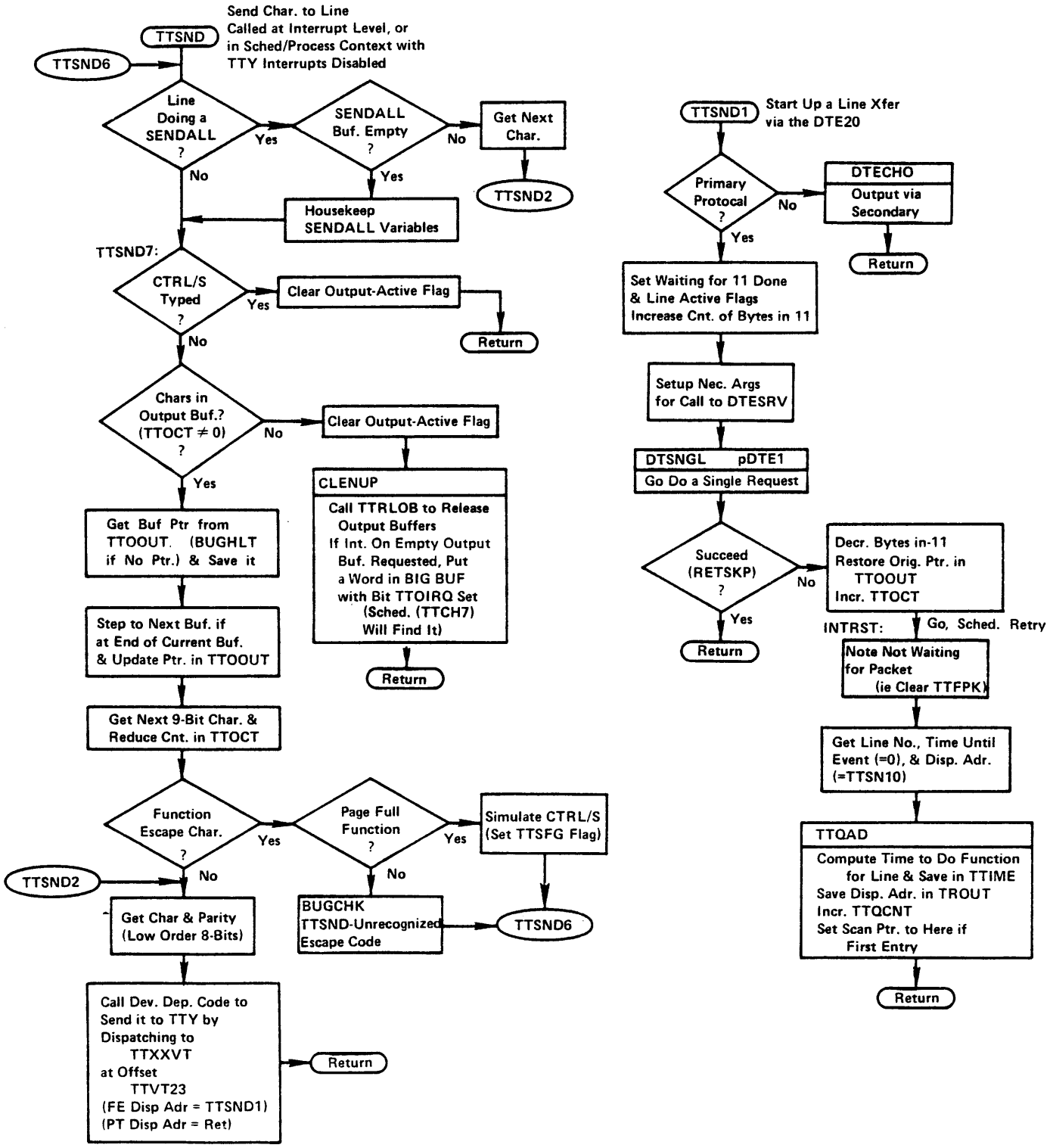
STT5


2nd Level: Called from TCO/TCOE
Handle Dev. Idiosyncrasies
and Character Accounting
TCOY
Save Registers
Rubout Char. ?
Yes
No
TCOUT pSTT7
Send Char. Out Directly
Immediate Echo Char. (IMECHF ≠ 0) ?
Yes
No
Linked Char? (LINKF ≠ 0) or Full Duplex TTY ?
Yes
No
TTOHA
Account Spacing Performed by Half Duplex TTY
CTRL Char. & Mode 10 (ie Send Code) ?
No, No Echo
TCOY3
Yes, Echo CTRL Func.
Char. in CTRL Groups (Code < 40) ?
Yes
No
TTCO1:
Use Char. Code to Index in CHITAB to Pick up Disp. Adr.:
TCOUT pSTT7
or
Format Routine (ie for CR, FF etc.)
(Later Calls TCOUT)
TTY Has Lower Case ?
Yes
No
Upper Case Letter ?
No
If Lower Case Char. Convert to Upper Case
Yes
Indicate Bit, TT%UOC, On ?
Yes
TCOP
Output ↑Char. & Account Space
No
TTCO6:
TCOP
Output Char. & Account Space
TCOY3
Restore Registers
Return
TCOP
Step to Next Position on Line
Reached End of Line ?
Yes
Call Routines to Output CR & LF & Reset Line Position
No
TCOUT pSTT7
Output Char. (3rd Level Output)
Return

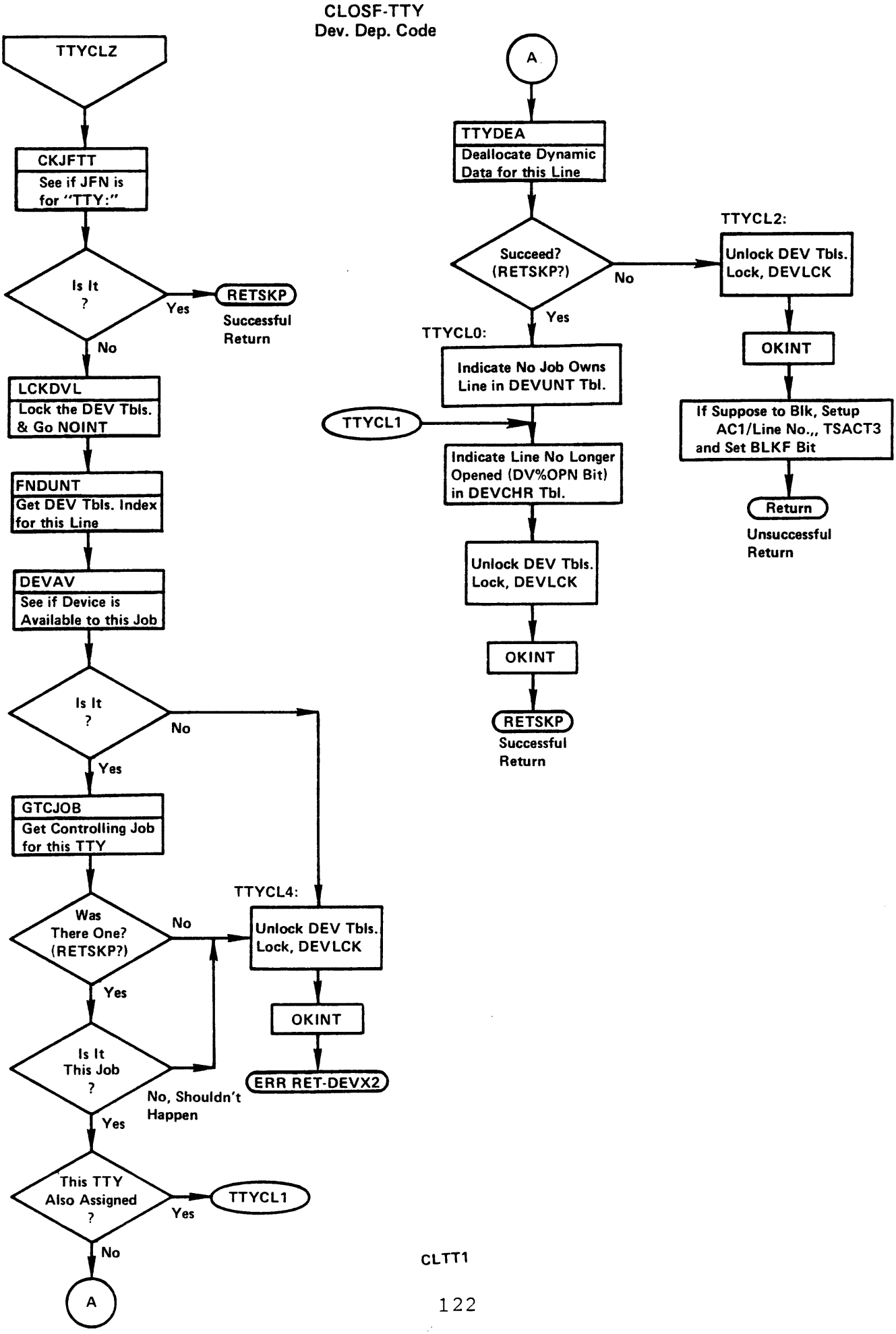
STT6


TCOUT
3rd Level: Called From TCOY/TCOB
Do Buffering & Output 1 Char.
Parity Bit Needed for Char. ?
Yes
Do Parity Logic By Executing Instr. in Tbl. TTXVT at Offset TTVT12
Entered From TCOB
No
TCOU1
Any Links ?
Yes
TTLNK3
Scan Link Word, Sending Chars. to Lines Specified
No
TCOU3:
CTRL/O Typed or CTY Line ?
Yes
Return
No
TCOU6
Will Char. Fit in Output Buffer
No
In the Sched. or NOSKED ?
No
Setup Sched. Test Word Line No.,,TCOTST & MDISMS
Yes
TCOU5
Yes
Indicate Failure to Sched. (-1 to TCOEER)
Line a PTY & Not Opened
Yes
Return
No
NOSKD1 & Turn Off DLS Chan
Buffers Exist (TTOIN ≠ 0) ?
No
TTGBF
Try to Assign Needed No. of Buffers
Fail ?
Yes
In the Sched. or NOSKED ?
Yes
No
No
Yes
Step to Next Buffer if at End of Current Buf.
Store Byte Ptr. (to New Buf) into TTOUT
Setup Sched. Test Word No. Needed,, TTBUFW & MDISMS
TCOU5
Store Char. in Output Buf. Updated Ptr. in TTOIN & Updated Cnt in TTOCT
Turn on DLS Chan. & OKSKD1
STRTOU
Start Output to Line if Needed
Dispatch to Table TTXVT at Offset TTVT13
(Disp. Adr = STRTO1 for TTFEVT)
Return
SCOTCO
Special Sched Entry Point When Entering from TTEMES
If Parity Bit Needed For Char., Do Parity Logic By Executing Instr. in Tbl. TTXXVT at Offset TTVT12
TCOU6
Go Store in Output Buffer
STRTO1
Start Output to Line if Nec.
Line Active ?
Yes
No
Return
TTSNDO
Send Char. Routine NOSKD1
Turn Off DLS Chan.
Call TTSND pSTT8
Turn On DLS Chan. OKSKD1
Callable from Non-PI Context

STT7


Send Char. to Line
Called at Interrupt Level, or
in Sched/Process Context with
TTY Interrupts Disabled
TTSND
TTSND6
Line Doing a SENDALL ?
Yes
No
SENDALL Buf. Empty ?
No
Yes
Get Next Char.
TTSND2
Housekeep SENDALL Variables
TTSND7:
CTRL/S Typed ?
Yes
Clear Output-Active Flag
Return
No
Chars in Output Buf.? (TTOCT ≠ 0) ?
No
Clear Output-Active Flag
CLENUP
Call TTRLOB to Release Output Buffers
If Int. On Empty Output Buf. Requested, Put a Word in BIG BUF with Bit TTOIRQ Set (Sched. (TTCH7) Will Find It)
Return
Yes
Get Buf Ptr from TTOOUT. (BUGHLT if No Ptr.) & Save it
Step to Next Buf. if at End of Current Buf. & Update Ptr. in TTOOUT
Get Next 9-Bit Char. & Reduce Cnt. in TTOCT
Function Escape Char. ?
Yes
Page Full Function ?
Yes
Simulate CTRL/S (Set TTSFG Flag)
TTSND2
No
No
Get Char & Parity (Low Order 8-Bits)
BUGCHK
TTSND-Unrecognized Escape Code
TTSND6
Call Dev. Dep. Code to Send it to TTY by Dispatching to
TTXXVT
at Offset
TTVT23
(FE Disp Adr = TTSND1)
(PT Disp Adr = Ret)
Return
TTSND1
Start Up a Line Xfer via the DTE20
Primary Protocal ?
No
DTECHO
Output via Secondary
Return
Yes
Set Waiting for 11 Done & Line Active Flags
Increase Cnt. of Bytes in 11
Setup Nec. Args for Call to DTESRV
DTSNGL pDTE1
Go Do a Single Request
Succeed (RETSKP) ?
No
Decr. Bytes in-11
Restore Orig. Ptr. in TTOOUT
Incr. TTOCT
Yes
Return
INTRST:
Go, Sched. Retry
Note Not Waiting for Packet (ie Clear TTFPK)
Get Line No., Time Until Event (=0), & Disp. Adr. (=TTSN10)
TTQAD
Compute Time to Do Function for Line & Save in TTIME
Save Disp. Adr. in TROUT
Incr. TTQCNT
Set Scan Ptr. to Here if First Entry
Return

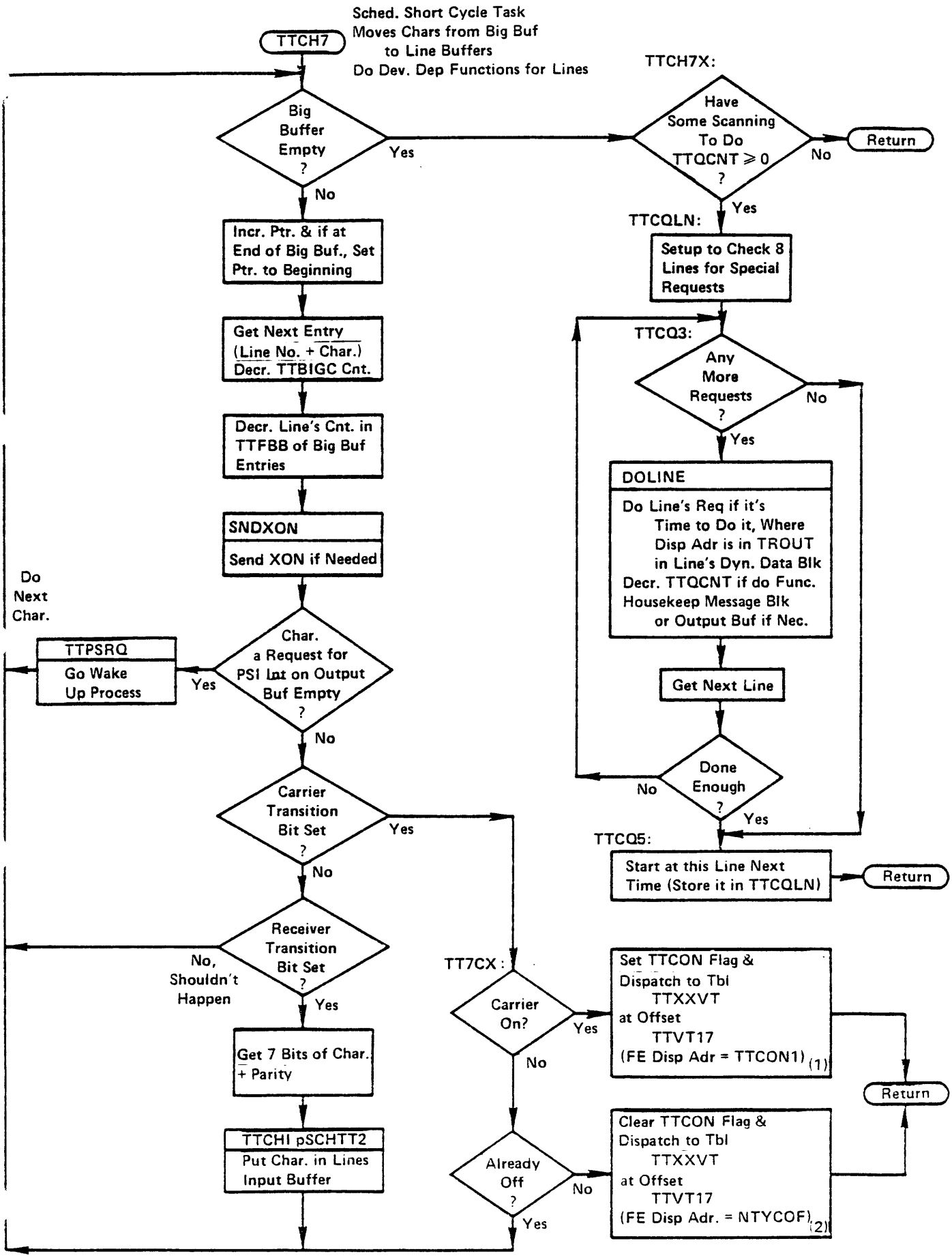
STT8

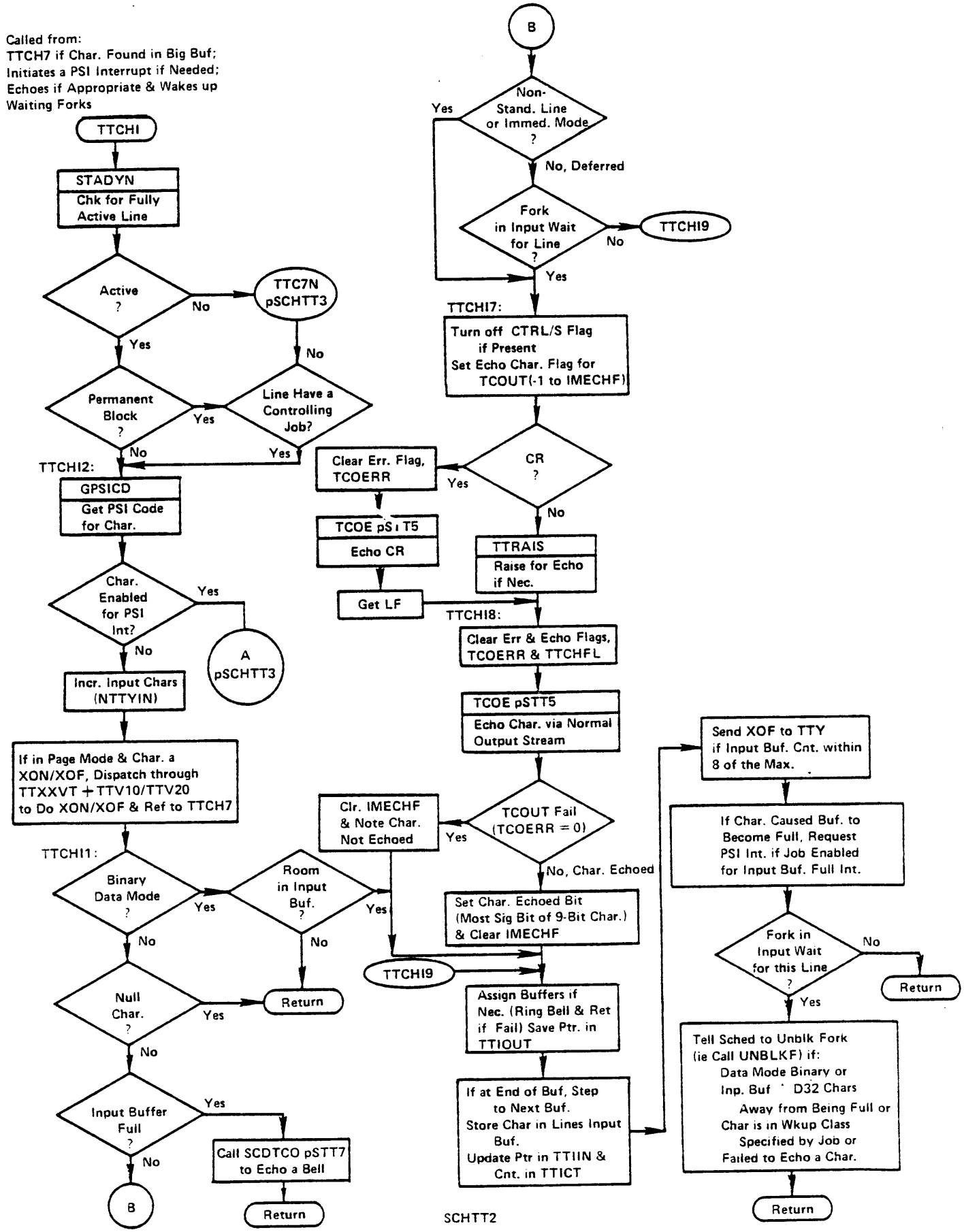

CLOSF-TTY
Dev. Dep. Code
TTYCLZ
CKJFTT
See if JFN is for "TTY:"
Is It ?
Yes
RETSKP
Successful Return
No
LCKDVL
Lock the DEV Tbls. & Go NOINT
FNDUNT
Get DEV Tbls. Index for this Line
DEVAV
See if Device is Available to this Job
Is It ?
No
Yes
GTCJOB
Get Controlling Job for this TTY
TTYCL4:
Was There One? (RETSKP?)
No
Unlock DEV Tbls. Lock, DEVLCK
Yes
OKINT
ERR RET-DEVX2
Is It This Job ?
No, Shouldn't Happen
Yes
This TTY Also Assigned ?
Yes
TTYCL1
No
A
A
TTYDEA
Deallocate Dynamic Data for this Line
Succeed? (RETSKP?)
No
TTYCL2:
Unlock DEV Tbls. Lock, DEVLCK
OKINT
If Suppose to Blk, Setup AC1/Line No.,, TSACT3 and Set BLKF Bit
Return
Unsuccessful Return
Yes
TTYCL0:
Indicate No Job Owns Line in DEVUNT Tbl.
TTYCL1
Indicate Line No Longer Opened (DV%OPN Bit) in DEVCHR Tbl.
Unlock DEV Tbls. Lock, DEVLCK
OKINT
RETSKP
Successful Return

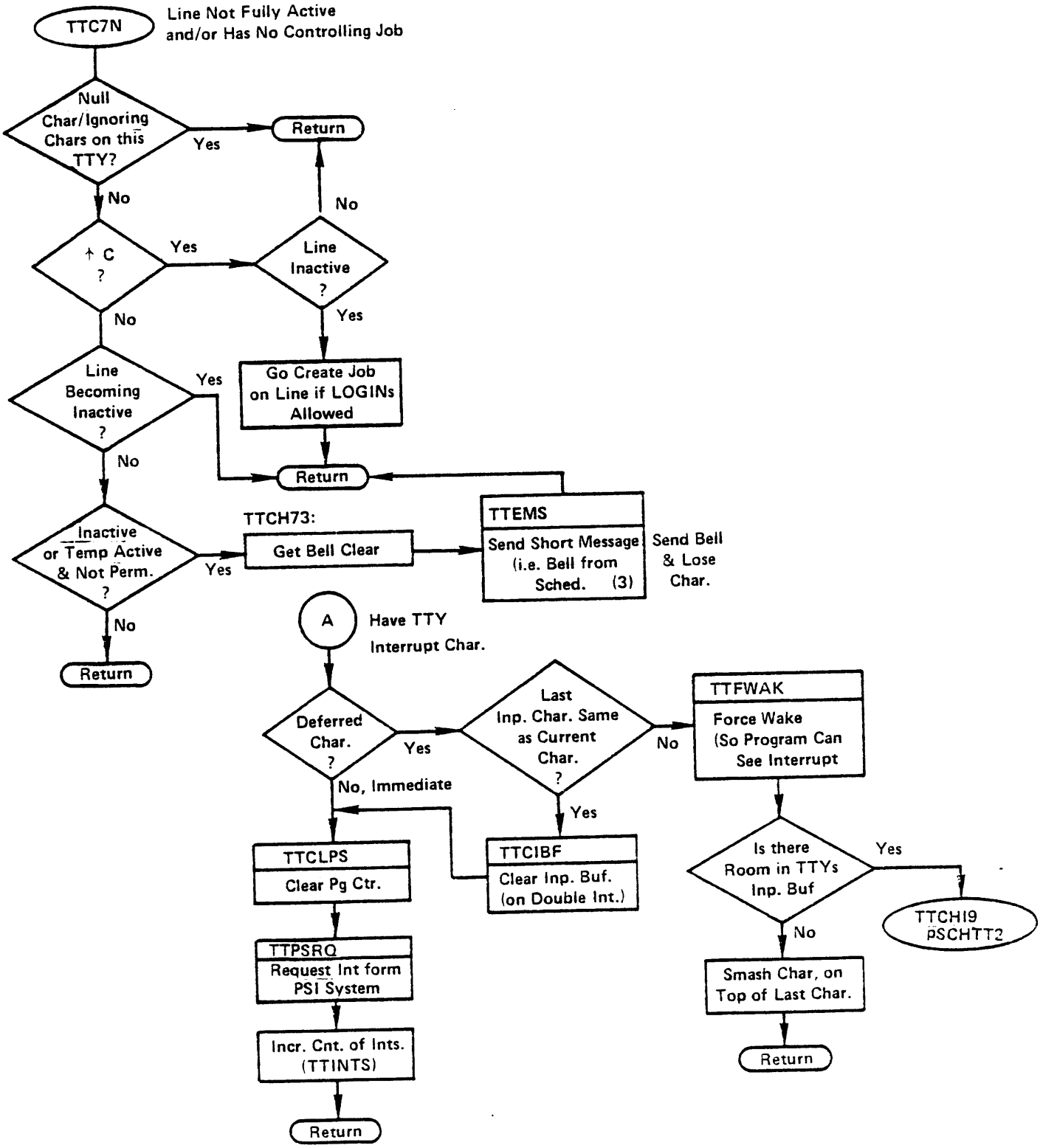
SCHEDULER TTY INPUT ANALYSIS & STORAGE

TTCH7 - Moves Characters from the Big Buffer to Line Buffers — SCHTT1

TTCHI - Initiates a PSI Interrupt if Needed, Echoes if Appropriate & Wakes Up Waiting Forks — SCHTT2


Sched. Short Cycle Task
Moves Chars from Big Buf
to Line Buffers
Do Dev. Dep Functions for Lines
TTCH7
Big Buffer Empty ?
Yes
No
Incr. Ptr. & if at End of Big Buf., Set Ptr. to Beginning
Get Next Entry (Line No. + Char.) Decr. TTBIGC Cnt.
Decr. Line's Cnt. in TTFBB of Big Buf Entries
SNDXON
Send XON if Needed
Do Next Char.
Char. a Request for PSI Int on Output Buf Empty ?
Yes
TTPSRQ
Go Wake Up Process
No
Carrier Transition Bit Set ?
Yes
No
Receiver Transition Bit Set ?
No, Shouldn't Happen
Yes
Get 7 Bits of Char. + Parity
TTCHI pSCHTT2
Put Char. in Lines Input Buffer
TTCH7X:
Have Some Scanning To Do TTQCNT ⩾ 0 ?
No
Return
Yes
TTCQLN:
Setup to Check 8 Lines for Special Requests
TTCQ3:
Any More Requests ?
No
Yes
DOLINE
Do Line's Req if it's Time to Do it, Where Disp Adr is in TROUT in Line's Dyn. Data Blk
Decr. TTQCNT if do Func.
Housekeep Message Blk or Output Buf if Nec.
Get Next Line
Done Enough ?
No
Yes
TTCQ5:
Start at this Line Next Time (Store it in TTCQLN)
Return
TT7CX:
Carrier On?
Yes
No
Set TTCON Flag & Dispatch to Tbl TTXXVT at Offset TTVT17 (FE Disp Adr = TTCON1) (1)
Already Off ?
No
Yes
Clear TTCON Flag & Dispatch to Tbl TTXXVT at Offset TTVT17 (FE Disp Adr. = NTYCOF) (2)
Return


Called from:
TTCH7 if Char. Found in Big Buf;
Initiates a PSI Interrupt if Needed;
Echoes if Appropriate & Wakes up
Waiting Forks
TTCHI
STADYN
Chk for Fully
Active Line
Active
?
No
TTC7N
pSCHTT3
Yes
Permanent
Block
?
Yes
Line Have a
Controlling
Job?
No
Yes
No
TTCHI2:
GPSICD
Get PSI Code
for Char.
Char.
Enabled
for PSI
Int?
Yes
A
pSCHTT3
No
Incr. Input Chars
(NTTYIN)
If in Page Mode & Char. a
XON/XOF, Dispatch through
TTXXVT + TTV10/TTV20
to Do XON/XOF & Ref to TTCH7
TTCHI1:
Binary
Data Mode
?
Yes
Room
in Input
Buf.
?
Yes
No
No
Null
Char.
?
Yes
Return
No
Input Buffer
Full
?
Yes
Call SCDTCO pSTT7
to Echo a Bell
No
B
Return
B
Non-
Stand. Line
or Immed. Mode
?
Yes
No, Deferred
Fork
in Input Wait
for Line
?
No
TTCHI9
Yes
TTCHI7:
Turn off CTRL/S Flag
if Present
Set Echo Char. Flag for
TCOUT(-1 to IMECHF)
CR
?
Yes
Clear Err. Flag,
TCOERR
No
TCOE pST5
Echo CR
TTRAIS
Raise for Echo
if Nec.
Get LF
TTCHI8:
Clear Err & Echo Flags,
TCOERR & TTCHFL
TCOE pSTT5
Echo Char. via Normal
Output Stream
TCOUT Fail
(TCOERR = 0)
Yes
Clr. IMECHF
& Note Char.
Not Echoed
No, Char. Echoed
Set Char. Echoed Bit
(Most Sig Bit of 9-Bit Char.)
& Clear IMECHF
TTCHI9
Assign Buffers if
Nec. (Ring Bell & Ret
if Fail) Save Ptr. in
TTIOUT
If at End of Buf, Step
to Next Buf.
Store Char in Lines Input
Buf.
Update Ptr in TTIIN &
Cnt. in TTICT
Send XOF to TTY
if Input Buf. Cnt. within
8 of the Max.
If Char. Caused Buf. to
Become Full, Request
PSI Int. if Job Enabled
for Input Buf. Full Int.
Fork in
Input Wait
for this Line
?
No
Return
Yes
Tell Sched to Unblk Fork
(ie Call UNBLKF) if:
Data Mode Binary or
Inp. Buf ' D32 Chars
Away from Being Full or
Char is in Wkup Class
Specified by Job or
Failed to Echo a Char.
Return
SCHTT2


TTC7N
Line Not Fully Active and/or Has No Controlling Job
Null Char/Ignoring Chars on this TTY?
Yes
Return
No
No
↑ C ?
Yes
Line Inactive ?
No
Yes
Line Becoming Inactive ?
Yes
Go Create Job on Line if LOGINs Allowed
No
Return
Inactive or Temp Active & Not Perm. ?
TTCH73:
Yes
Get Bell Clear
TTEMS
Send Short Message (i.e. Bell from Sched. (3)
Send Bell & Lose Char.
No
Return
A
Have TTY Interrupt Char.
Deferred Char. ?
Yes
Last Inp. Char. Same as Current Char. ?
No
TTFWAK
Force Wake (So Program Can See Interrupt
No, Immediate
Yes
TTCLPS
Clear Pg Ctr.
TTCIBF
Clear Inp. Buf. (on Double Int.)
Is there Room in TTYs Inp. Buf
Yes
TTCH19 pSCHTT2
No
TTPSRQ
Request Int form PSI System
Smash Char, on Top of Last Char.
Incr. Cnt. of Ints. (TTINTS)
Return
Return

Scheduler TTY Input Comments

TTCH7

(1) The carrier-on routine for the FE device is TTCON1. If the line is in use or a job is being created, it just returns. Otherwise, it creates a job by the CTRL/C mechanism (i.e., putting a request in Scheduler's Request Queue, SCDRQB) before returning.

(2) The carrier-off routine for the FE device is NTYCOF. It flushes outputs and issues an interrupt via the PSI system if process has enabled for carrier-off interrupt. It then issues a monitor-internal interrupt via routine, PSIR4, which causes the top fork to go to JOBCOF in MEXEC to cause the job to be detached.

TTC7N

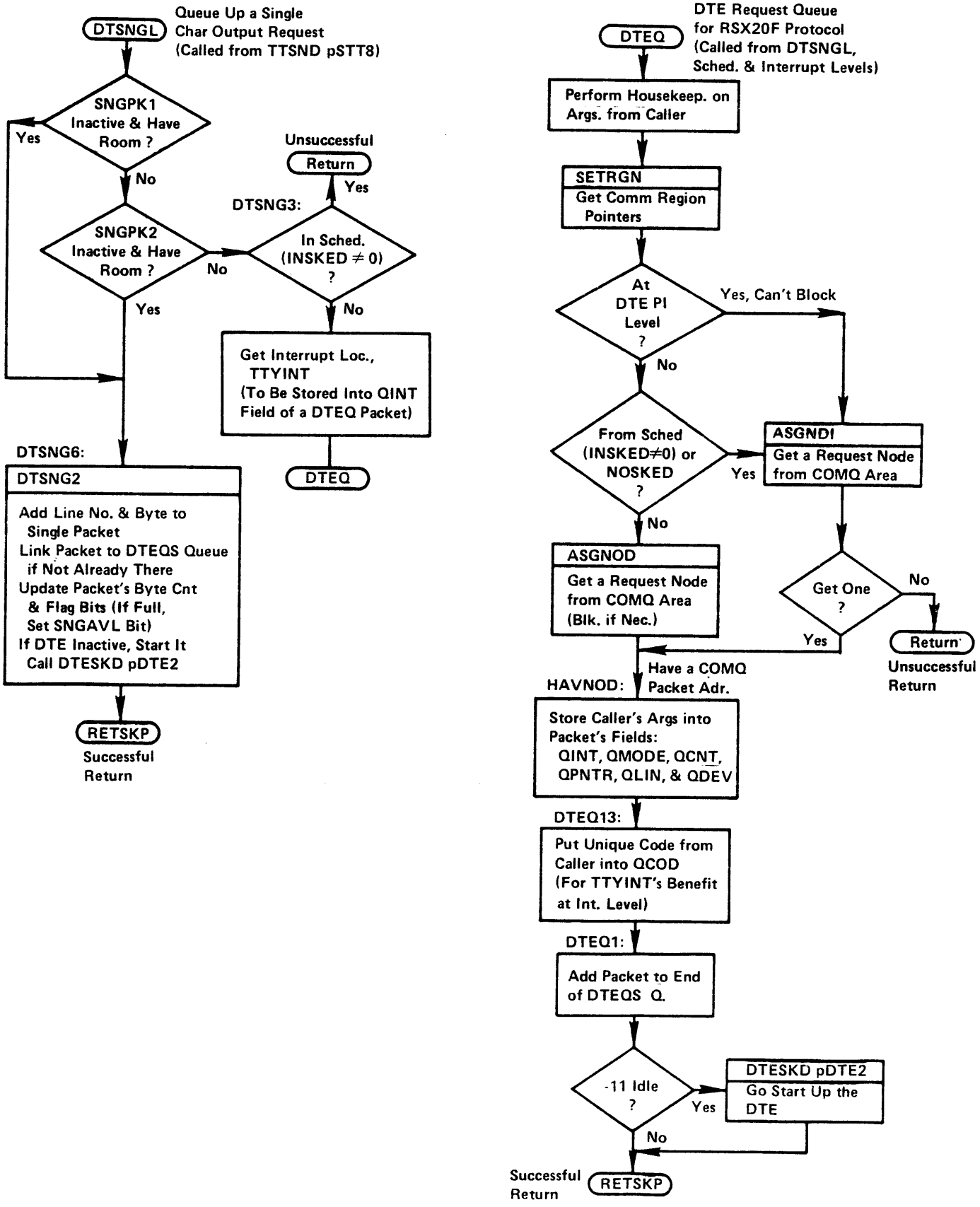
(3) TTEMES is called at Scheduler Level to send a short message to a line. If the line is active, it appends characters to the line's output buffer. If the line is not active, it creates a message-length dynamic block for the line and puts the characters into this block.

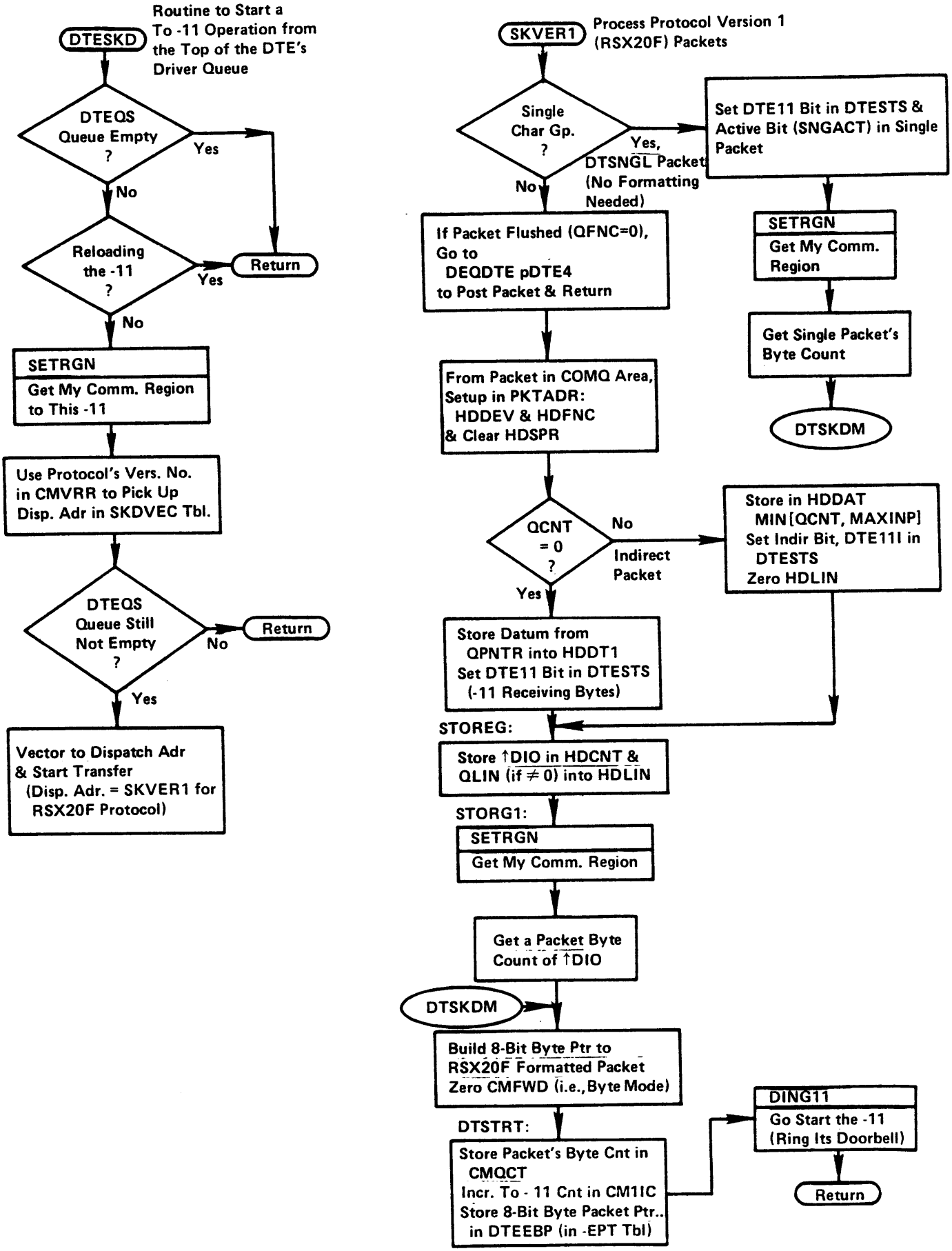
TTEMES calls SCDTCO (pSTT7) to output each character via TCOUT to the buffer or message block.

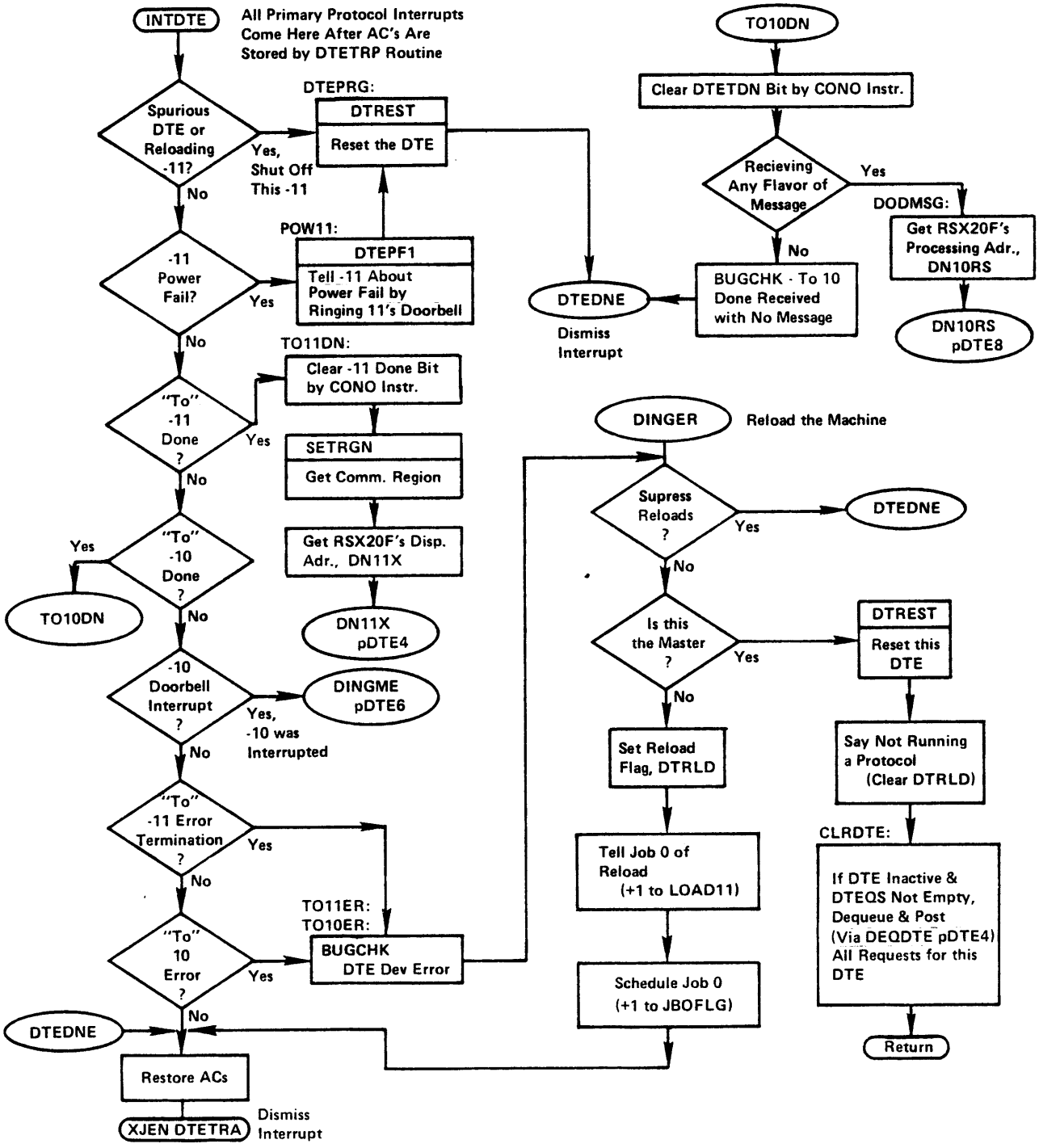
REQUESTING DTE OUTPUT & DTE INTERRUPT HANDLING FLOWCHARTS

(DTE PROTOCOL HANDLER)

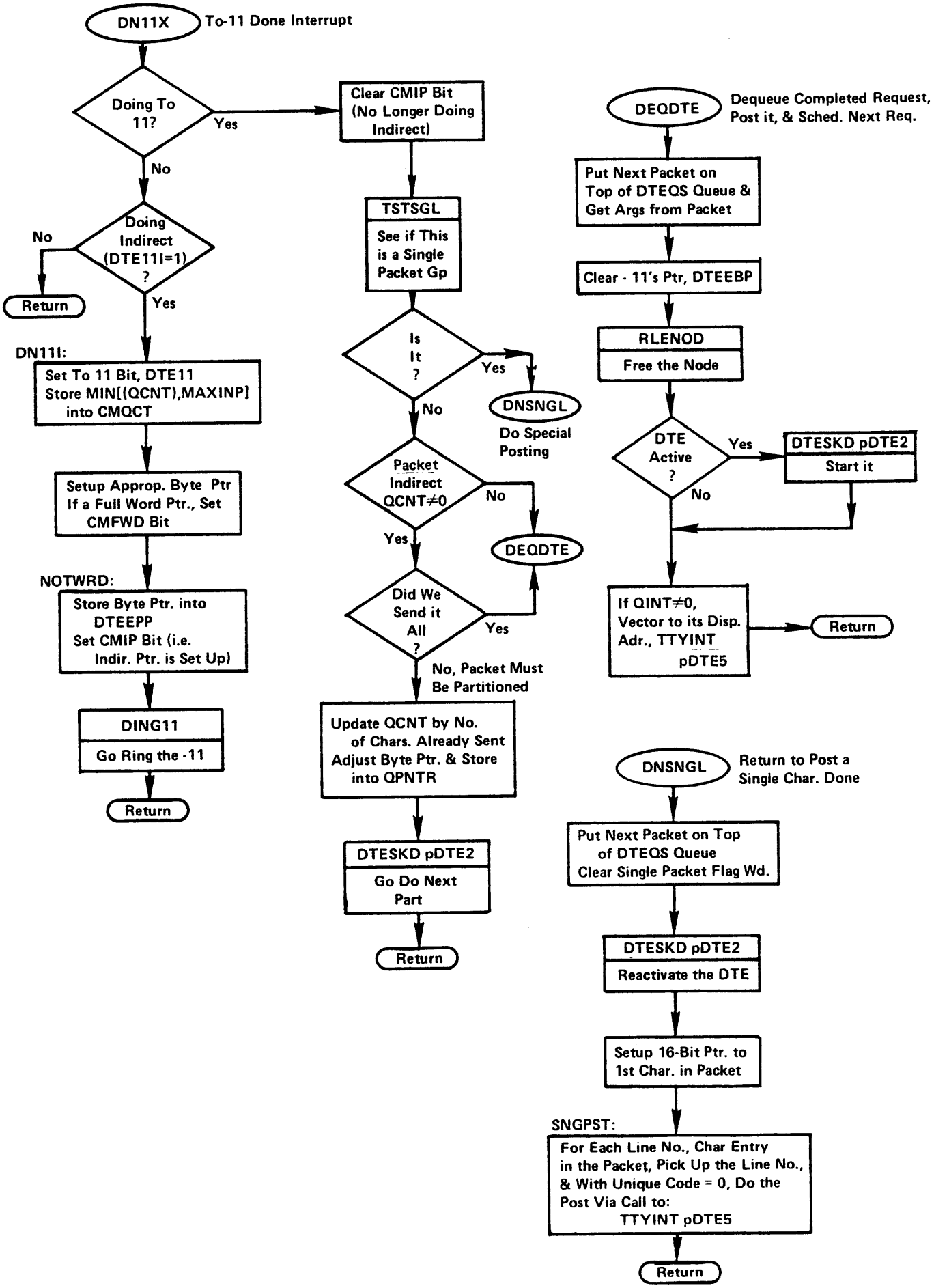
REQUESTING DTE OUTPUT
Called From Interrupt, JSYS & Scheduler Levels

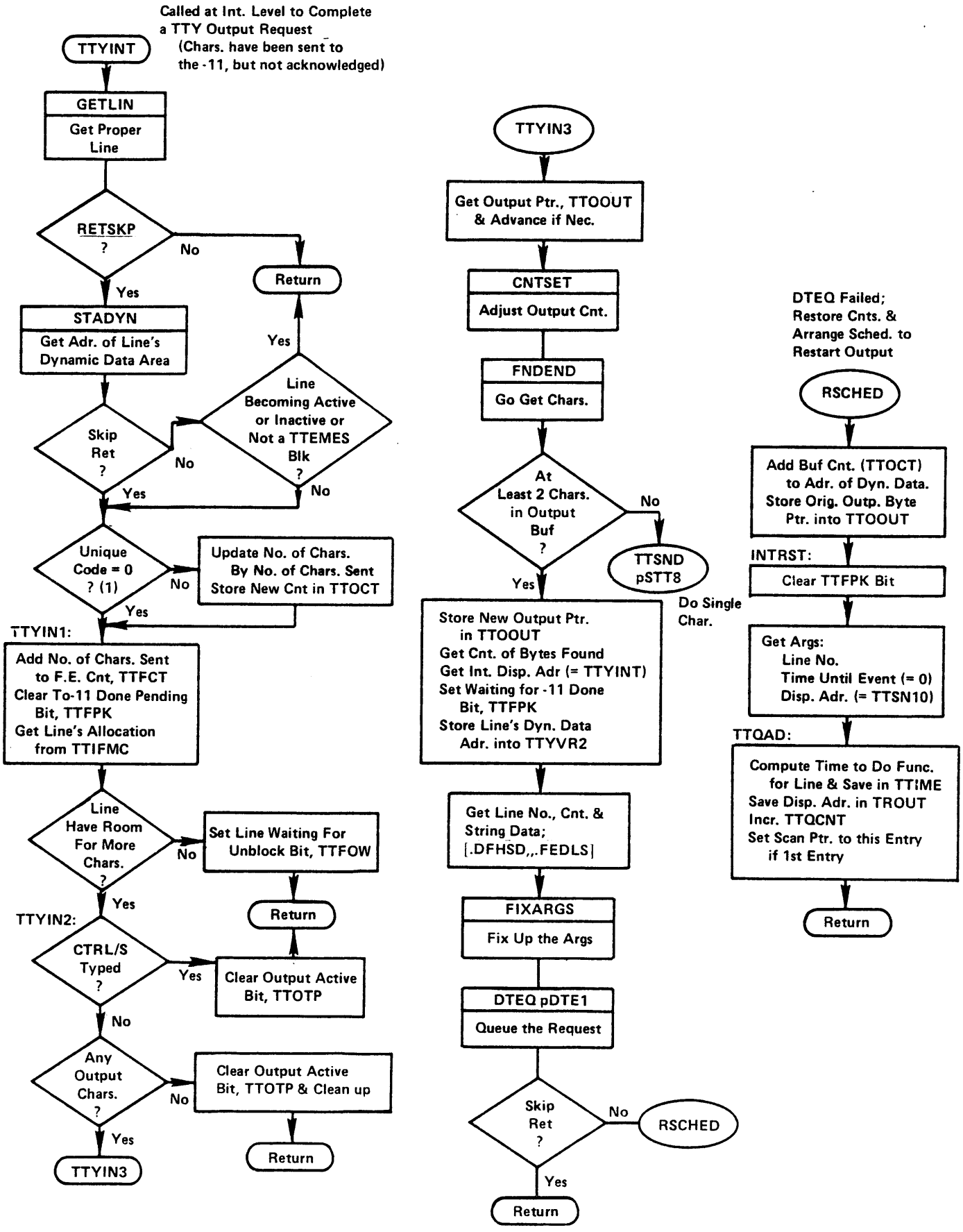



Routine to Start a To -11 Operation from the Top of the DTE's Driver Queue
DTESKD
DTEQS Queue Empty ?
Yes
No
Reloading the -11 ?
Yes
Return
No
SETRGN
Get My Comm. Region to This -11
Use Protocol's Vers. No. in CMVRR to Pick Up Disp. Adr in SKDVEC Tbl.
DTEQS Queue Still Not Empty ?
No
Return
Yes
Vector to Dispatch Adr & Start Transfer (Disp. Adr. = SKVER1 for RSX20F Protocol)
SKVER1
Process Protocol Version 1 (RSX20F) Packets
Single Char Gp. ?
Yes, DTSNGL Packet (No Formatting Needed)
Set DTE11 Bit in DTESTS & Active Bit (SNGACT) in Single Packet
No
If Packet Flushed (QFNC=0), Go to DEQDTE pDTE4 to Post Packet & Return
SETRGN
Get My Comm. Region
Get Single Packet's Byte Count
DTSKDM
From Packet in COMQ Area, Setup in PKTADR: HDDEV & HDFNC & Clear HDSPR
QCNT = 0 ?
No
Indirect Packet
Store in HDDAT MIN[QCNT, MAXINP] Set Indir Bit, DTE11I in DTESTS Zero HDLIN
Yes
Store Datum from QPNTR into HDDT1 Set DTE11 Bit in DTESTS (-11 Receiving Bytes)
STOREG:
Store ↑DIO in HDCNT & QLIN (if ≠ 0) into HDLIN
STORG1:
SETRGN
Get My Comm. Region
Get a Packet Byte Count of ↑DIO
DTSKDM
Build 8-Bit Byte Ptr to RSX20F Formatted Packet Zero CMFWD (i.e., Byte Mode)
DTSTRT:
Store Packet's Byte Cnt in CMQCT Incr. To - 11 Cnt in CM1IC Store 8-Bit Byte Packet Ptr.. in DTEEBP (in -EPT Tbl)
DING11
Go Start the -11 (Ring Its Doorbell)
Return

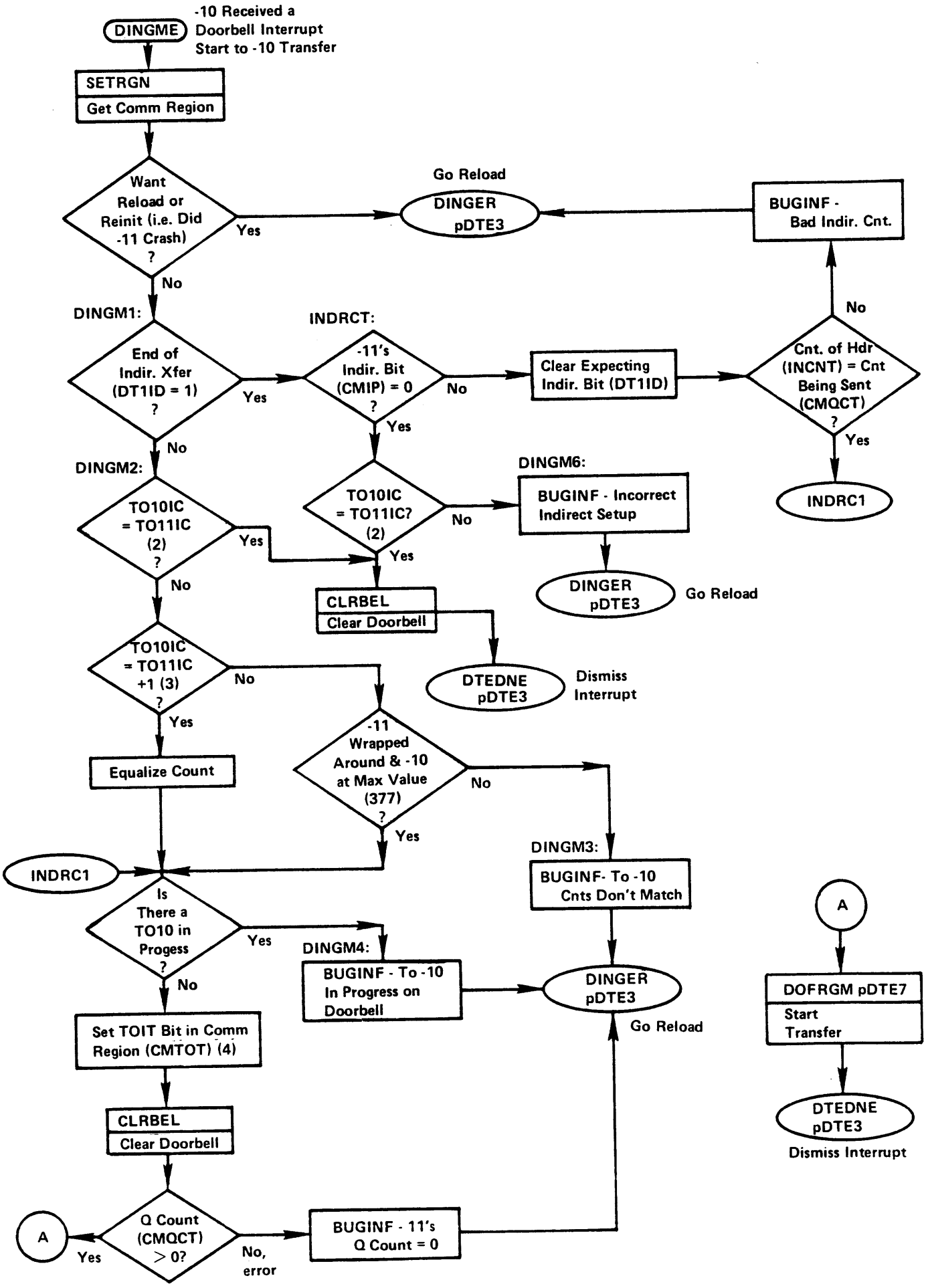

INTDTE
All Primary Protocol Interrupts Come Here After AC's Are Stored by DTETRP Routine
Spurious DTE or Reloading -11?
Yes, Shut Off This -11
DTEPRG:
DTREST
Reset the DTE
No
-11 Power Fail?
Yes
POW11:
DTEPF1
Tell -11 About Power Fail by Ringing 11's Doorbell
No
"To" -11 Done ?
Yes
TO11DN:
Clear -11 Done Bit by CONO Instr.
SETRGN
Get Comm. Region
Get RSX20F's Disp. Adr., DN11X
DN11X pDTE4
No
"To" -10 Done ?
Yes
TO10DN
No
-10 Doorbell Interrupt ?
Yes, -10 was Interrupted
DINGME pDTE6
No
"To" -11 Error Termination ?
Yes
No
"To" 10 Error ?
Yes
TO11ER:
TO10ER:
BUGCHK DTE Dev Error
No
DTEDNE
Restore ACs
XJEN DTETRA
Dismiss Interrupt
TO10DN
Clear DTETDN Bit by CONO Instr.
Recieving Any Flavor of Message
Yes
DODMSG:
Get RSX20F's Processing Adr., DN10RS
DN10RS pDTE8
No
BUGCHK - To 10 Done Received with No Message
DTEDNE
Dismiss Interrupt
DINGER
Reload the Machine
Supress Reloads ?
Yes
DTEDNE
No
Is this the Master ?
Yes
DTREST
Reset this DTE
No
Set Reload Flag, DTRLD
Tell Job 0 of Reload (+1 to LOAD11)
Schedule Job 0 (+1 to JBOFLG)
Say Not Running a Protocol (Clear DTRLD)
CLRDTE:
If DTE Inactive & DTEQS Not Empty, Dequeue & Post (Via DEQDTE pDTE4) All Requests for this DTE
Return

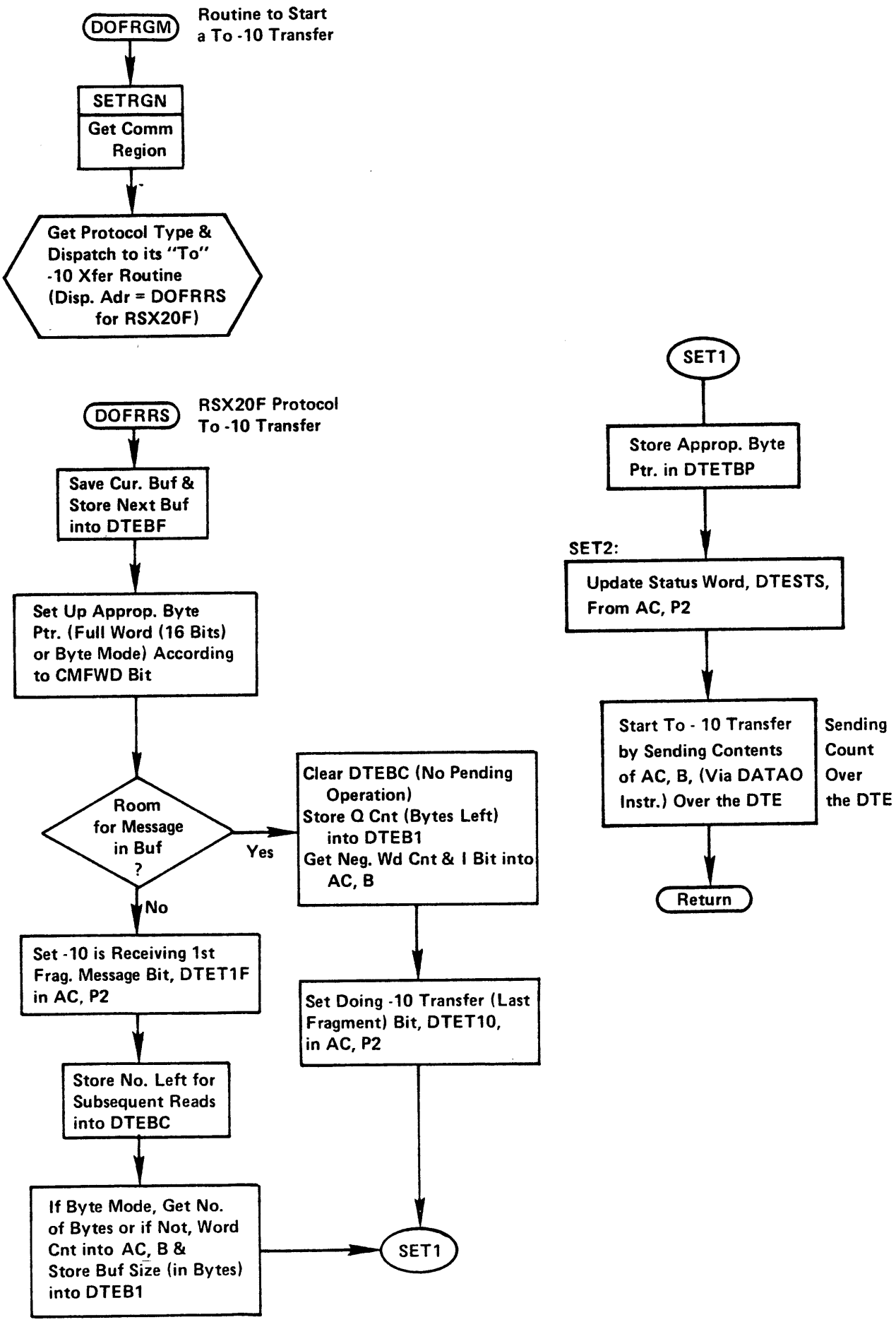
DTE3

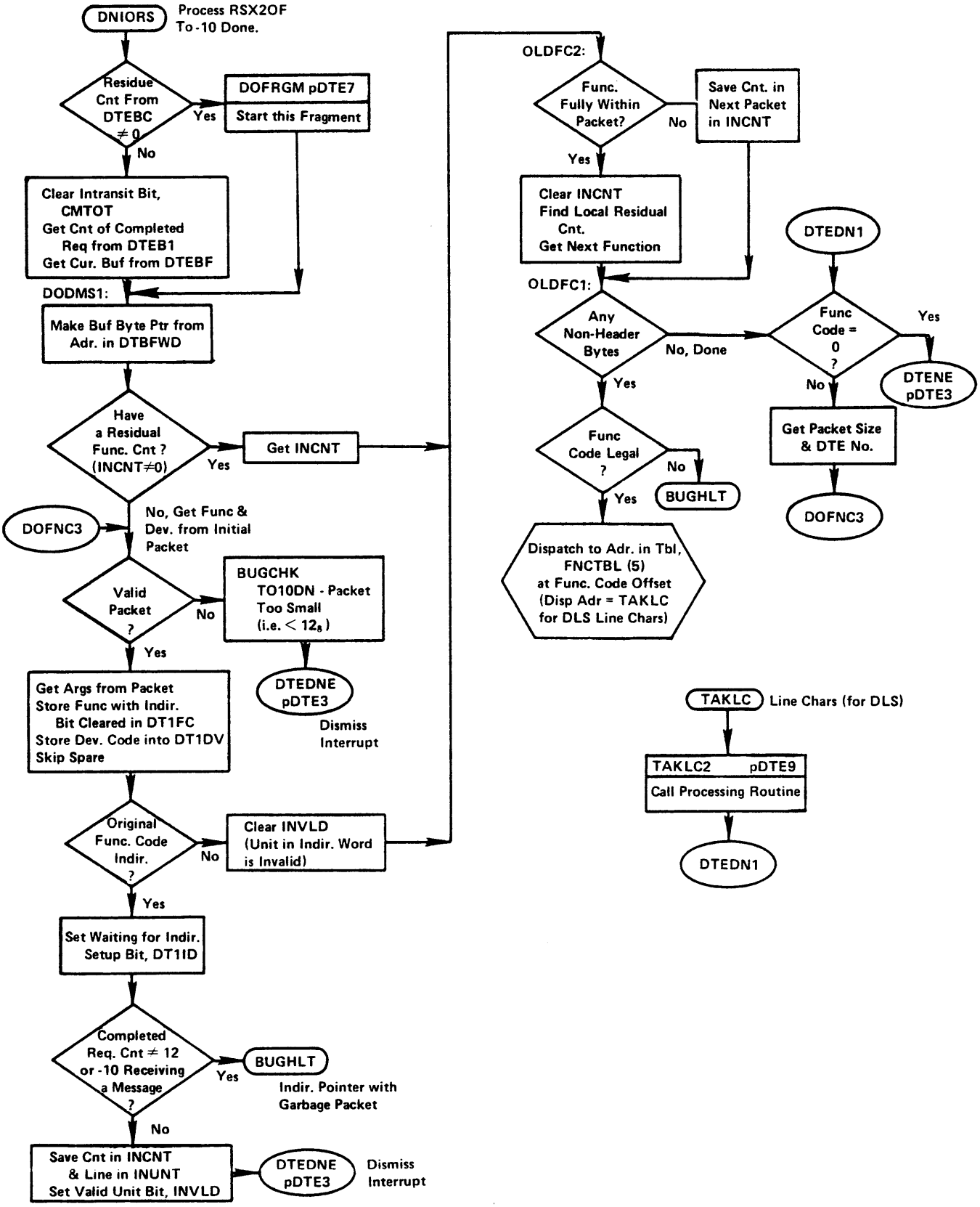

DN11X
To-11 Done Interrupt
Doing To 11?
Yes
No
Clear CMIP Bit (No Longer Doing Indirect)
Doing Indirect (DTE11I=1) ?
No
Return
Yes
DN11I:
Set To 11 Bit, DTE11
Store MIN[(QCNT),MAXINP] into CMQCT
Setup Approp. Byte Ptr If a Full Word Ptr., Set CMFWD Bit
NOTWRD:
Store Byte Ptr. into DTEEPP
Set CMIP Bit (i.e. Indir. Ptr. is Set Up)
DING11
Go Ring the -11
Return
TSTSGL
See if This is a Single Packet Gp
Is It ?
Yes
DNSNGL
Do Special Posting
No
Packet Indirect QCNT≠0
No
DEQDTE
Yes
Did We Send it All ?
Yes
No, Packet Must Be Partitioned
Update QCNT by No. of Chars. Already Sent
Adjust Byte Ptr. & Store into QPNTR
DTESKD pDTE2
Go Do Next Part
Return
DEQDTE
Dequeue Completed Request, Post it, & Sched. Next Req.
Put Next Packet on Top of DTEQS Queue & Get Args from Packet
Clear - 11's Ptr, DTEEBP
RLENOD
Free the Node
DTE Active ?
Yes
DTESKD pDTE2
Start it
No
If QINT≠0, Vector to its Disp. Adr., TTYINT pDTE5
Return
DNSNGL
Return to Post a Single Char. Done
Put Next Packet on Top of DTEQS Queue
Clear Single Packet Flag Wd.
DTESKD pDTE2
Reactivate the DTE
Setup 16-Bit Ptr. to 1st Char. in Packet
SNGPST:
For Each Line No., Char Entry in the Packet, Pick Up the Line No., & With Unique Code = 0, Do the Post Via Call to:
TTYINT pDTE5
Return


Called at Int. Level to Complete a TTY Output Request (Chars. have been sent to the -11, but not acknowledged)
TTYINT
GETLIN
Get Proper Line
RETSKP ?
No
Return
Yes
STADYN
Get Adr. of Line's Dynamic Data Area
Yes
Line Becoming Active or Inactive or Not a TTEMES Blk ?
Skip Ret ?
No
No
Yes
Unique Code = 0 ? (1)
No
Update No. of Chars. By No. of Chars. Sent Store New Cnt in TTOCT
Yes
TTYIN1:
Add No. of Chars. Sent to F.E. Cnt, TTFCT
Clear To-11 Done Pending Bit, TTFPK
Get Line's Allocation from TTIFMC
Line Have Room For More Chars. ?
No
Set Line Waiting For Unblock Bit, TTFOW
Return
Yes
TTYIN2:
CTRL/S Typed ?
Yes
Clear Output Active Bit, TTOTP
No
Any Output Chars. ?
No
Clear Output Active Bit, TTOTP & Clean up
Return
Yes
TTYIN3
TTYIN3
Get Output Ptr., TTOOUT & Advance if Nec.
CNTSET
Adjust Output Cnt.
FNDEND
Go Get Chars.
At Least 2 Chars. in Output Buf ?
No
TTSND pSTT8
Do Single Char.
Yes
Store New Output Ptr. in TTOOUT
Get Cnt. of Bytes Found
Get Int. Disp. Adr (= TTYINT)
Set Waiting for -11 Done Bit, TTFPK
Store Line's Dyn. Data Adr. into TTYVR2
Get Line No., Cnt. & String Data; [.DFHSD,,.FEDLS]
FIXARGS
Fix Up the Args
DTEQ pDTE1
Queue the Request
Skip Ret ?
No
RSCHED
Yes
Return
DTEQ Failed; Restore Cnts. & Arrange Sched. to Restart Output
RSCHED
Add Buf Cnt. (TTOCT) to Adr. of Dyn. Data. Store Orig. Outp. Byte Ptr. into TTOOUT
INTRST:
Clear TTFPK Bit
Get Args:
Line No.
Time Until Event (= 0)
Disp. Adr. (= TTSN10)
TTQAD:
Compute Time to Do Func. for Line & Save in TTIME
Save Disp. Adr. in TROUT
Incr. TTQCNT
Set Scan Ptr. to this Entry if 1st Entry
Return

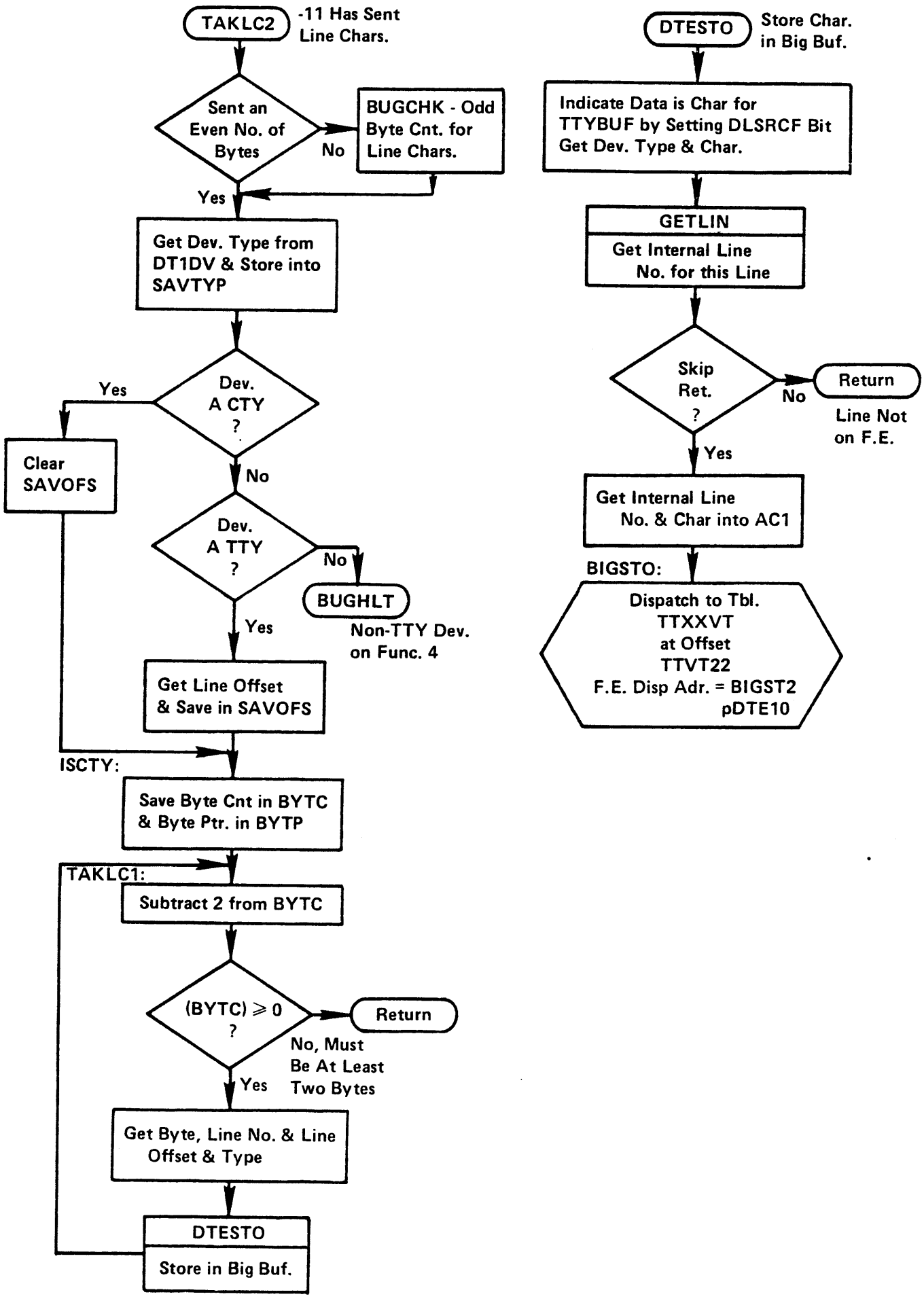
DTE5

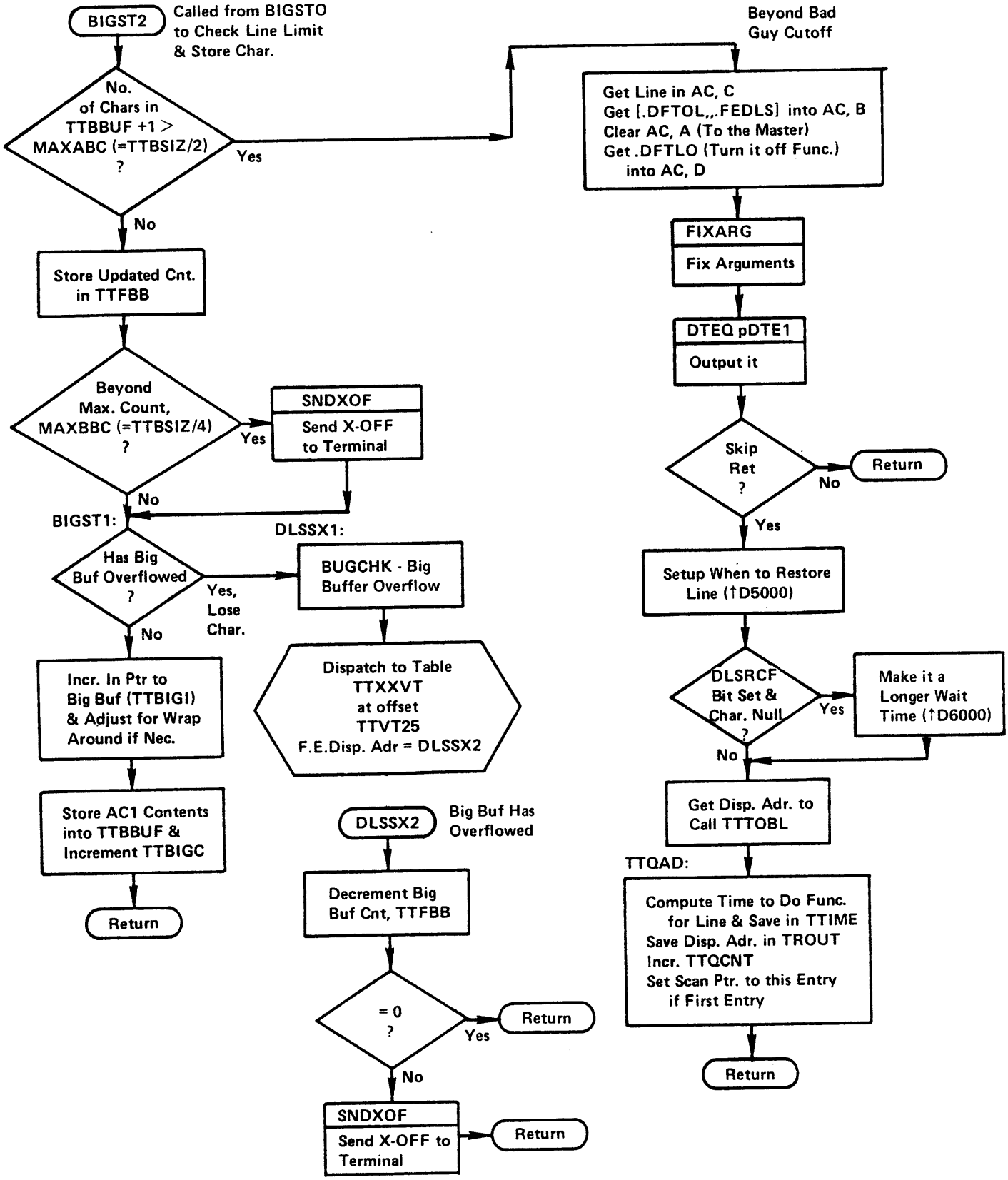

DINGME
-10 Received a Doorbell Interrupt Start to -10 Transfer
SETRGN
Get Comm Region
Want Reload or Reinit (i.e. Did -11 Crash) ?
Yes
Go Reload
DINGER pDTE3
BUGINF - Bad Indir. Cnt.
No
DINGM1:
End of Indir. Xfer (DT1ID = 1) ?
Yes
INDRCT:
-11's Indir. Bit (CMIP) = 0 ?
No
Clear Expecting Indir. Bit (DT1ID)
Cnt. of Hdr (INCNT) = Cnt Being Sent (CMQCT) ?
No
Yes
INDRC1
No
DINGM2:
TO10IC = TO11IC (2) ?
Yes
TO10IC = TO11IC? (2)
No
DINGM6:
BUGINF - Incorrect Indirect Setup
DINGER pDTE3
Go Reload
Yes
CLRBEL
Clear Doorbell
DTEDNE pDTE3
Dismiss Interrupt
No
TO10IC = TO11IC +1 (3) ?
No
Yes
-11 Wrapped Around & -10 at Max Value (377) ?
No
Yes
Equalize Count
INDRC1
DINGM3:
BUGINF- To -10 Cnts Don't Match
Is There a TO10 in Progess ?
Yes
DINGM4:
BUGINF - To -10 In Progress on Doorbell
DINGER pDTE3
Go Reload
No
Set TOIT Bit in Comm Region (CMTOT) (4)
CLRBEL
Clear Doorbell
A
Q Count (CMQCT) > 0?
Yes
No, error
BUGINF - 11's Q Count = 0
A
DOFRGM pDTE7
Start Transfer
DTEDNE pDTE3
Dismiss Interrupt


DOFRGM
Routine to Start a To -10 Transfer
SETRGN
Get Comm Region
Get Protocol Type & Dispatch to its "To" -10 Xfer Routine (Disp. Adr = DOFRRS for RSX20F)
DOFRRS
RSX20F Protocol To -10 Transfer
Save Cur. Buf & Store Next Buf into DTEBF
Set Up Approp. Byte Ptr. (Full Word (16 Bits) or Byte Mode) According to CMFWD Bit
Room for Message in Buf ?
Yes
No
Clear DTEBC (No Pending Operation)
Store Q Cnt (Bytes Left) into DTEB1
Get Neg. Wd Cnt & I Bit into AC, B
Set -10 is Receiving 1st Frag. Message Bit, DTET1F in AC, P2
Set Doing -10 Transfer (Last Fragment) Bit, DTET10, in AC, P2
Store No. Left for Subsequent Reads into DTEBC
If Byte Mode, Get No. of Bytes or if Not, Word Cnt into AC, B & Store Buf Size (in Bytes) into DTEB1
SET1
SET1
Store Approp. Byte Ptr. in DTETBP
SET2:
Update Status Word, DTESTS, From AC, P2
Start To - 10 Transfer by Sending Contents of AC, B, (Via DATAO Instr.) Over the DTE
Sending Count Over the DTE
Return


DNIORS
Process RSX2OF To-10 Done.
Residue Cnt From DTEBC ≠ 0
Yes
DOFRGM pDTE7
Start this Fragment
No
Clear Intransit Bit, CMTOT
Get Cnt of Completed Req from DTEB1
Get Cur. Buf from DTEBF
DODMS1:
Make Buf Byte Ptr from Adr. in DTBFWD
Have a Residual Func. Cnt ? (INCNT≠0)
Yes
Get INCNT
DOFNC3
No, Get Func & Dev. from Initial Packet
Valid Packet ?
No
BUGCHK
TO10DN - Packet Too Small
(i.e. < 12₈)
DTEDNE pDTE3
Dismiss Interrupt
Yes
Get Args from Packet
Store Func with Indir. Bit Cleared in DT1FC
Store Dev. Code into DT1DV
Skip Spare
Original Func. Code Indir. ?
No
Clear INVLD (Unit in Indir. Word is Invalid)
Yes
Set Waiting for Indir. Setup Bit, DT1ID
Completed Req. Cnt ≠ 12 or -10 Receiving a Message ?
Yes
BUGHLT
Indir. Pointer with Garbage Packet
No
Save Cnt in INCNT & Line in INUNT
Set Valid Unit Bit, INVLD
DTEDNE pDTE3
Dismiss Interrupt
OLDFC2:
Func. Fully Within Packet?
No
Save Cnt. in Next Packet in INCNT
Yes
Clear INCNT
Find Local Residual Cnt.
Get Next Function
OLDFC1:
Any Non-Header Bytes
No, Done
Yes
Func Code Legal ?
No
BUGHLT
Yes
Dispatch to Adr. in Tbl, FNCTBL (5) at Func. Code Offset (Disp Adr = TAKLC for DLS Line Chars)
DTEDN1
Func Code = 0 ?
Yes
DTENE pDTE3
No
Get Packet Size & DTE No.
DOFNC3
TAKLC
Line Chars (for DLS)
TAKLC2 pDTE9
Call Processing Routine
DTEDN1

DTE8


TAKLC2
-11 Has Sent Line Chars.
Sent an Even No. of Bytes
No
BUGCHK - Odd Byte Cnt. for Line Chars.
Yes
Get Dev. Type from DT1DV & Store into SAVTYP
Dev. A CTY ?
Yes
Clear SAVOFS
No
Dev. A TTY ?
No
BUGHLT
Non-TTY Dev. on Func. 4
Yes
Get Line Offset & Save in SAVOFS
ISCTY:
Save Byte Cnt in BYTC & Byte Ptr. in BYTP
TAKLC1:
Subtract 2 from BYTC
(BYTC) ⩾ 0 ?
Return
No, Must Be At Least Two Bytes
Yes
Get Byte, Line No. & Line Offset & Type
DTESTO
Store in Big Buf.
DTESTO
Store Char. in Big Buf.
Indicate Data is Char for TTYBUF by Setting DLSRCF Bit
Get Dev. Type & Char.
GETLIN
Get Internal Line No. for this Line
Skip Ret. ?
No
Return
Line Not on F.E.
Yes
Get Internal Line No. & Char into AC1
BIGSTO:
Dispatch to Tbl. TTXXVT at Offset TTVT22
F.E. Disp Adr. = BIGST2
pDTE10


BIGST2
Called from BIGSTO to Check Line Limit & Store Char.
No. of Chars in TTBBUF +1 > MAXABC (=TTBSIZ/2) ?
Yes
No
Store Updated Cnt. in TTFBB
Beyond Max. Count, MAXBBC (=TTBSIZ/4) ?
Yes
SNDXOF
Send X-OFF to Terminal
No
BIGST1:
Has Big Buf Overflowed ?
Yes, Lose Char.
No
Incr. In Ptr to Big Buf (TTBIGI) & Adjust for Wrap Around if Nec.
Store AC1 Contents into TTBBUF & Increment TTBIGC
Return
DLSSX1:
BUGCHK - Big Buffer Overflow
Dispatch to Table TTXXVT at offset TTVT25 F.E.Disp. Adr = DLSSX2
DLSSX2
Big Buf Has Overflowed
Decrement Big Buf Cnt, TTFBB
= 0 ?
Yes
Return
No
SNDXOF
Send X-OFF to Terminal
Return
Beyond Bad Guy Cutoff
Get Line in AC, C
Get [.DFTOL,,.FEDLS] into AC, B
Clear AC, A (To the Master)
Get .DFTLO (Turn it off Func.) into AC, D
FIXARG
Fix Arguments
DTEQ pDTE1
Output it
Skip Ret ?
No
Return
Yes
Setup When to Restore Line (↑D5000)
DLSRCF Bit Set & Char. Null ?
Yes
Make it a Longer Wait Time (↑D6000)
No
Get Disp. Adr. to Call TTTOBL
TTQAD:
Compute Time to Do Func. for Line & Save in TTIME
Save Disp. Adr. in TROUT
Incr. TTQCNT
Set Scan Ptr. to this Entry if First Entry
Return

DTE10

DTE Interrupt Handling Comments

TTYINT

(1) The Unique Code argument of form (Ø, count) tells TTYINT the number of characters that have been sent to the -11 in some call to DTEQ that specified TTYINT as its return address.

Count = Ø implies this was a single character (DTSNGL was called) and buffer counts have already been updated.

Count ≠ Ø implies this was multiple characters and the count must be updated.

DINGME

(2) TO1ØIC and TO11IC are wrap-around counters of Indirect Transfers where TO1ØIC is maintained by the -11 and TO11IC is maintained by the -1Ø. If the two wrap-around counters are equal, it means the transfer finished correctly.

(3) If the difference between the wrap-around counters is greater than 1, the -11 has tried to send a direct transfer before the last indirect transfer finished or a doorbell has been lost in a previous transaction.

(4) Receiver sets TOIT equal to 1 in Sender's section of Receiver's communication region after Sender sets @ or increments Q count and rings the doorbell; Receiver clears TOIT upon getting To-Receiver Done (This assures that the Receiver doesn't lose an interrupt).

(5) The function table has dispatches for such features as:

- F.E. telling about the CTY
- String data for the CDR
- Line characters (for DLS)
- -11 Sending error information
- -11 wants or is sending time of day
- Line dialed up, hung up or line buffer empty
- Set line speed or allocation
- Take -11 reload information
- Acknowledge all devices and units
- Take KLINIK data.

Monitor Tables

ALOC1
ALOC2
BALSET
BAT
BTB
CDB
CDR
CDS
CHNPIT
CHNTAB
CSTO
CST1
CST2
CST3
CST5
DEVCHR
DEVCH1
DEVCH2
DEVDSP
DEV'DTB
DEVNAM
DEVUNT
DIRECTORY
DRMBBT
DRMCNT
DSKSIZ
DSKSZ'n
DSKUTP
DST
DTE
DTEDTV
FCMODx
FDB
FE
FKCNO
FKINT
FKINTB
FKJOB
FKNR
FKPGS
FKPGST
FKPT
FKQ1
FKQ2
FKSTAT
FKTIME
FKWSP
HOM
HOME
HOMTAB
IDXFIL
INDEX
INIDEV
INIDVT
INIDV1
IORB
JOBDIR
JOBNAM
JOBPNM
JOBPT
JOBRT
JOBRTL
JSB
KDB
LPT
MTA
NAMUTP
NBQ
NBW
PHYCHT
PHYUNT
PSB
PTYSTS
SCDRQB
SDB
SNAMES
SNBLKS
SPFLTS
SPT
SPTH
SPTO
SSIZE
STARTUP
STIMES
STRTAB
TTBFRC
TTBUFS
TTCS
TTDPSI
TTExx
TTFEWD
TTFLGS
TTFORK
TTIxx
TTLINK
TTLPOS
TTMOD1
TTOxx
TTPSI
TTSPWD
TTYPE
UDB
UDIORB
UDS

Name: ALOC1

Description: Used to help enforce disk quotas for each active directory. This table is of length NOFN (size of OFN area in SPT).

Defined in: STG

Reference by: PAGEM

Format

ALOC1

ADIRN	ODIRC
Directory No.	OFN Count
.	.
.	.
.	.
.	.
.	.
.	.
.	.

Note: Each SPT entry in the OFN area contains an index into this table.

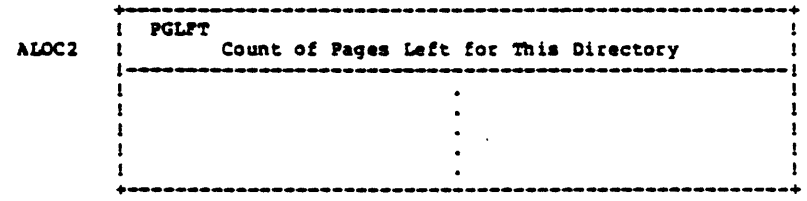
Name: ALOC2

Description: Used in disk quota enforcement for each active directory. This table is of length NOFN (size of OFN area is SPT).

Defined in: STG

Referenced by: DISC, PAGEM

Format



Note: Each SPT entry in the OFN area contains an index into this table.

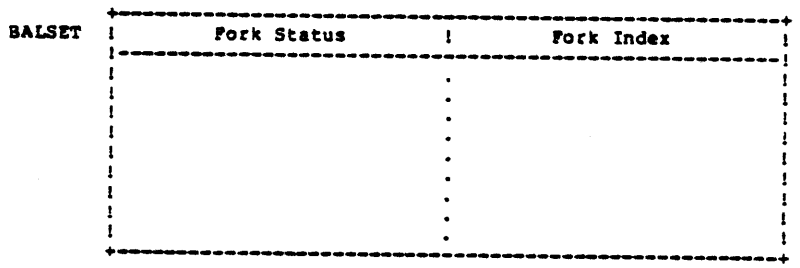
Name: BALSET

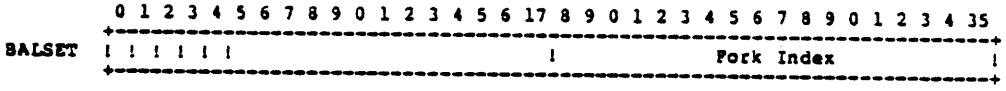
Description: Balance Set Table. This table contains the set of most eligible forks for CPU service whose combined working set sizes are balanced with the amount of physical core available. Only forks in this table can be chosen to run. Position in this table is arbitrary and has no effect on run priority (Position on GOLST determines this).

Defined in: SCHED

Referenced by: PAGEM

Format





Symbol	Bits	Content
BSWTB	0	If 1, fork waiting for I/O (disk or drum)
BSNSK	1	If 1, fork is NOSKED (no scheduling of other forks allowed) or NOSWAP
BSNUL	2	If 1, free BALSET slot (Deleted entry)
BSHLD	4	If 1, fork being held in Balance Set

BAT BL 1 = 2 Sector NUMBER of first bad block

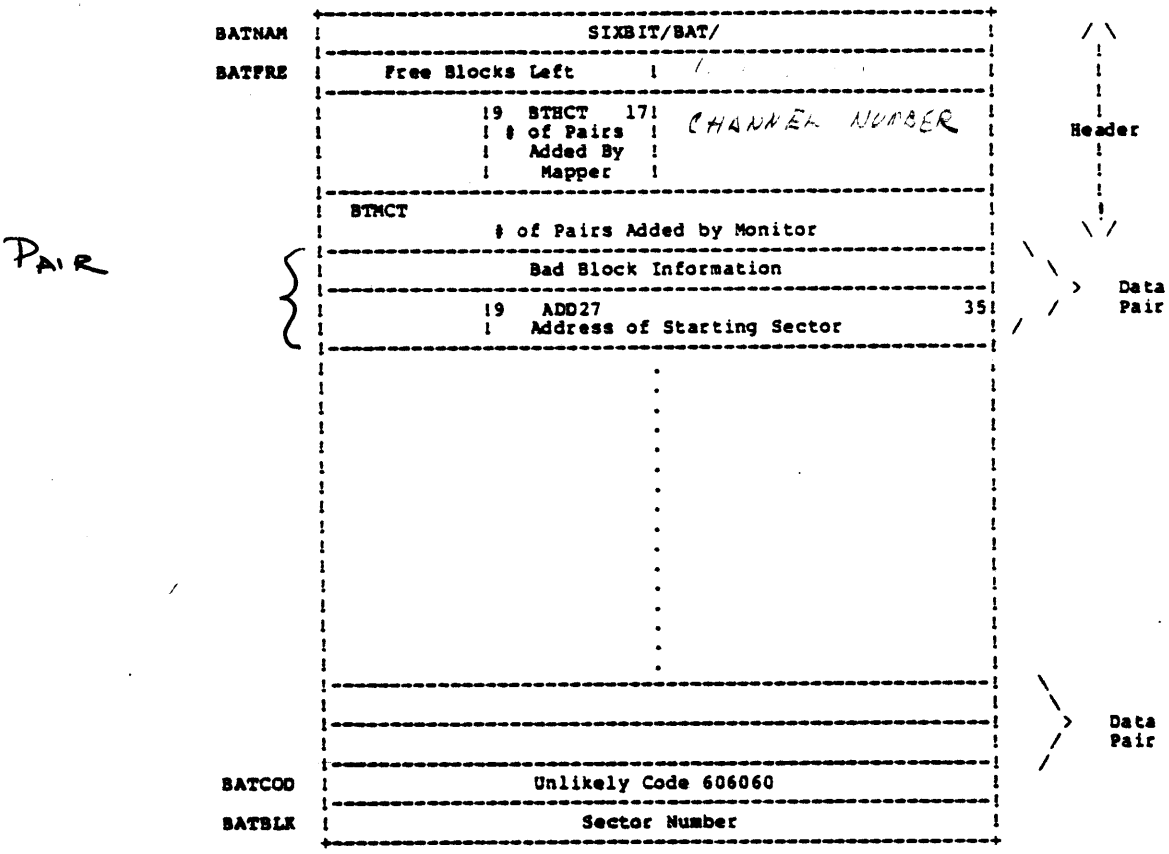
Name: BAT

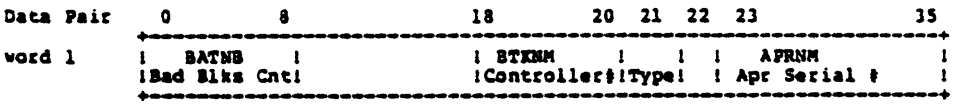
Description: Bad Allocation Table. The BAT Block is one sector in length (128 words). It consists of 4 words of header, followed by data; each 2 word data entry indicates the bad spots on the disk.

Defined in: PROLOG

Referenced by: DSKALC, DISC, DEVICE, PHYH2

Format





	Bits	Pointer	Content
	0-8	BATNB	Count of Bad Blks in Pair
	18-20	BTKNB	Massbus Controller #
	21	BADT	Type field in BAT Pair
	23-35	APRNM	APR Serial #
word 2			
	Bits		
	18-35	ADD18	Old style disk address of starting sector
	9-35	ADD27	New style address of starting sector

Name: BTB

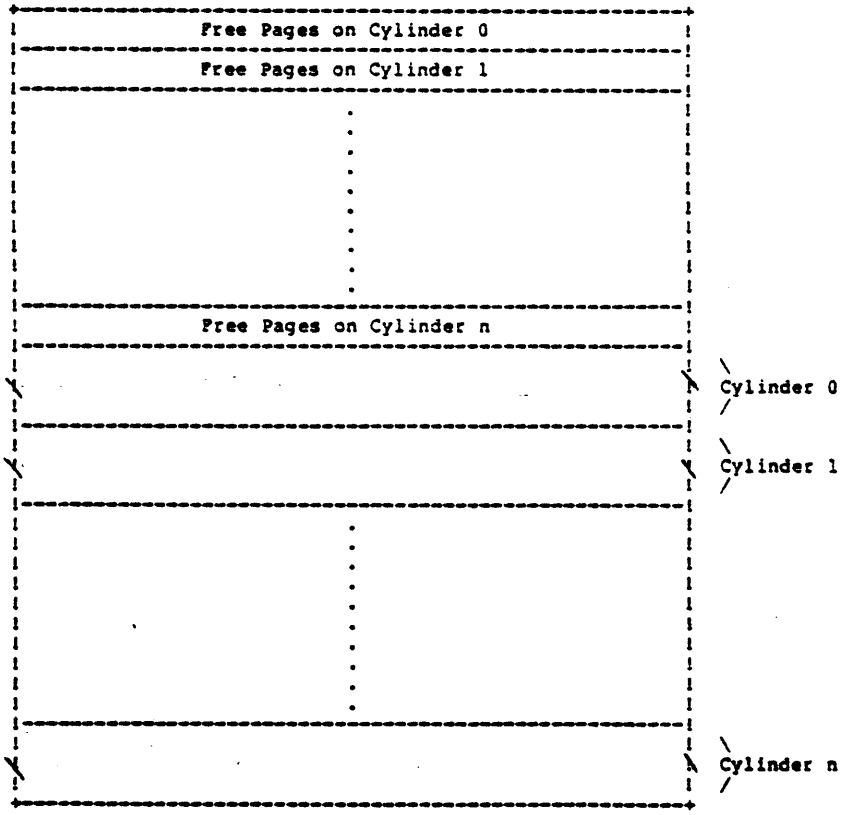
Description: Bit Table for Disk. This table has mapped into it, STRNAM:<ROOT-DIRECTORY>DSKBTTBL, when pages are allocated or deallocated from the disk unit(s) belonging to structure, STRNAM. The bit table indicates which pages are assigned (bits off) and which are available (bits on).

It consists of two parts; the top half contains the number of free pages for each cylinder in the structure and the bottom half contains a bit map (1 bit per page) for all pages of each cylinder in the structure.

Defined in: STG

Referenced by: DSKAL

Format



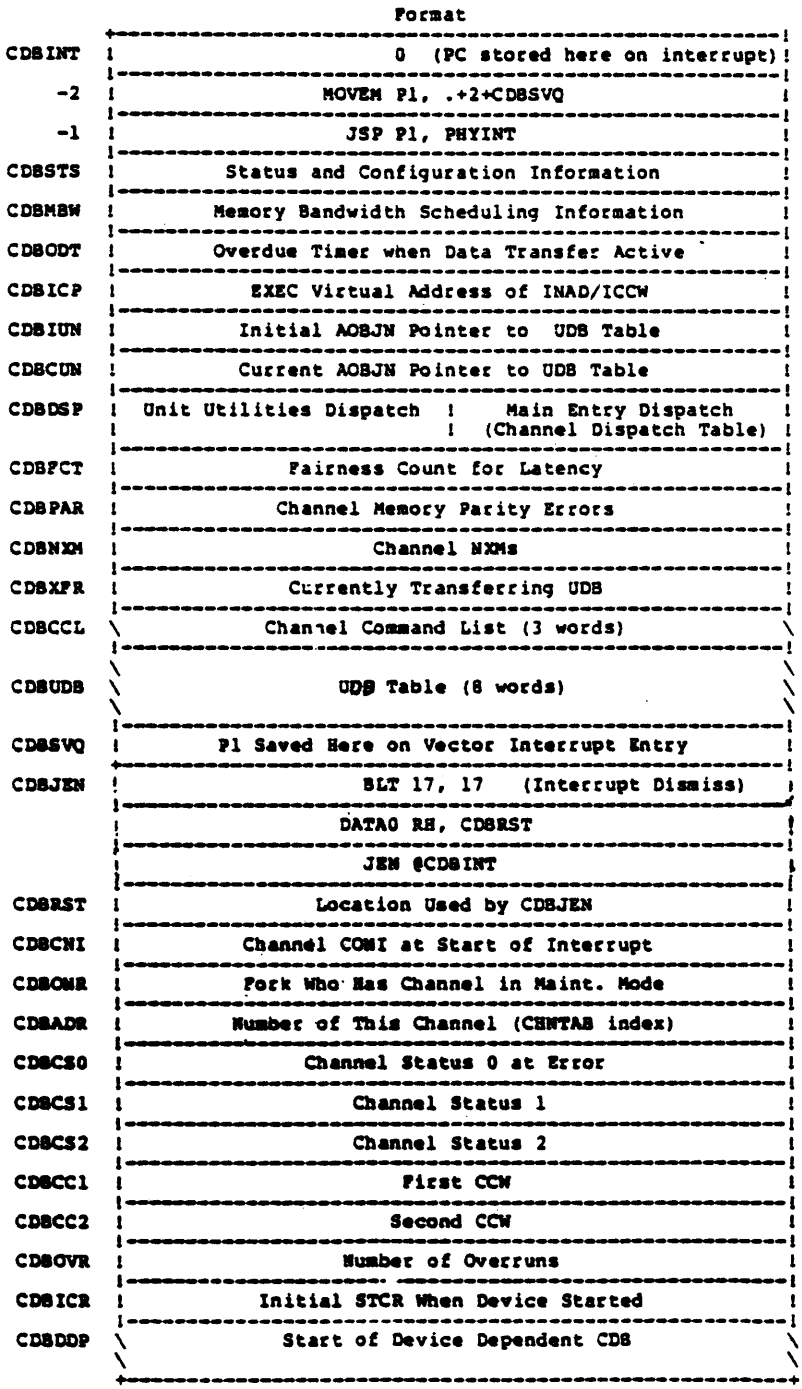
Note: In the bit map each cylinder starts on a word boundary and contains as many full words as are needed for all of its pages.

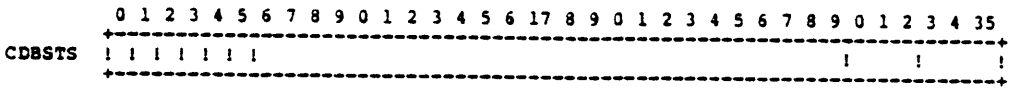
Name: CDB

Description: Channel Data Block. This table, one per channel, contains channel dependent instructions and data, pointers to the units (i.e. UDBs) belonging to the channel and information about the currently active unit. When the channel interrupts, control passes (via a JSP instruction) to CDBINT. The CDB address is stored in AC, P1, and the principal analysis routine, PHYINT, is called.

Defined in: PHYPAR

Referenced by: PHYSIO, PHYH2, PHYM2, PHYP4





Symbol	Bits	Content
CS.FOL	0	Offline
CS.AC1	1	Primary command active
CS.AC2	2	Secondary command active
CS.MAI	3	Channel is in maint. mode
CS.MRQ	4	Maint. mode requested for some unit
CS.ERC	5	Error recovery in progress
	30-32	PIA field
	33-35	Channel type field

CDBDSP

See Tables UDS and CDS

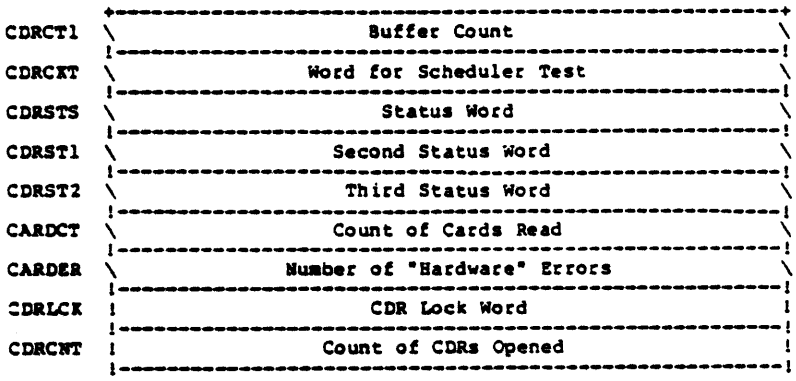
Name: CDR

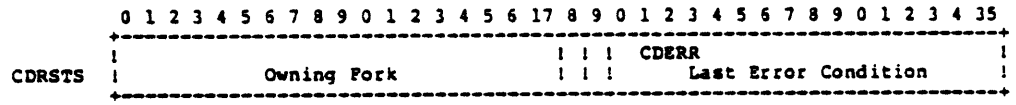
Description: Storage area for card readers (physical). Each entry (except for CDRLCK and CDRCNT) is CDRN words long where CDRN equals the number of card readers on the system.

Defined in: STG

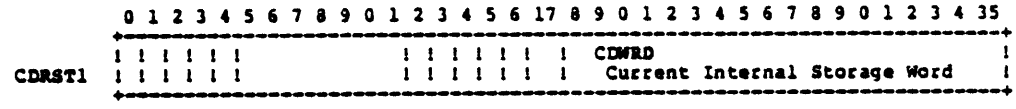
Referenced by: CDRSRV

Format

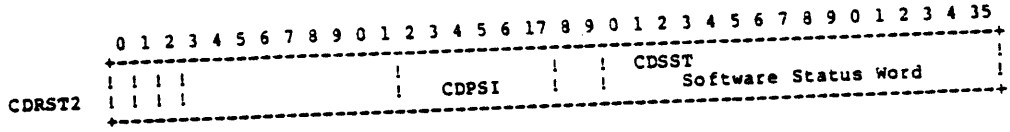




Bits	Pointer	Content
0-17		Owning fork
18	CDOL	If one, cards in reader
19	CDBLK	Waiting for a card
20-35	CDERR	Last error condition



Bits	Pointer	Content
0	CDAII	CDR opened in ASCII
1	CDATN	CDR needs attention
2	CDMSG	Suppress system messages
3	CDOPN	CDR is open
4	CDER	Error in this CDR
12	CDCNT	Count of bytes now in buffer
13	CDEOF	EOF button was pushed
14	CDBUF	Buffer for process level
15	CDPIR	Process needs interrupt
16	CDBFI	Buffer for PI level
17	CDDON	If one, doing a buffer by process



Bits	Pointer	Content
0	CDSHA	"Status has arrived" flag
1	CDMWS	MTOPR is waiting for status to arrive
2	CDRLD	Front end has reloaded
12-17	CDPSI	PSI chan. no. for on-line transitions
20-35	CDSST	Software status word

Symbol	Bits	Content
.DVFFE	28	Device has a fatal, unrecoverable error
.DVFLG	29	Error logging information follows
.DVFEF	30	EOF
.DVFIP	31	I/O in progress
.DVFSE	32	Software condition
.DVFHE	33	Hardware error
.DVFOL	34	Offline
.DVFNX	35	Nonexistent device

Name: CDS

Description: Channel Dispatch Service Routine Table. This table contains vectored addresses to channel dependent functions, and is given in its generalized form. The specific channel dispatch table for the RH20 begins at RH2DSP in PHYH2. See PHYPAR for definitions of arguments given and returned on calls to these channel routines.

Defined in: PHYPAR

Referenced by: PHYH2, PHYM2, PHYP4, PHYSIO, STG

Format

CDSINI	Initialize and Build Data Structure
	*** Unused ***
CDSSIO	Start I/O on IORB (skipsif started O.K.)
CDSPOS	Do Positioning to Idle Unit (skips if O.K.)
CDSLTM	Return Latency and Best Request (i.e. best IORB)
CDSINT	Interrupt Entry
CDSCCW	Generate Single CCW Entry
CDSHNG	Hung Reset
CDSRST	On Restart, Reset Channel and All Devices
CDSCHK	Periodic Check Entry, PIA, etc.

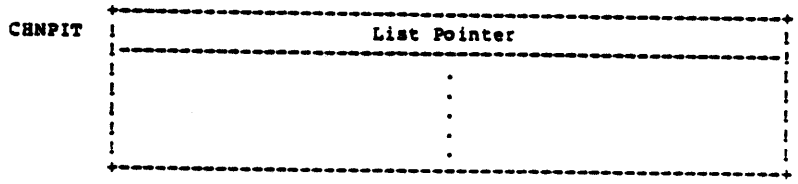
Name: CHNPIT

Description: Channel Priority Interrupt Table. This table contains the head of the list of channels on each PI level indexed by priority interrupt level.

Defined in: STG

Referenced by: STG

Format



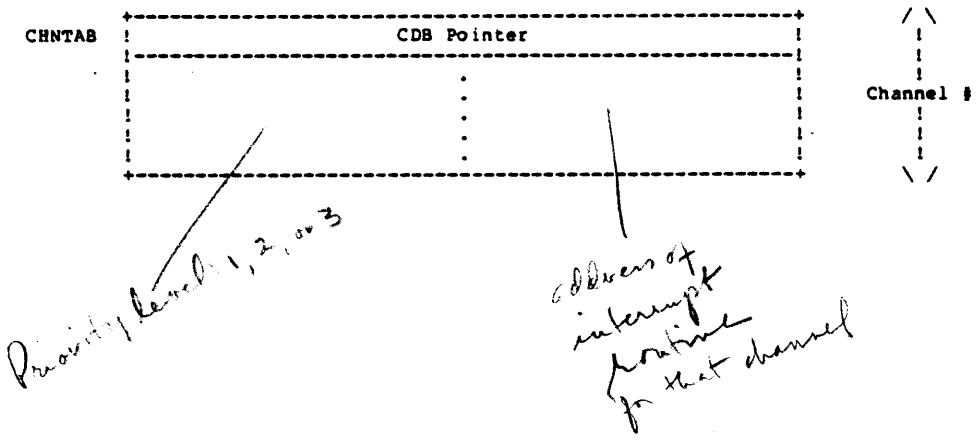
Name: CHNTAB

Description: Channel Table, indexed by channel number, contains channel data block (CDB) pointers.

Defined in: STG

Referenced by: DSKALC, PHYH2, PHYSIO

Format



PAGE STATE Codes (CSTQ - CSTAGE FIELD)

PSRPQ = 0 REPLACABLE QUE
PSDEL = 1 DELETED (ON DELETED QUE)
PSRDN = 2 READ COMPLETED
PSWIP = 4 WRITE IN PROGRESS
PSRIP = 6 READ IN PROGRESS
PSSPQ = 7 PAGE IN SPECIAL MEMORY QUE
PSASN = 10 PAGE ASSIGNED to PROCESS if .GE. PSASN

CSTPST

PSTAVL = 0 AVAILABLE FOR RPLQ WHEN FREED (.MCPSA)
PSTSPM = 1 PLACE ON SPMQ WHEN FREED (.MCPSS)
PSTOFL = 2 OFFLINE - ACTION AS PSTSPM (.MCPSO)
PSTERR = 3 OFFLINE DUE TO ERROR ACTION AS PSTSPM (.MCPSE)

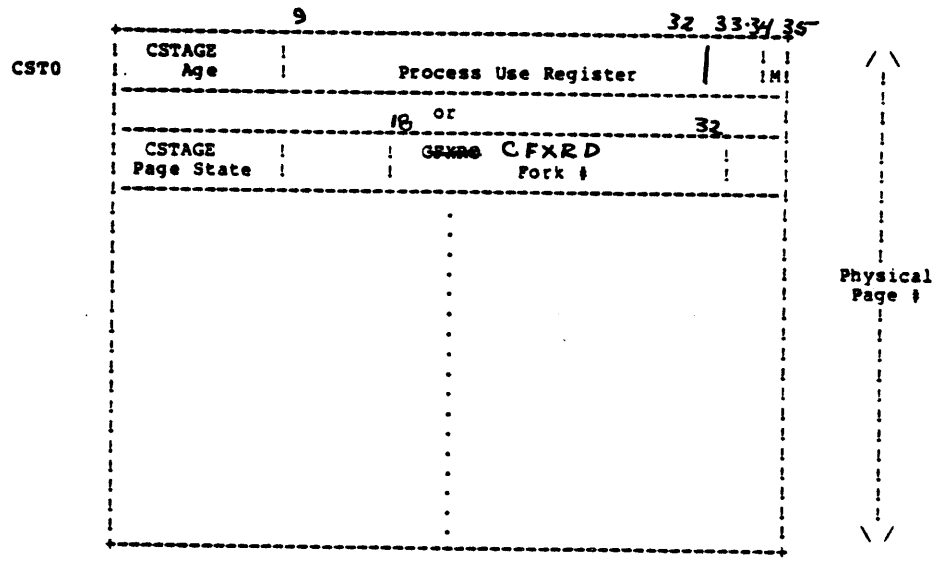
Name: CST0

Description: Core Status Table 0 (sometimes referred to as CST). Each entry in this table, indexed by physical page #, is principally defined by the pager. If the page is in use, the entry contains the age stamp for the page, which processes have referenced it and whether the page has been modified. The age stamp field is used to show the page's state if it is not assigned to a process.

Defined in: STG (MASKS - etc IN PROLOG)

Referenced by: APRSRV, DSKALC, PAGEM

Format



Bits	Pointer	Content
0-8	CSTAGE	If page in use, contents of pager age register (>= 100) at last age register reload
9-32		Process use register if age field indicates page is in use (i.e., age >=100). Bit n is 1 if process with core number n has referenced it.
33-34	CSTPST (DEF BELOW)	
35	CORMB	This is the "modified" bit which is set by the pager on any write reference. This bit will be 1 if the page has been written since the last operation.

Bits	Pointer	Content
0-8	CSTAGE	If page not in use, this field indicates (right-justified) the page state as follows:

PSRPQ = 0	On replaceable queue
PSDEL = 1	To be put on replaceable queue
PSRDN = 2	Read completed
PSWIP = 4	Write in progress
PSRIP = 6	Read in progress
PSSPQ = 7	Page on special memory queue
PSASN = 10	Page assign to process if age field >= PSASN. (The age field should always be strictly greater than 10 since it is initialized to to 100 and increases in value as the process runs.)

15-32	CFXRD	Number of fork which initiated read if page not in use (i.e. age field < 10).
33-34	CSTPST	Special page state
35	CORMB	

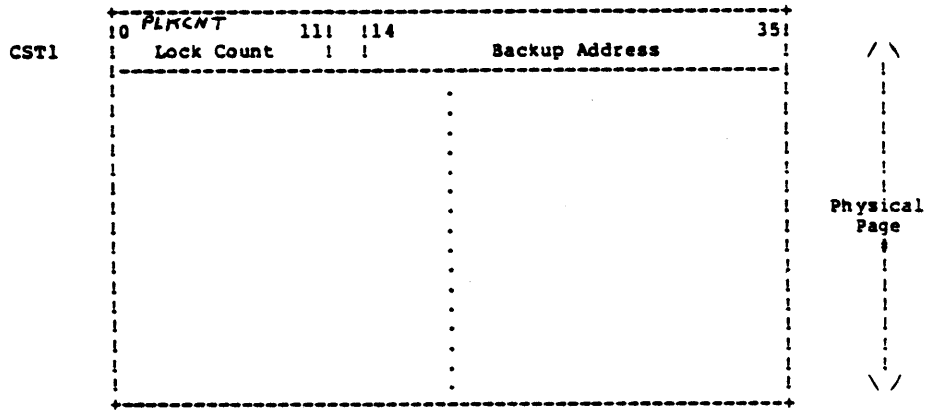
Name: CST1

Description: Core Status Table 1. This table, indexed by physical core page number, is referenced only by the software and is parallel to CST0. It contains the lock count which indicates the number of system events requiring the page be locked in core (i.e., page table contains other core addresses) and the backup address (next level of storage) for each page in core (1000000 if unassigned).

Defined in: STG (MASK etc Defined in PROLOG)

Referenced by: PAGEM, PHYSIO, SCHED

Format



Note: If the lock count is non-zero, the page will not be considered for swapping.

PLKCNT = Lock COUNT MASK

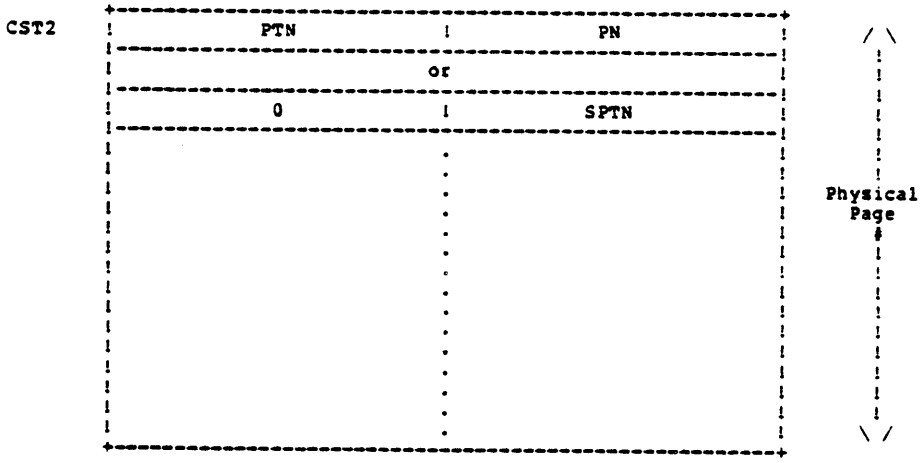
Name: CST2

Description: Core Status Table 2 (Home Map Location). This table, indexed by physical page number, is referenced only by the software and is parallel to CST0. It contains the home map location for the page (i.e., the page table which contains the core address pointing to the page). If the left half is 0, the home map is the SPT and the right half contains the SPT index. If the left half is not 0, the home map is a page table or index block, where PTN is the SPT index of that map and PN is the page number within that map.

Defined in: STG

Referenced in: PAGEM

Format



Name: CST3

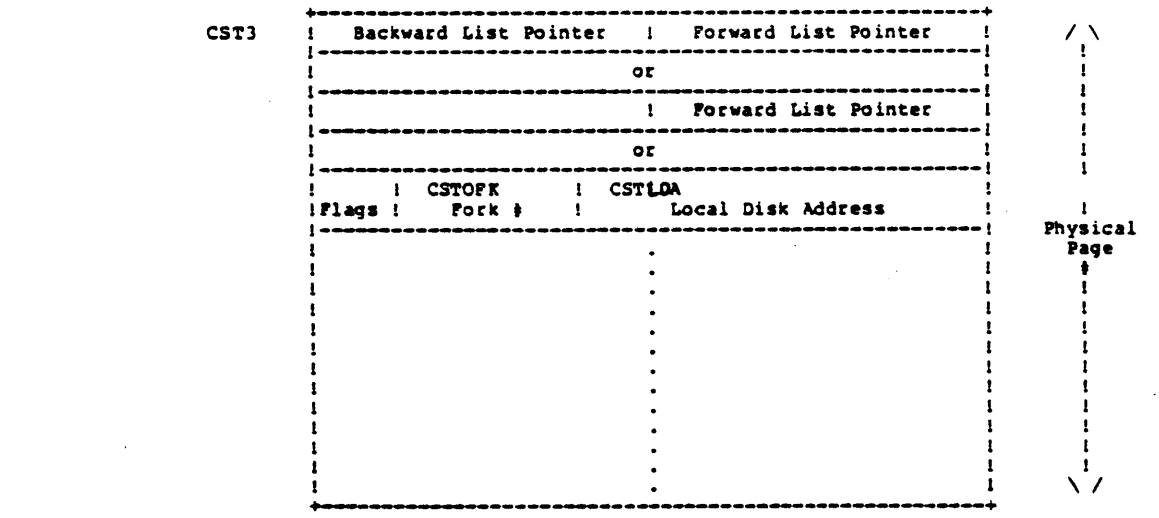
Description: Core Status Table 3. This table, indexed by physical core page number, is referenced only by the software and is parallel to CST0. An entry in this table is used for a variety of purposes, generally as a list pointer for groups of pages on various queues. For example, when on the replaceable queue, the left half and right half contain backward and forward list pointers, respectively. When on a swapping device queue, the right half contains a forward list pointer and B0 is 1 if write and 0 if read.

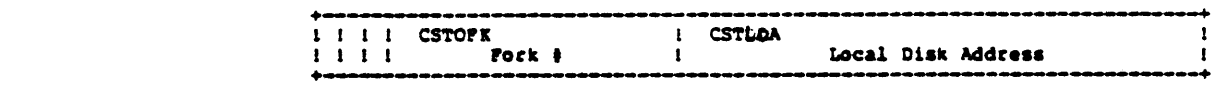
When the page is in use (not linked on one of the queues) it contains the local disk address for PHYSIO and the fork # assigned to the page.

Defined in: STG

Referenced by: PAGEM, PHYSIO, SCHED

Format





Symbol	Bits	MASK Pointer	Content
DWRBIT	0		Write in progress. This bit is 1 if the page was referenced and assigned while a write to the swapping storage was in progress. The bit is cleared by the swapper when the write completes.
SWPERR	1		Set if an unrecoverable error occurred when reading in this page from disk/drum
DSKSWB	2		Swap to disk requested by DDMP (periodic routine that trickles file pages to the disk) or by monitor when certain monitor calls are issued, e.g., CLOSF
-	3-14	CSTOFK	Process to which this page is assigned (7777 if not assigned).
	15-35	CSTLDA	Local disk address

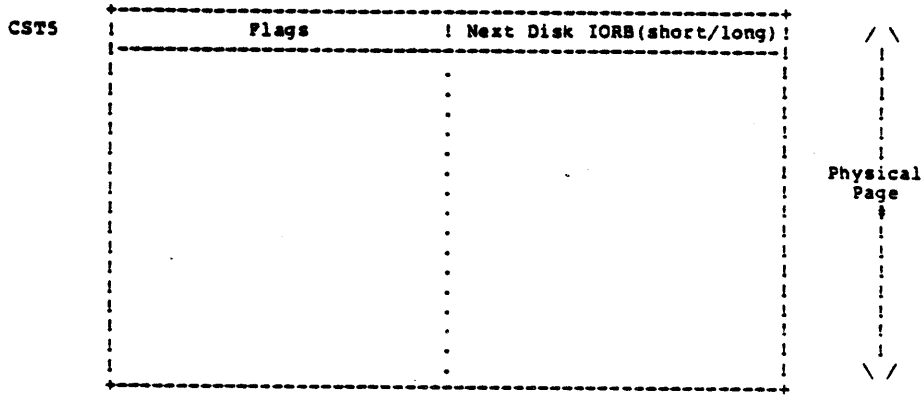
Name: CST5

Description: Core Status Table 5. This table, indexed by physical core page number, is referenced only by the software and is parallel to CST0. It is a table of short IORBs. (See the IORB table description for format of the IRBSTS entry.)

Defined in: STG

Referenced by: PHYSIO

Format



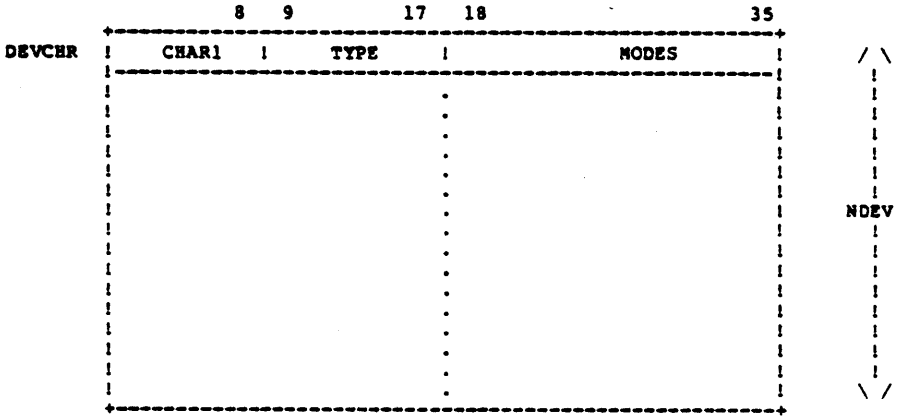
Name: DEVCHR

Description: Device Characteristics Table. This table contains indexed information about each device unit and is initialized from the INIDVT table at system startup time. See INIDVT table for bit definitions.

Defined in: STG

Referenced by: DEVICE, DISC, GTJFN, JSYSA, JSYSF, MEXEC

Format



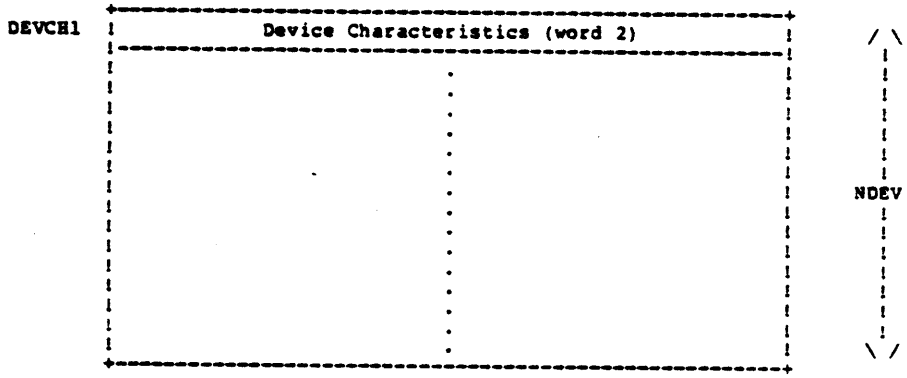
Name: DEVCH1

Description: Device Characteristics Table 1. This table, contains another word of information about each device unit in the system and is initialized from the INIDVT table at system start up time.

Defined in: STG

Referenced by: DEVICE, DISC, GTJFN, JSYSA, JSYSF

Format



Symbol	Bit	Content
D1%SPL	0	Device is spooled
D1%ALC	1	Device is under control of allocator
D1%VVL	2	Volume valid
D1%NIU	3	Device slot not is use
D1%INI	4	Device is being initialized (currently for structures only)

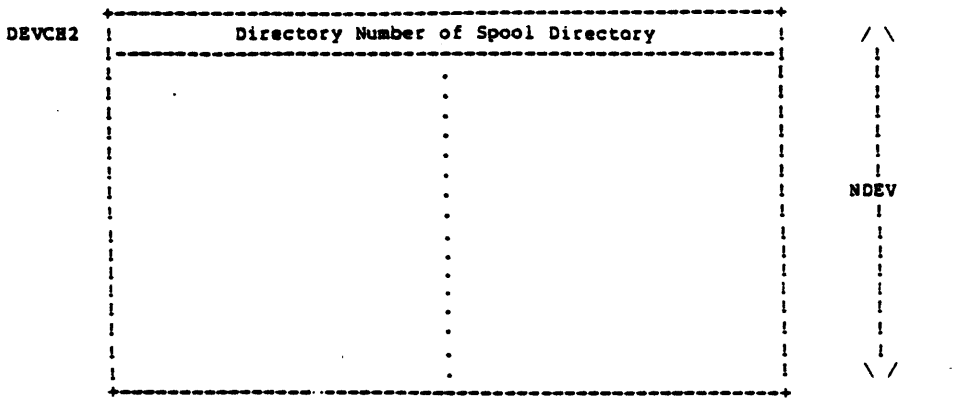
Name DEVCH2

Description: Device Characteristcs Table 2. This table contains spool directory information and is parallel to the DEVCH1 table.

Defined in: STG

Referenced by: DISC, JSYSA

Format



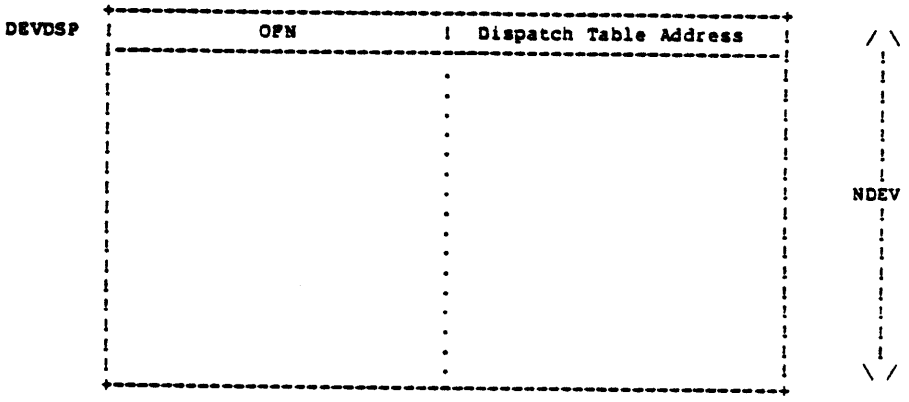
Name: DEVDSP

Description: Device Dispatch Table. This table contains the open file number (if any) and device dispatch table for each device unit and is initialized from the INIDVT table at system startup time.

Defined in: STG

Referenced by: DEVICE, JSYSF

Format



Name: DEV`DTB

Description: Dispatch Table. Each device has its own dispatch table that conforms to the format described below. An error return dispatch address is placed in those words which have no corresponding device function. The naming convention for these tables is the device name concatenated with DTB (i.e. MTADTB, DSKDTB, TTYDTB, etc.)

Defined in: PROLOG

Referenced by:

Format

DLUKD	Directory Setup
NLUKD	Name Lookup
ELUKD	Extension Lookup
VLUKD	Version Lookup
PLUKD	Protection Insertion
ALUKD	Account Insertion
SLUKD	Status Modification
OPEND	Open File
BIND	Sequential Byte Input
BOUTD	Sequential Byte Output
CLOSD	Close File
REND	Rename File
DELD	Delete File
DMPID	Dump Mode Input
DMPOD	Dump Mode Output
MNTD	Mount
DSMD	Dismount
INDD	Initialize a Directory
MTPD	MTAPE Operations
GDSTD	Get Device Status
SDSTD	Set Device Status
RECOUT	Force Record Out (SOUTR)
RFTADD	Read File Time and Date
SFTADD	Set File Time and Date

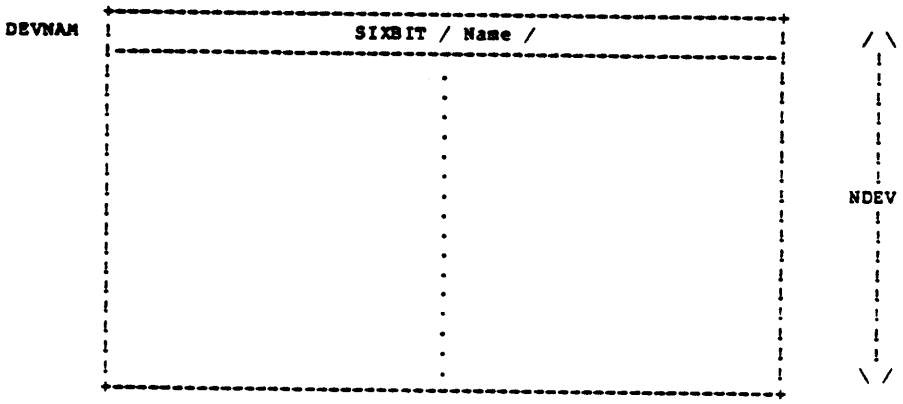
Name: DEVNAM

Description: Device Name Table. This table contains the SIXBIT device name for each device unit. The generic device name is obtained from INIDVT, modified to include unit number (if device has units) and stored in this table at system start up time.

Defined in: STG

Referenced by: DEVICE, JSYSA, MEXEC

Format



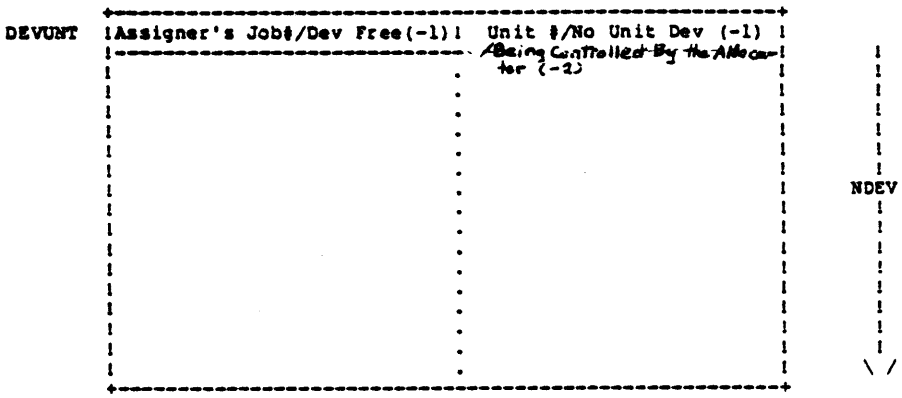
Name: DEVUNT

Description: Device Unit Table. This table, contains information about the job associated with a unit, where the unit information in DEVUNT is built at system startup utilizing data from the INIDVT table.

Defined in: STG

Referenced by: TTYSRV, DEVICE, JSYSA, JSYSF, MEXEC

Format



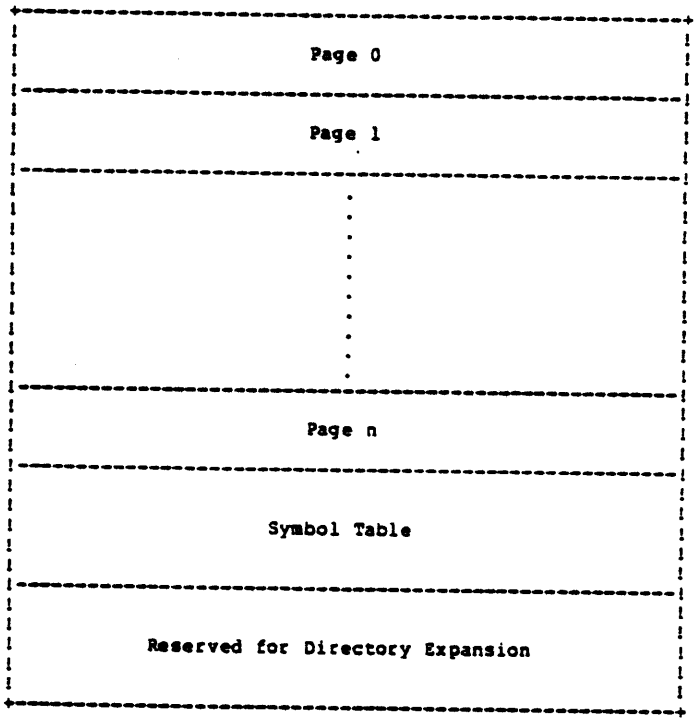
Name: DIRECTORY

Description: Directory Format. The following illustrations show the format of a TOPS-20 directory.

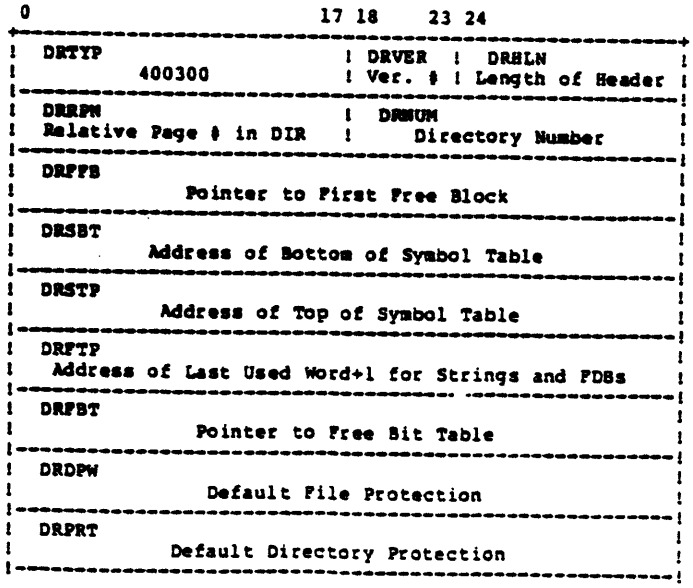
Defined: PROLOG

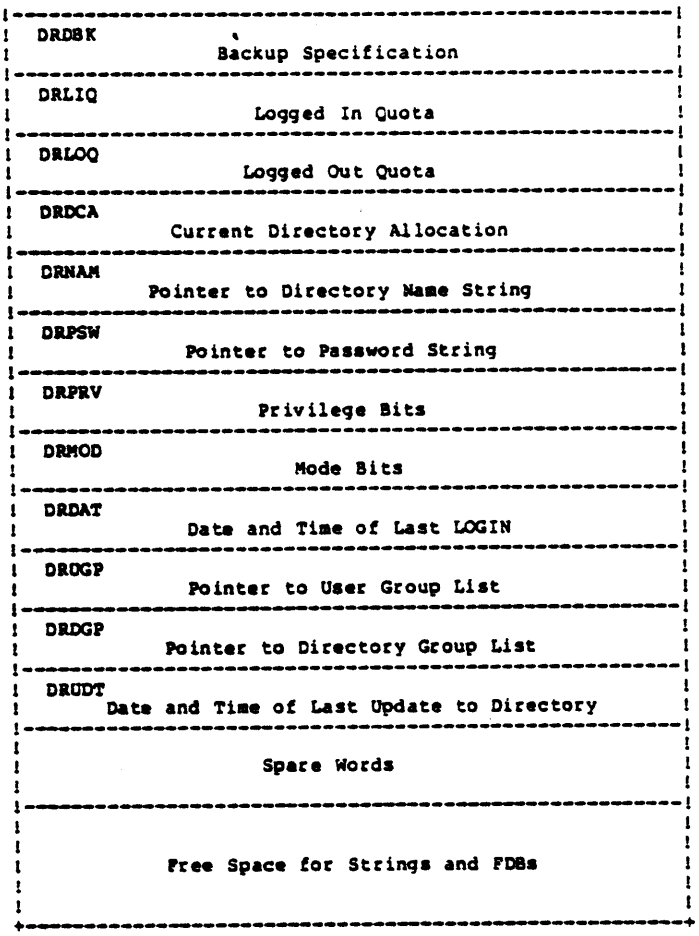
Referenced by: DIRECT, DISC, DSKALL

Overview of a Directory

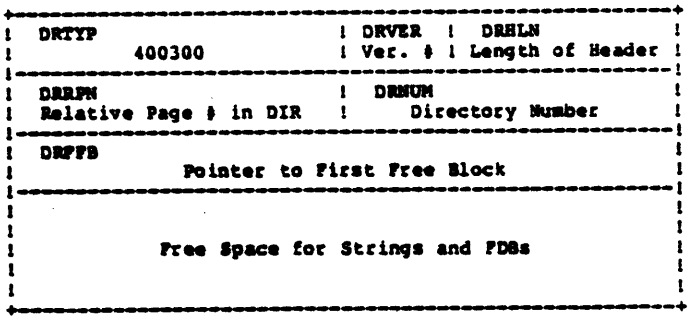


First Page of a Directory

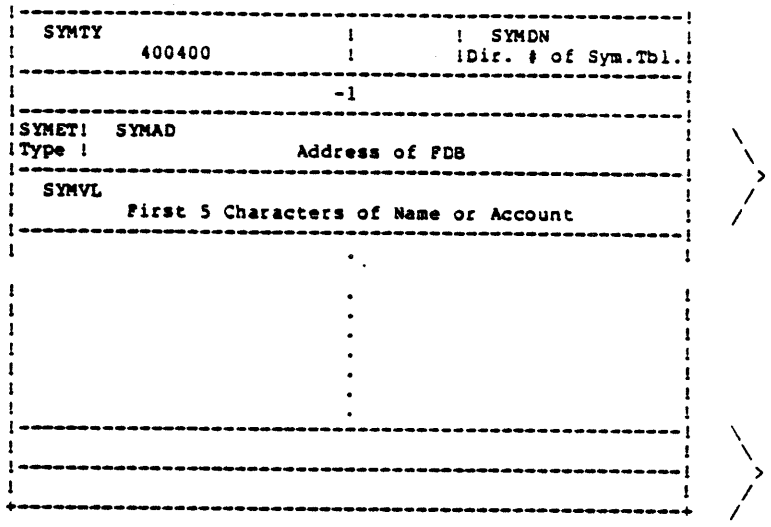


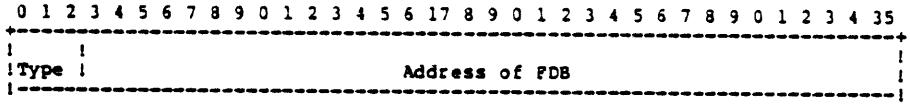


Subsequent Directory Pages



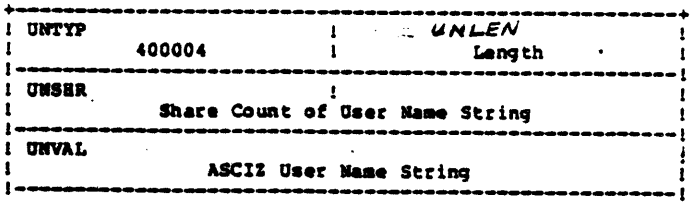
Symbol Table



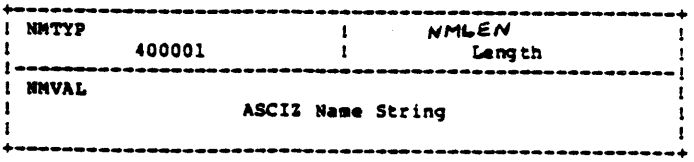


Bits	Pointer	Content	
0-2	SYMET	Entry Type	
		0 = .ETNAM	Name
		2 = .ETUNS	User Name
		4 = .ETACT	Account
3-35	SYMAD	Address of FDB	

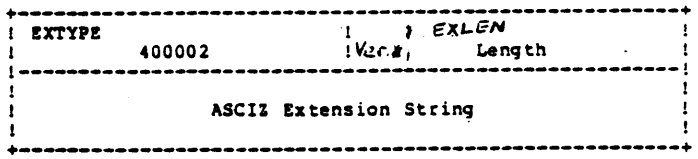
User Name String



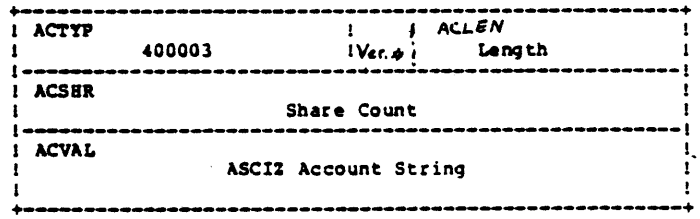
Name String



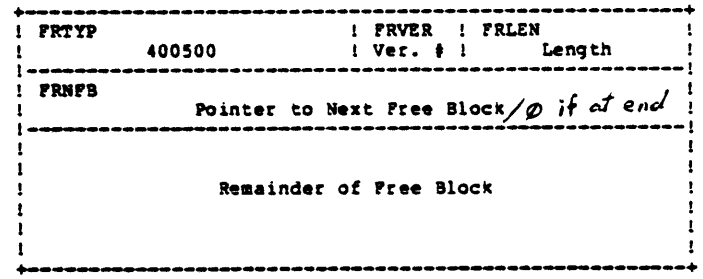
Extension String



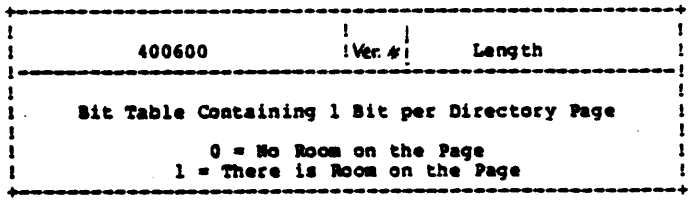
Account String



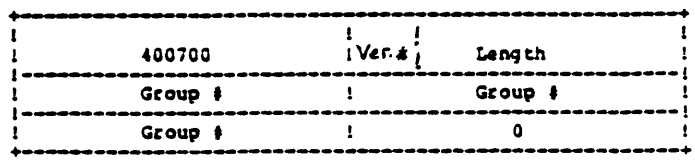
Free Space



Free Bit Table



Group List



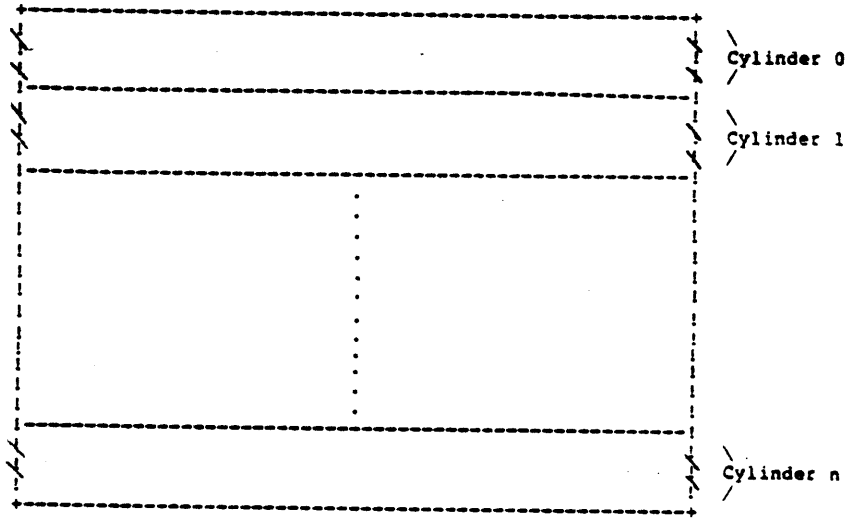
Name: DRMBBT

Description: Drum Bit Table. This bit table indicates which pages are in use and which pages are available in the swapping area.

Defined in: STG

Referenced by: SWPALC

Format



Note: The bit map for each cylinder starts on a word boundary and contains as many full words as are needed for all of its pages.

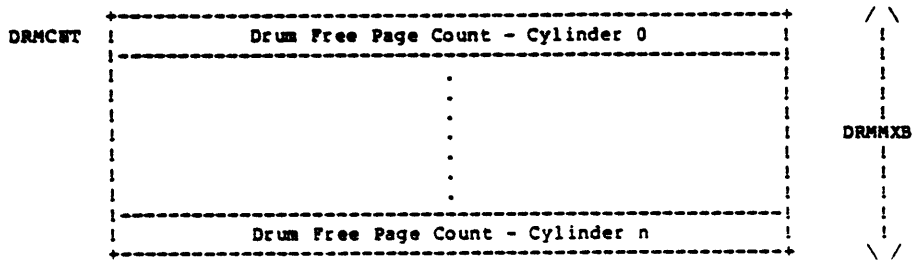
Name: DRMCNT

Description: Drum Count Table. This table, indexed by cylinder records the free page count for the drum (logical swapping area).

Defined in: STG

Referenced by: SWPALC

Format



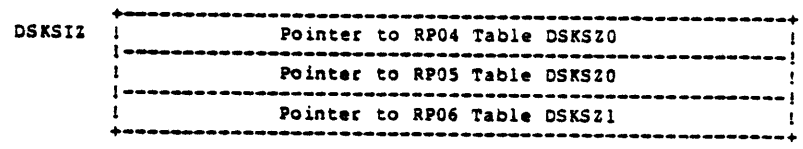
Name: DSKSIZ

Description: Disk Size Pointer Table. This table contains pointers to the disk size data tables. DSKSIZ is parallel to DSKUTP which contains codes for the known disk types. When an entry is added to DSKUPT, a corresponding entry must be added to DSKSIZ to point to the correct size data for that type of disk.

Defined in: STG

Referenced by: DSKALC

Format



Name: DSKSZ`n

Description: Disk Size Table (for type n). The resident table contains size data (for disks) based on type.

n = 0 for RP04 and RP05
n = 1 for RP06

Defined in: STG

Referenced by: DSkALC

Format

DSKSZ`n/SEGPG`n	Sectors per Page
SECCY`n	Sectors per Cylinder
PAGCY`n	Pages per Cylinder
CYLUN`n	Cylinders per Unit
SECUN`n	Sectors per Unit
BTWCY`n	Bit Words in Bit Table per Cylinder
MINFP`n	Minimum Free Pages for Free Choice Allocation

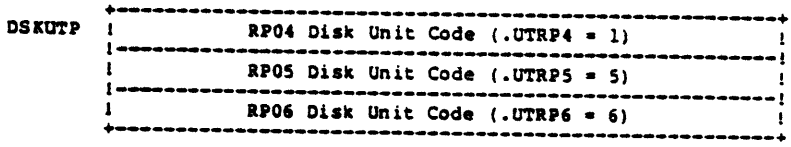
Name: DSKUTP

Description: Disk Unit Type. This table contains the unit types used by the file system.

Defined in: PHYSIO

Referenced by: DSKALC

Format



Name: DST

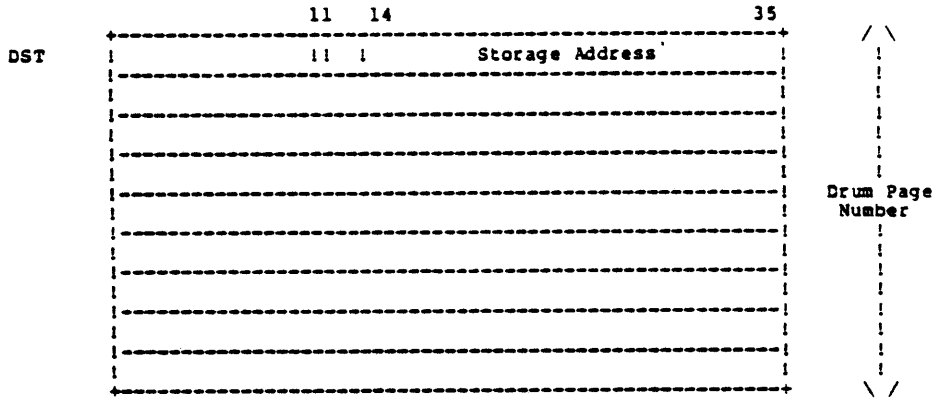
Description: Drum Status Table. This table is indexed as a function of the drum (swapping space) address. The routine GDSTX converts a drum address into a DST index. The DST holds the address of the next lower level of storage (usually disk) for the page stored at that address on the drum.

BWRBIT (bit 11) indicates if the page has been changed since being read from the lower level storage. The page will only be copied back on to the lower level storage if BWRBIT is set (i.e., page modified) when the page is no longer in use. A slot no longer in use contains a -1.

Defined in: STG

Referenced by: PAGEM

Format



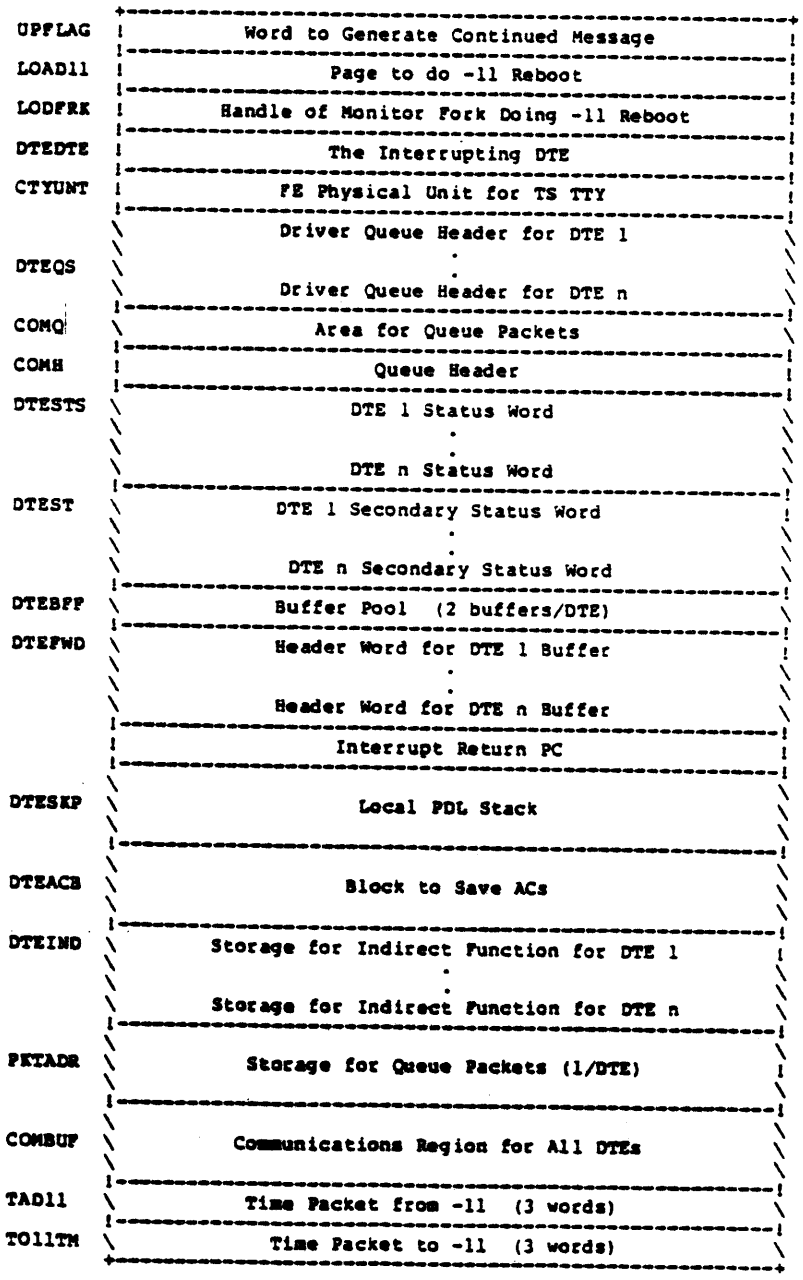
Name: DTE

Description: Storage area for DTEs

Defined in: STG

Referenced by: DTESRV, APRSRV, MEXEC, SCHED

Format



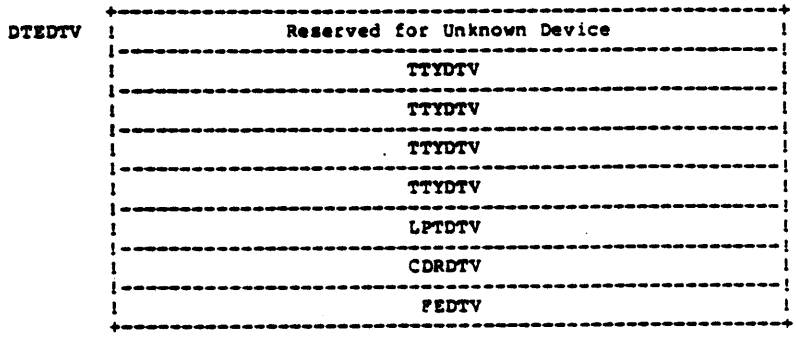
Name: DTEDTV

Description: Protocol Device Dispatch Table.

Defined in: STG

Referenced by: DTESRV

Format



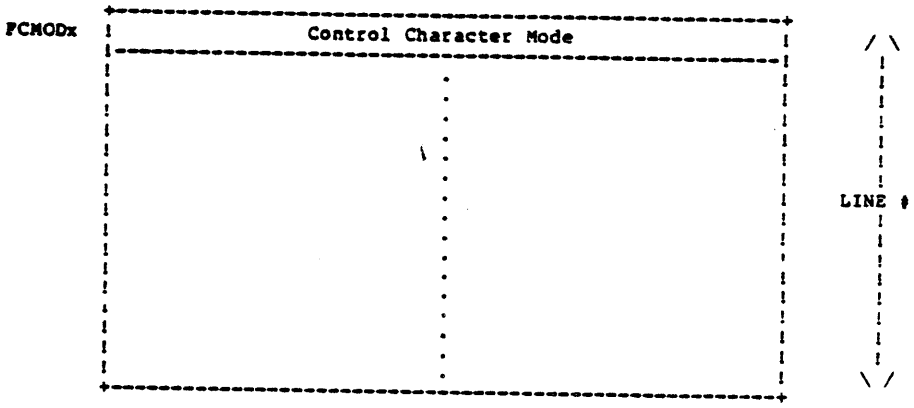
Name: FCMODx

Description: File Control Character Mode Tables. There are two tables (FCMOD1 and FCMOD2), each indexed by line number, which contain control character mode information.

Defined in: STG

Referenced by: TTYSRV

Format

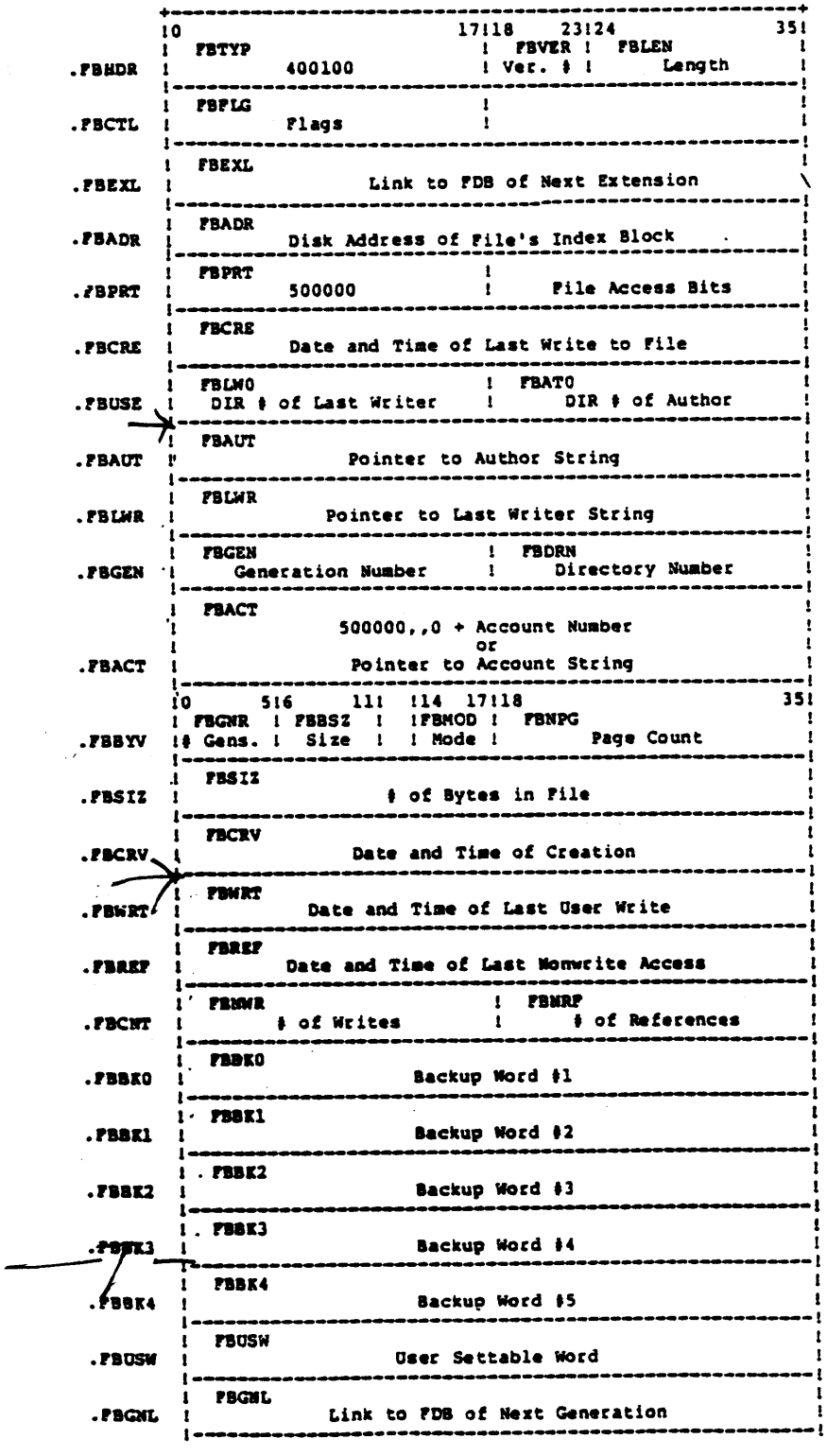


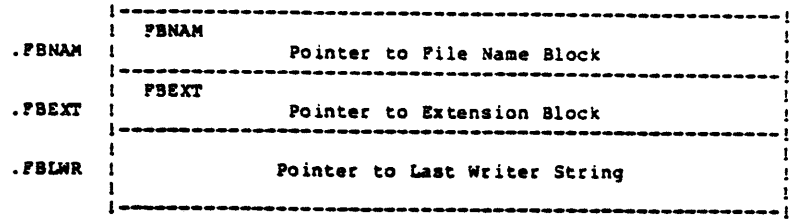
Name: FDB

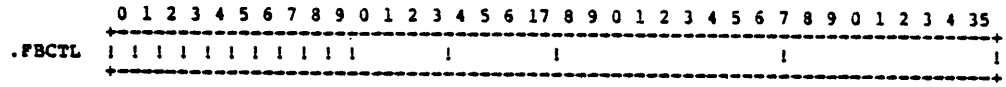
Description: File Description Block. All attributes of a file are stored in its description block (FDB) maintained in the file's directory. An FDB is built in the directory's free space area when a file is created.

Defined in: PROLOG, MONSYM

Referenced by: DISC, DIRECT, DSKALC, GTJEN, JSYSA, JSYSF, FILINI, LINEPR, IO, SYSERR, DTESRV







Symbol	Bits	Pointer	Content
FB%TMP	0		File is temporary
FB%PRM	1		File is permanent
FB%NEX	2		No extension for this file yet; file doesn't really exist.
FB%DEL	3		File is deleted
FB%NXF	4		File doesn't exist (first write not complete)
FB%LNG	5		Long file
	6		Reserved for DEC
FB%DIR	7		File is a directory
FB%NOD	8		File is not saved by backup system
FB%BAT	9		File may have bad pages
FB%FCF	14-17		File class field 0 = .FBNRM Not an RMS file 1 = .FBRMS RMS file

Note: See Monitor Call's Reference Manual (Chapter 2) for more information.

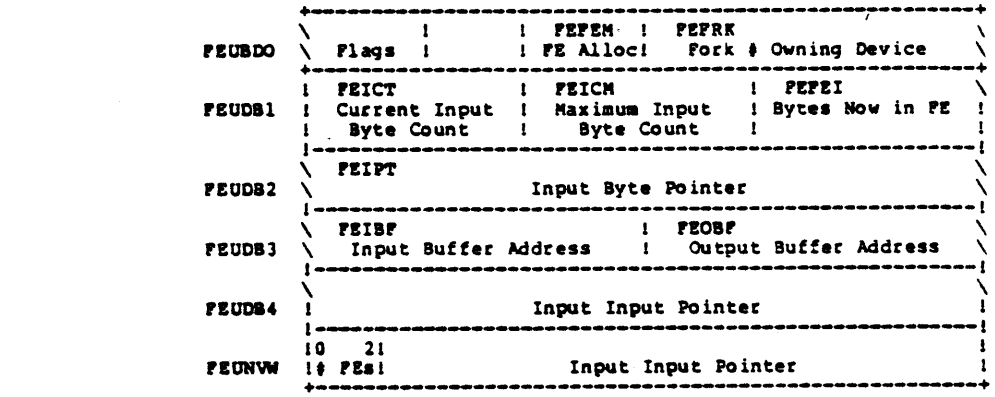
Name: FE

Description: Storage area for front end devices. Each entry is FEN words long (except FEUNVW), where FEN equals the number of front end devices.

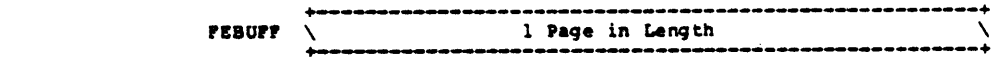
Defined in: STG

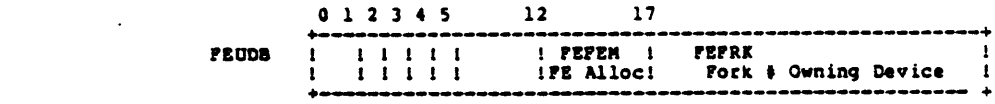
Referenced by: FESRV

Format



The buffer area for the front end is in the monitor's nonresident address space.





Bit(s)	Pointer	Content
0-1	FEDTE	DTE owning this device
2	FEBLK	Unit is blocked
3	FEPST	Waiting for DTE Post
4	FEEOF	Input EOF declared by FE
5	FEVDT	FE assignment is valid
12-17	FEFEM	FE Allocation
18-35	FEFRK	Fork # owning device

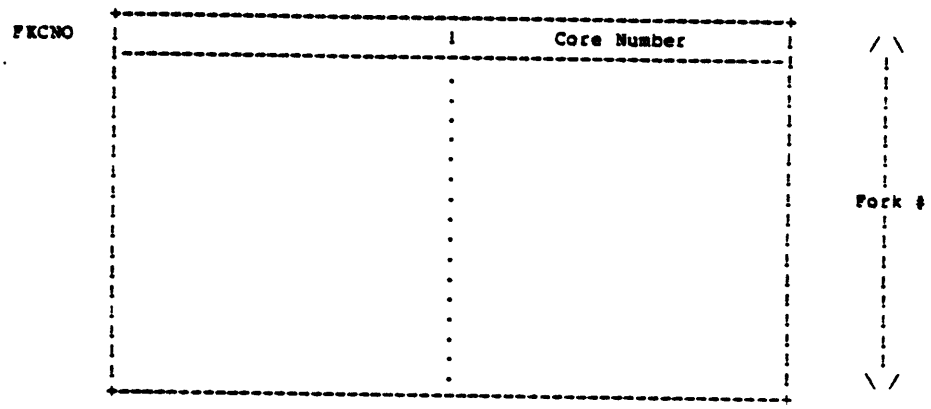
Name: FKCNO

Description: Fork Core Number Table. This table, indexed by fork #, contains the core number (given when a process enters the balance set) for each fork. The core number is used to set a corresponding bit in the pager's process use register (PUR) when the fork is chosen to run.

Defined in: STG

Referenced by: PAGEM, SCHED

Format



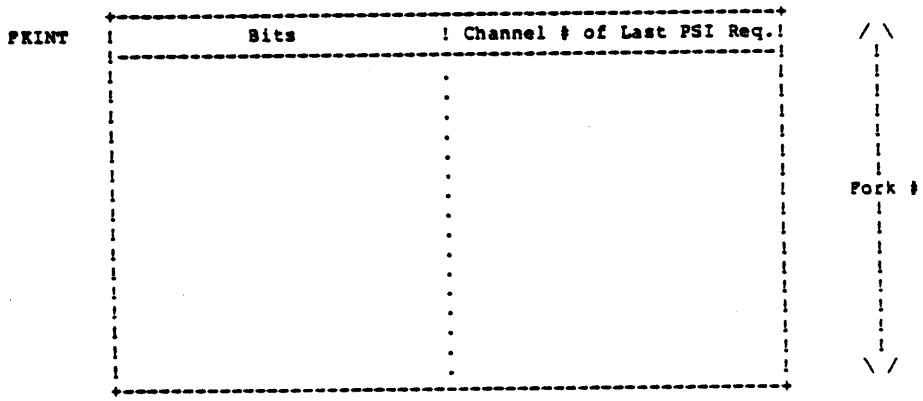
Name: FKINT

Description: Fork Interrupt Table. This table, indexed by fork #, contains the pseudo-interrupt communication register for each fork. The left half of each entry contains bits recording the type of request. The symbols for these requests have right half bit assignments (i.e. bits 20-31) but are tested against the left half of the table.

Defined in: STG

Referenced by: TTYSRV, FORK, MEXEC, SCHED

Format



Symbol	Bit	Content
	0	Interrupt Request(s) pending
	1	Fork not interruptable
NEWFKF(B20)	2	Initiate new fork
NEWJBF(1B21)	3	Initiate new job
PSIIF(1B22)	4	Channel interrupt requested in FKINTB
PSIT1F(1B23)	5	Terminal code Interrupt, Phase 1
PSIT2F(1B24)	6	Terminal code Interrupt, Phase 2
SUSFKR(1B25	7	Suspend fork request
PSIWTF(1B26)	8	Job was in wait state
PSILOB(1B27)	9	Logout job request
FRZB1(1B28)	10	Direct freeze has been done
FRZB2(1B29)	11	Indirect freeze has been done
PSICOB(1B30)	12	Carrier off action request
PSITLE(1B31)	13	Time Limit Exceeded interrupt

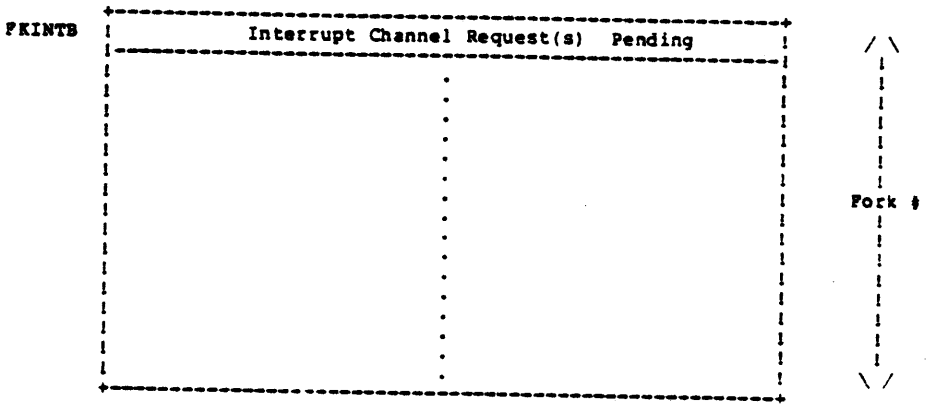
Name: FKINTB

Description: Fork Interrupt Buffer Table. This table, indexed by fork #, contains the buffer for the pseudo-interrupt channel requests pending for each fork since the fork's last PSI interrupt.

Defined in: STG

Referenced by: SCHED

Format



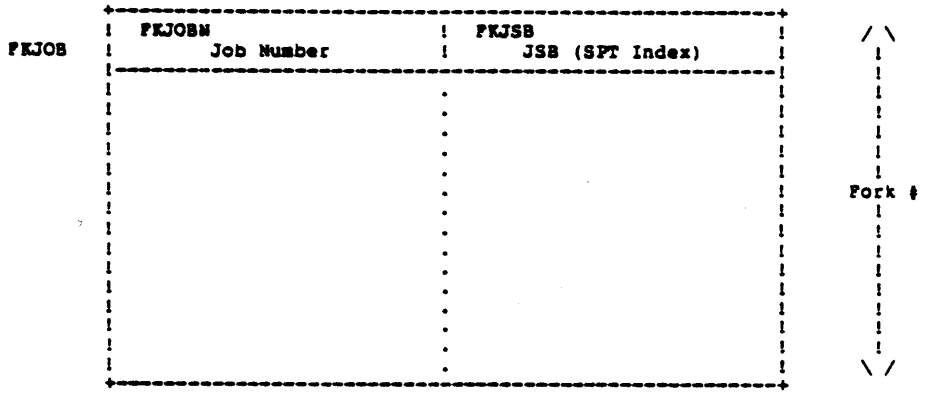
Name: FKJOB

Description: Fork Job Table. This table, indexed by fork #, holds each fork's job number and JSB address (SPT index).

Defined in: STG

Referenced by: APRSRV, DTESRV, ENQ, FILMSC, FORK, IPCF, MEXEC, PAGEM, PHYSIO, SCHED

Format



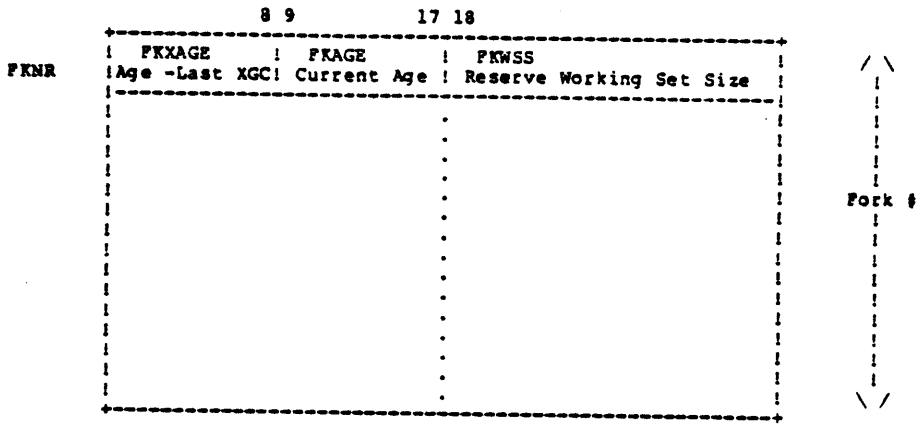
Name: FKNR

Description: Fork Number of Reserve Pages Table. This table, indexed by fork #, contains in the right half the current reserve working set size for each fork. Also, it contains in the left half the current age stamp (to be loaded into the pager's age resister when the fork is chosen to run) and the age stamp value at the last time local garbage collection (removal of less frequently used pages) took place for the fork.

Defined in: STG

Referenced by: PAGEM, SCHED

Format



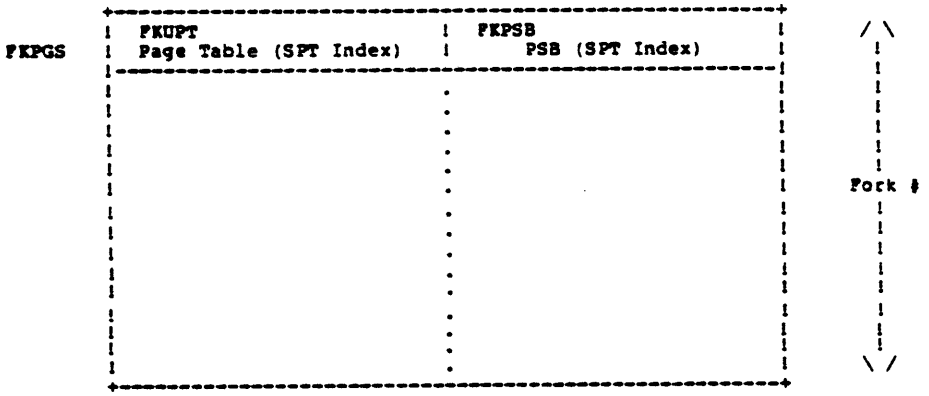
Name: FKPGS

Description: Fork Page and Process Storage Table. This table, indexed by fork #, contains the page table and PSB locations (SPT indexes) for each fork.

Defined in: STG

Referenced by: FORK, IPCF, JSYSA, PAGEM, SCHED

Format



Name: FKPGST

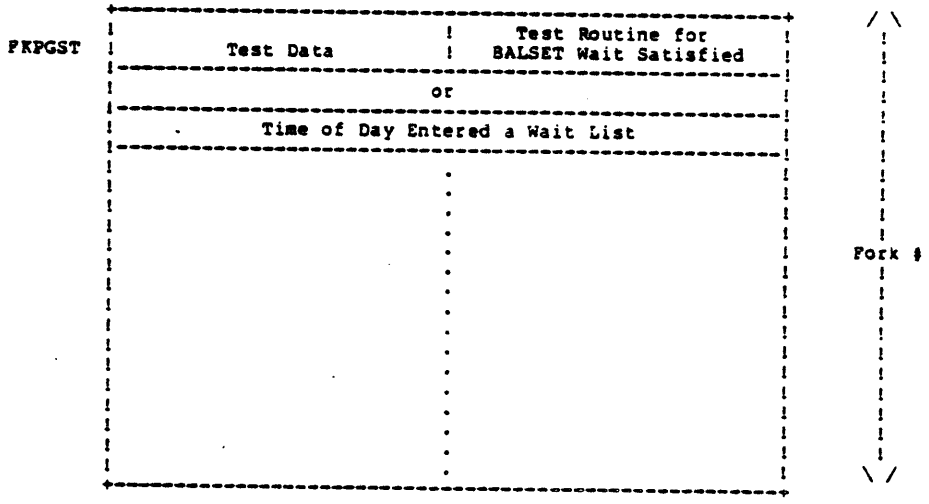
Description: This table, indexed by fork #, holds test routine information for forks in a balance set wait state. The test routine checks if wait satisfied has occurred.

For forks on a wait list (and therefore not in the balance set), this table contains the time of day the fork entered the list.

Defined in: STG

Referenced by: SCHED

Format



Name: FKPT

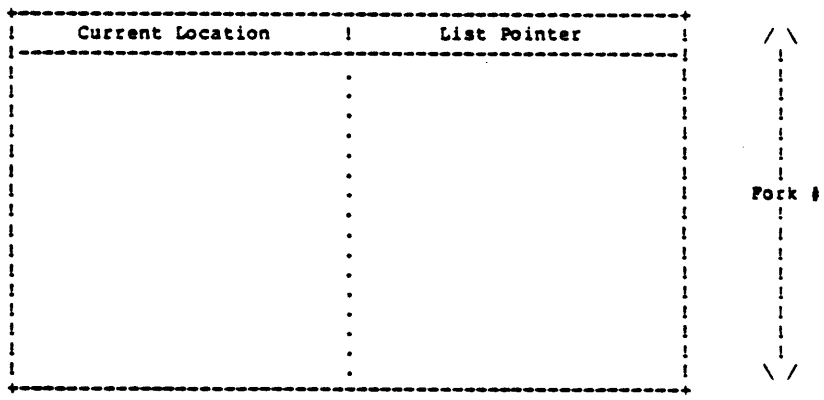
Description: Fork List Pointer Table. This table, indexed by fork #, gives the chain of forks for each list of forks in the system. That is, it holds the linked list of forks on TTILST, CLKLST, GOLST, etc. A fork is either on one of the wait-lists or the go-list. The right half contains the list pointer to the next fork on the same list and the left half contains WTLST or GOLST. If the left half contains WTLST, the type of wait-list can be obtained from the right half of FKQ2.

Defined in: STG

Referenced by: APRSRV, FORK, MEXEC, PAGEM, SCHED

Fork# = index

Format



GOLST
WTLST - some wait list, not nec. WTLST

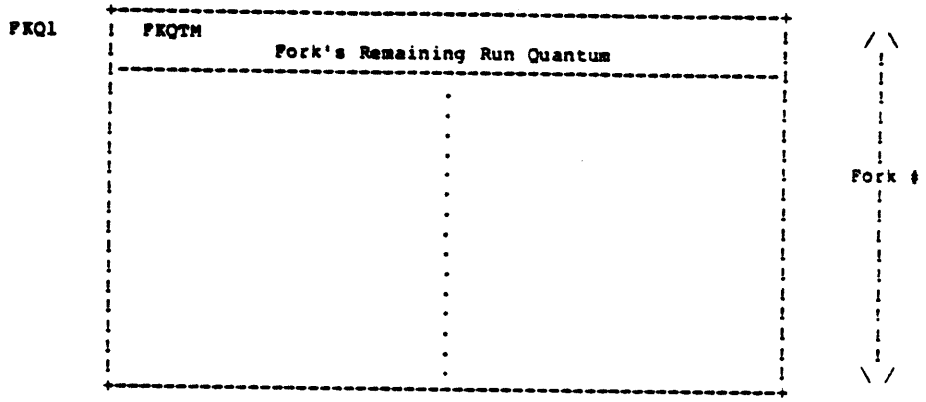
Name: FKQ1

Description: Fork Run Queue Table 1. This table, indexed by fork #, contains each fork's remaining run quantum. When this quantum expires, the process will be requeued to a lower run position and given the the quantum associated with that run queue.

Defined in: STG

Referenced by: SCHED

Format



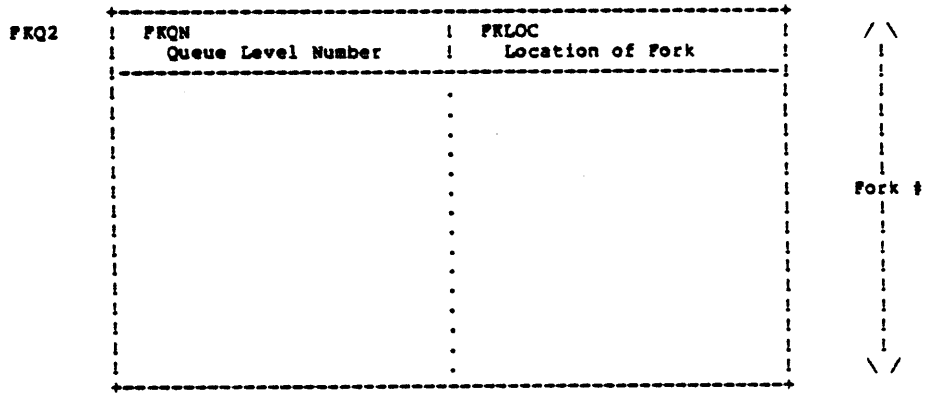
Name: FKQ2

Description: Fork Run Queue Table 2. This table, indexed by fork #, contains the queue level number and fork location [i.e., list address (TTILST, (GOLST, etc.) or BALSET table index] for each fork.

Defined in: STG

Referenced by: PAGEM, SCHED

Format



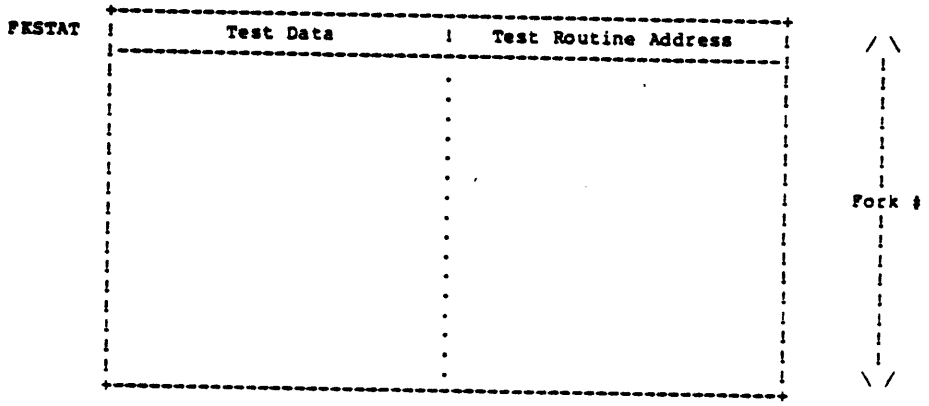
Name: FKSTAT

Description: Fork Status Table. This table, indexed by fork #, gives the address of each blocked fork's test routine which when called determines if wait satisfied has occurred for the fork.

Defined in: STG

Referenced by: DIRECT, FORK, SCHED

Format



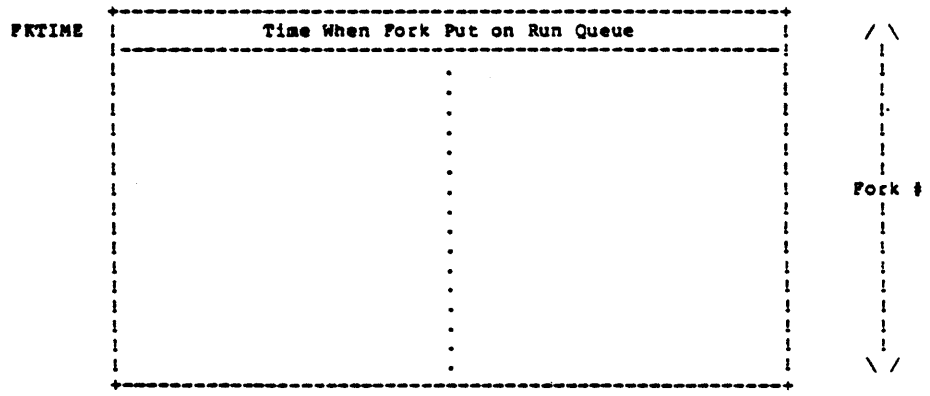
Name: PKTIME

Description: Fork Time Table. This table, indexed by fork #, gives the time of day (TODCLK) at which each fork was put on its current run queue.

Defined in: STG

Referenced by: SCHED

Format



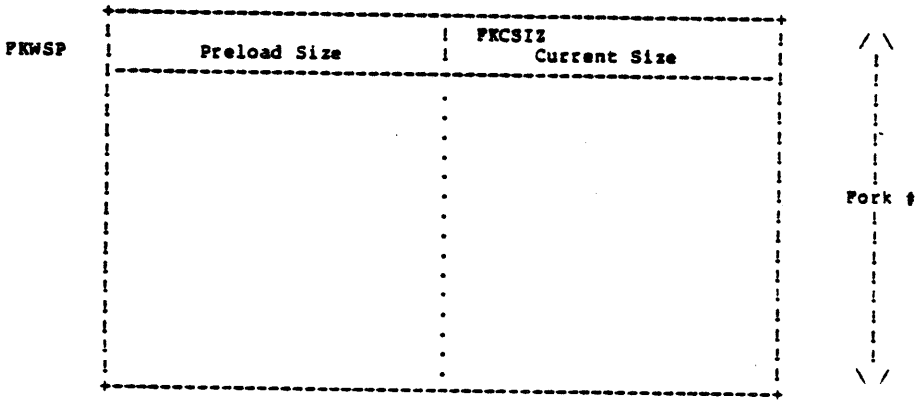
Name: FKWSP

Description: Fork Working Set (physical in-core size) Table. This table, indexed by fork #, contains in the right half the number of physical pages currently assigned to each fork. The left half is used to hold the pre-load size as determined by LDJOB when a fork enters the balance set.

Defined In; STG

Referenced by: PAGEM, SCHED

Format



Name: HOM

Description: Home Block. Block on each disk unit which contains vital statistics that cannot be built in when a monitor is generated. These are primarily parameters of the unit and the STR to which it belongs.

Defined in: DSKALC

Referenced in: DSKALC, PHYSIO, JSYSA

HOMNAM	SIXBIT/HOM/	
HOMID	SIXBIT/Unit ID/	
HOMPHY	Physical Disk Address of This Home Block	Physical Disk Address of Other Home Block
HOMSNM	SIXBIT/Structure Name/	
HOMLUN	# of Packs in STR	Logical Unit # Within STR
HOMEOM	Block # of This Home Block	Block # of Other Home Block
HOMP4S	# of Pages for Swapping on This Structure	
HOMFST	First Swapping Track on Unit	
HOMRXB	Address of Index Block of ROOT-DIRECTORY	
HOMBXB	Address of Index Block of BACKUP-COPY-OF-ROOT-DIRECTORY	
HOMFLG	Flags	
HOMSIZ	Number of Sectors in This Unit	
HOMBTB	Number of Tracks in Structure	
HOMMID	Pack Unique Code	
	Reserved for Expansion	
HOMFE0	Front End File System (sector #)	
HOMFE1	Front End File System (# of sectors)	
	Reserved for Expansion	
HOMUID	12 Character Unit I.D. (PDP-11 Format)	
HOMOID	12 Character Owner I.D. (PDP-11 Format)	
HOMFSN	12 Character File System Name (PDP-11 Format)	
HOMCOD	0	707070
HOMSLF	0	This Block #

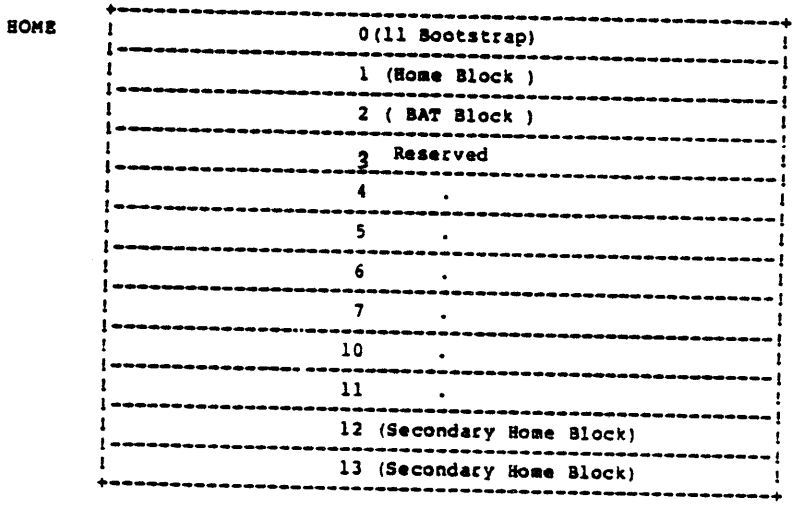
Name: HOME

Description: Home Table. This table contains the disk pages for the HOME and BAT blocks and the 11 Bootstrap program.

Defined in: STG

Referenced by: DSKALC

Format



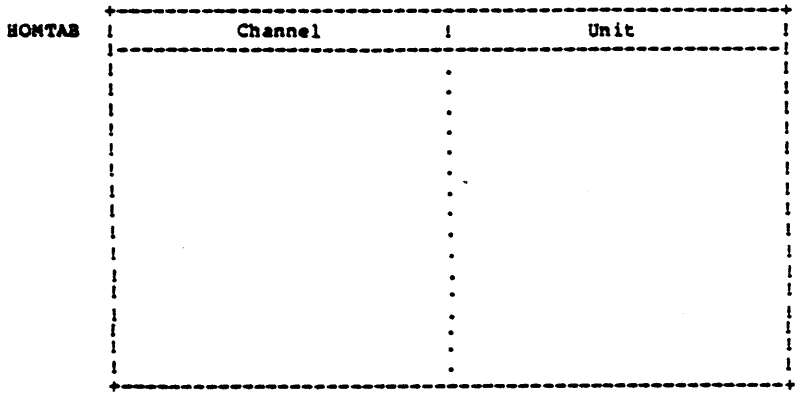
Name: HOMTAB

Description: This table contains the logical to physical mapping (channel and unit) per logical unit, and its length equals the maximum number of packs in a structure.

Defined in: STG

Referenced by: DSKACL

Format



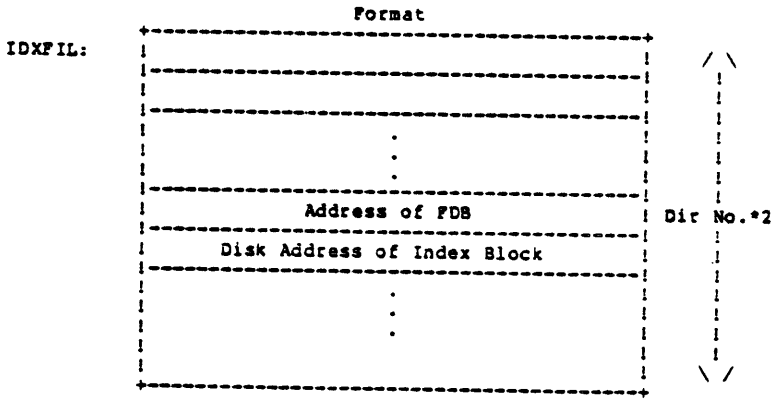
Name: IDXFIL

Defined in: PROLOG

Referenced in: FILINI, DIRECT

Description: The Index table of the structure currently mapped for a process. Each structure has an index table file. The file is indexed by 2* directory number as each entry is two words long. For each directory on the structure, an entry will contain the address of the FDB for the directory and the disk address of the index block for the directory.

The table, IDXFIL is mapped from the file STRNAM:<ROOT-DIRECTORY>INDEX-TABLE.BIN into the PSB area. When a structure is mounted (physically) the system gets on OFN for this file and stores it in entry, STRIDX, in the SDB table for that structure. The table entries are created at this time (mount-time).



Name: INDEX

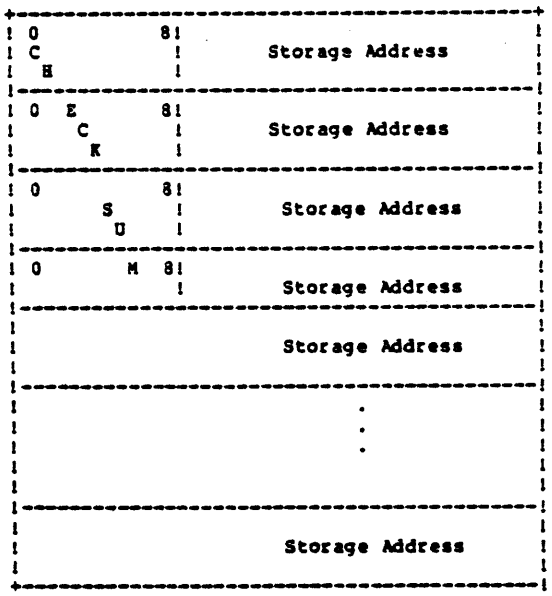
Description: The Index Block (1 page) exists for each disk file and contains pointers to where each of the file's pages resides on disk. If more than one index block is needed for non-directory files, a super index block (1 page) is created which points to the home disk address of each index block. (Note that the maximum file size is 512*512 pages.)

When the file is referenced, an in-core copy of the index block is maintained which keeps track of the file's active pages in the system. (i.e. Whether the pages are in-core, on the swapping area, or on disk.)

Defined in:

Referenced by: PAGEM, PHYSIO

Format



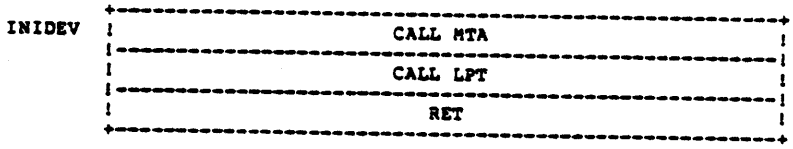
Name: INIDEV

Description: Initialize Devices. This table contains calls to initialize devices after loading the swappable monitor.

Defined in: STG

Referenced by: PILINI

Format



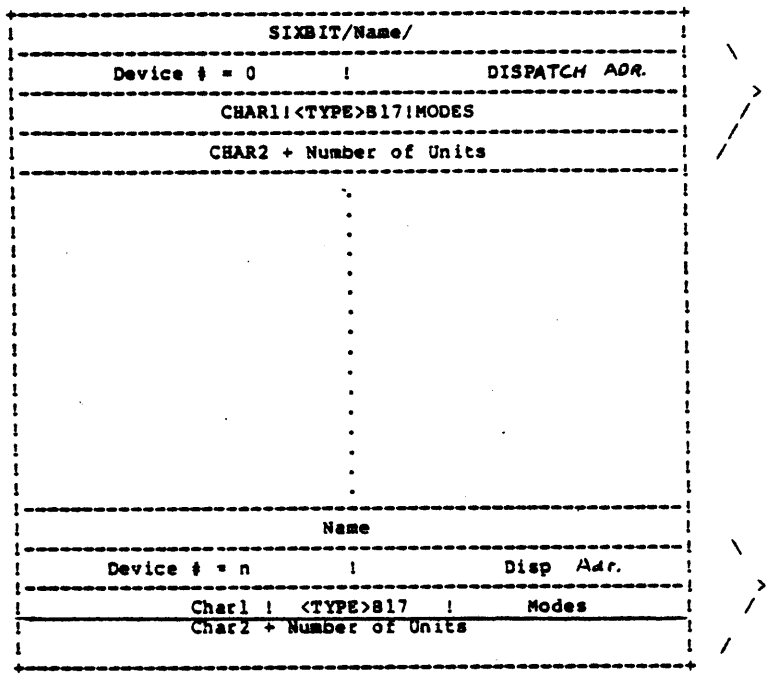
Name: INIDVT

Description: Device Initialization Table. This static table generated at assembly time, contains a 4 word block for each type of device on the system. It is used at system startup time to initialize the device tables, DEVCHR, DEVCH1, DEVNAM, & DEVUNT.

Defined in: STG, MONSYM

Referenced by: DEVICE

Format



CHAR1 can be a combination of the following:

Symbol	Bit	Meaning
DV%OUT	0	Can do output
DV%IN	1	Can do input
DV%DIR	2	Has a directory
DV%AS	3	Is assignable
DV%MDD	4	Is a multiple directory device
DV%AV *	5	Is available to this job
DV%ASN *	6	Is assigned by ASND
DV%MDV	7	Is a mountable device
DV%MNT *	8	Is mounted

TYPE is one of the following:

Symbol	Value	Meaning
.DVDSK	0	Disk
.DVMTA	2	Magtape
.DVPTP	5	Spooled PTP
.DVLPT	7	Spooled & physical line printer
.DVCDR	10	Spooled & physical card reader
.DVFE	11	Front End Device
.DVTTY	12	Terminal
.DVPTY	13	Pseudo TTY
.DVNUL	15	Null Device
.DVPLT	17	Spooled Plotter
.DVCDP	21	Spooled Card Punch

MODES can be a combination of the following:

Symbol	Bit	Meaning
DV%M0	35	Can be opened in mode 0
DV%M1	34	" 1
DV%M2	33	" 2
DV%M3	32	" 3
DV%M4	31	" 4
DV%M5	30	" 5
DV%M6	29	" 6
DV%M7	28	" 7
DV%M10	27	" 10
DV%M11	26	" 11
DV%M12	25	" 12
DV%M13	24	" 13
DV%M14	23	" 14
DV%M15	22	" 15
DV%M16	21	" 16
DV%M17	20	" 17

CHAR2 can be a combination of the following:

Symbol		Bit	Meaning
D1%SPL		0	Is spooled
D1%ALC	*	1	Is under control of allocator
D1%VVL	*	2	Volume valid
D1%NIU	*	3	Device slot not in use
D1%INI	*	4	Device is being initialized (currently for structures only)

* These bits are zero at assembly time and are set by the monitor when appropriate in their corresponding device tables. (i.e. DEVCHR or DEVCH1)

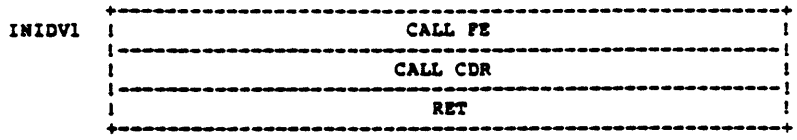
Name: INIDV1

Description: Device initiation for front end devices.

Defined in: STG

Referenced by: DTESRV

Format



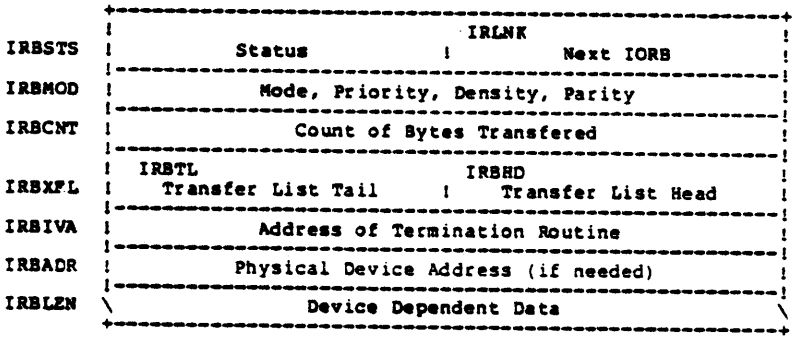
Name: IORB

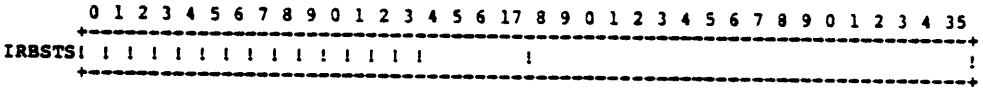
Description: I/O Request Block. Whenever a request for massbus I/O (i.e. DSK or MTA) occurs, an IORB is built for that request. It is of the long form described below for magtape requests and special disk I/O. However, the most common IORB format for disk I/O is a one word IORB, consisting of just the status word, IRBSTS, and stored in the CST5 table.

Defined: PHYPAR

Referenced by: PHYSIO, STG

Format

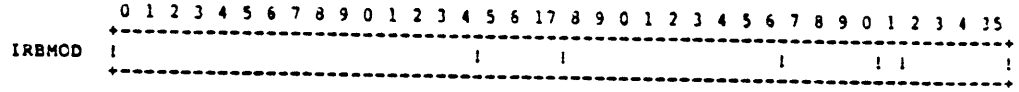




Symbol	Bits	Pointer	Content
IS.SHT	0		Short form (PAGEM) request
IS.DON	1		Done with this job
IS.ERR	2		Error on this operation
IS.NRT	3		No more retries
IS.WGU	4		Wrong unit interrupted
IS.TPM	5		Hit tape mark
IS.EOT	6		On write only, hit physical EOT
IS.WLK	7		Write locked
IS.IER	8		Inhibit error recovery
IS.DER	9		Data error
IS.HER	10		Hardware error on device
IS.BOT	11		Hit BOT
IS.RTL	12		Record too long (buffer too small)
IS.IEL	13		Inhibit error logging
	14-17	IFSCN	Function code
	18-35	IRLNK	When referring to link

Function Codes for ISFCN

Symbol	Code	Function
IRFRED	1	Read data
IRFRDF	2	Read data and format (count, key, header)
IRFWRT	3	Write Data
IRFWTF	4	Write format
IRFSEK	5	Seek
IRFFSB	6	Forward space block
IRFBSB	7	Backspace block
IRFWTM	10	Write tape mark
IRFERG	11	Erase gap
IRFREW	12	Rewind
IRFRUN	13	Rewind and unload
IRFRDR	14	Read reverse
IRFRCR	15	Recovery read

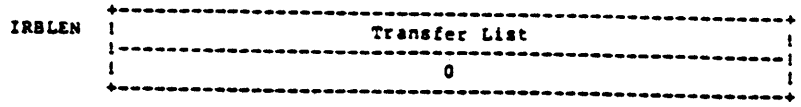


Bits	Pointer	Content
15-17	IRBDM	Data Mode
27-30	IRBPRI	Priority
31	IRBPAR	Parity
32-35	IRBDN	Density

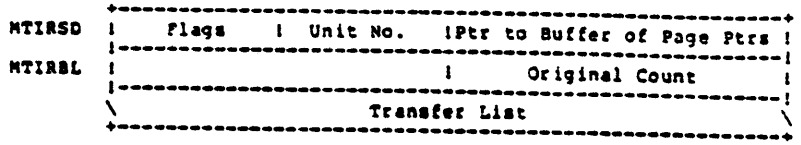
Data Modes for IRBDM

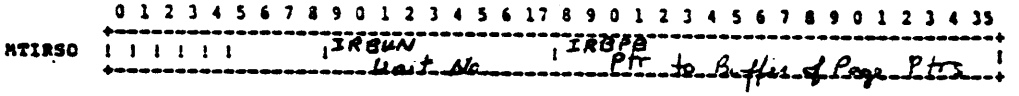
Symbol	Code	Meaning
IRMWRD	1	Word mode
IRM6BT	2	Six bit
IRM7BT	3	Seven bit
IRM8BT	4	Eight bit
IRMMAX	5	Maximum legal mode

If device is disk, IRBLEN becomes:



If device is magtape, IRBLEN becomes:





Bits	Pointer	Content
0	IRBFR	Buffer ready for use
1	IRBFQ	Current buffer flag
2	IRBFA	Active flag, IORB being filled or emptied by service routine
3	IRBAB	ICRB aborted due to an error
4	IRBFF	IORB free
9-17	IRBUN	Unit number
18-35	IRBPB	Pointer to buffer of page pointers

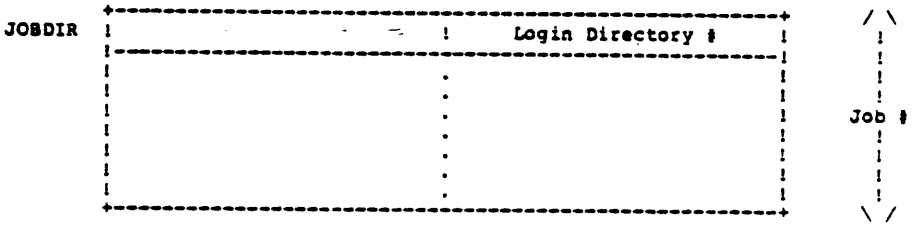
Name: JOBDIR

Description: Job Directory Table. This table, indexed by job #, contains the number of the attached directory and the login directory for each job.

Defined In: STG

Referenced by: APRSRV, TTYSRV, DIRECT, DISC, DTESRV, FILINI, IPCF, JSYSA, MAGTAP, MEXEC

Format



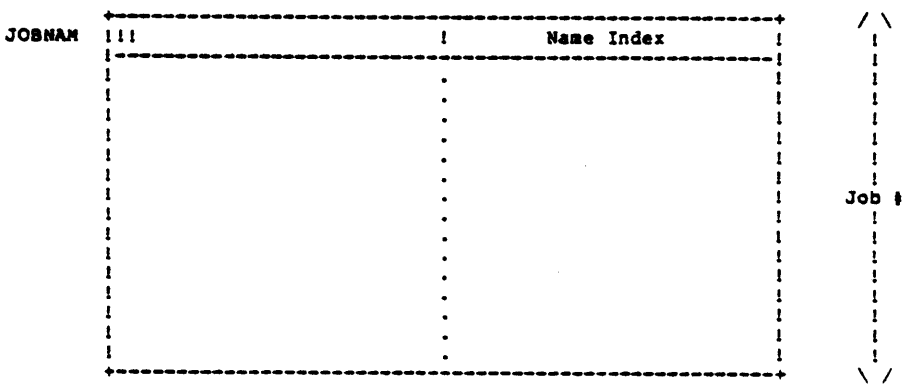
Name: JOBNAM

Description: Job Name Table. This table, indexed by job #, contains an index into the subsystem name tables (SNAMES, STIMES, etc.) indicating what subsystem, if any, each job is running. The name index is for statistics only and is not used by the monitor.

Defined In: STG

Referenced by: FORK, MEXEC, PAGEM, PHYSIO, SCHED

Format



Bit	Pointer	Content
0	DIAFL	Job has DIAG resource
1	HIBFL	Flag used by HIBER JSYS. If set, implies a wakeup signal to THIBR

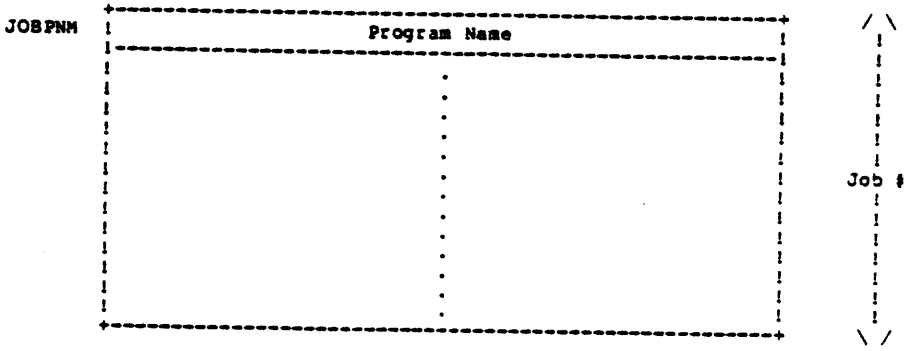
Name: JOBPNM

Description: Job Program Name. This table, indexed by job #, contains each job's program name.

Defined In: STG

Referenced by: DTESRV, IPCF, MAGTAP, MEXEC

Format



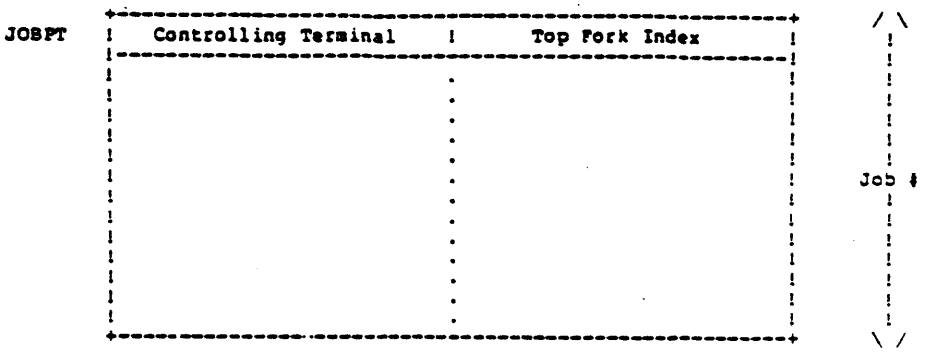
Name: JOBPT

Description: Job Process Table. This table, indexed by job #, contains the number of the controlling terminal, or -1 for a detached job, and the index of the top fork of the job.

Defined in: STG

Referenced by: TTYSRV, DEVICE, FILMSC, FORK, IO, IPCF, JSYSA, MEXEC, SCHED

Format



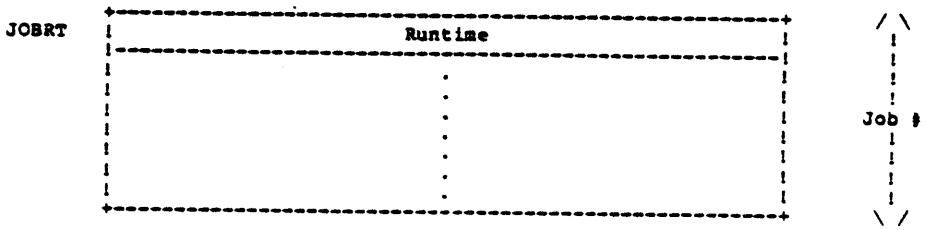
Name: JOBRT

Description: Job Runtime Table. This table, indexed by job #, contains the total runtime of each job (sum of all forks) in milliseconds. If a word contains a -1, the job does not exist.

Defined in: STG

Referenced by: ENQ, FORK, IPCF, JSYSA, MEXEC, SCHED

Format



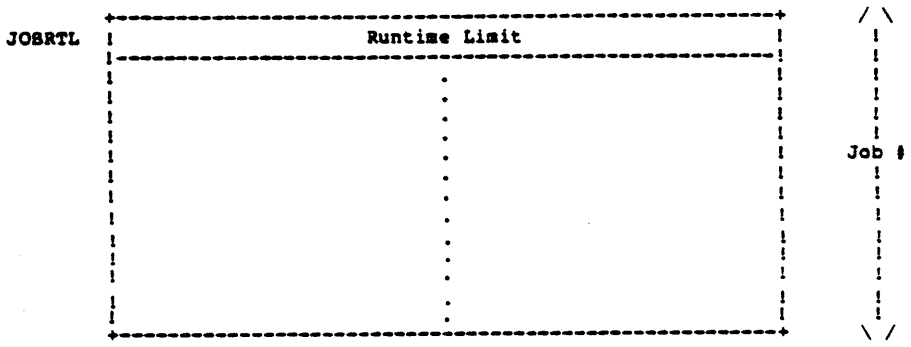
Name: JOBRTL

Description: Job Runtime Limit. This table, indexed by job #, contains each job's runtime limit.

Defined in: STG

Referenced by: FORK, JSYSA, MEXEC, STG

Format



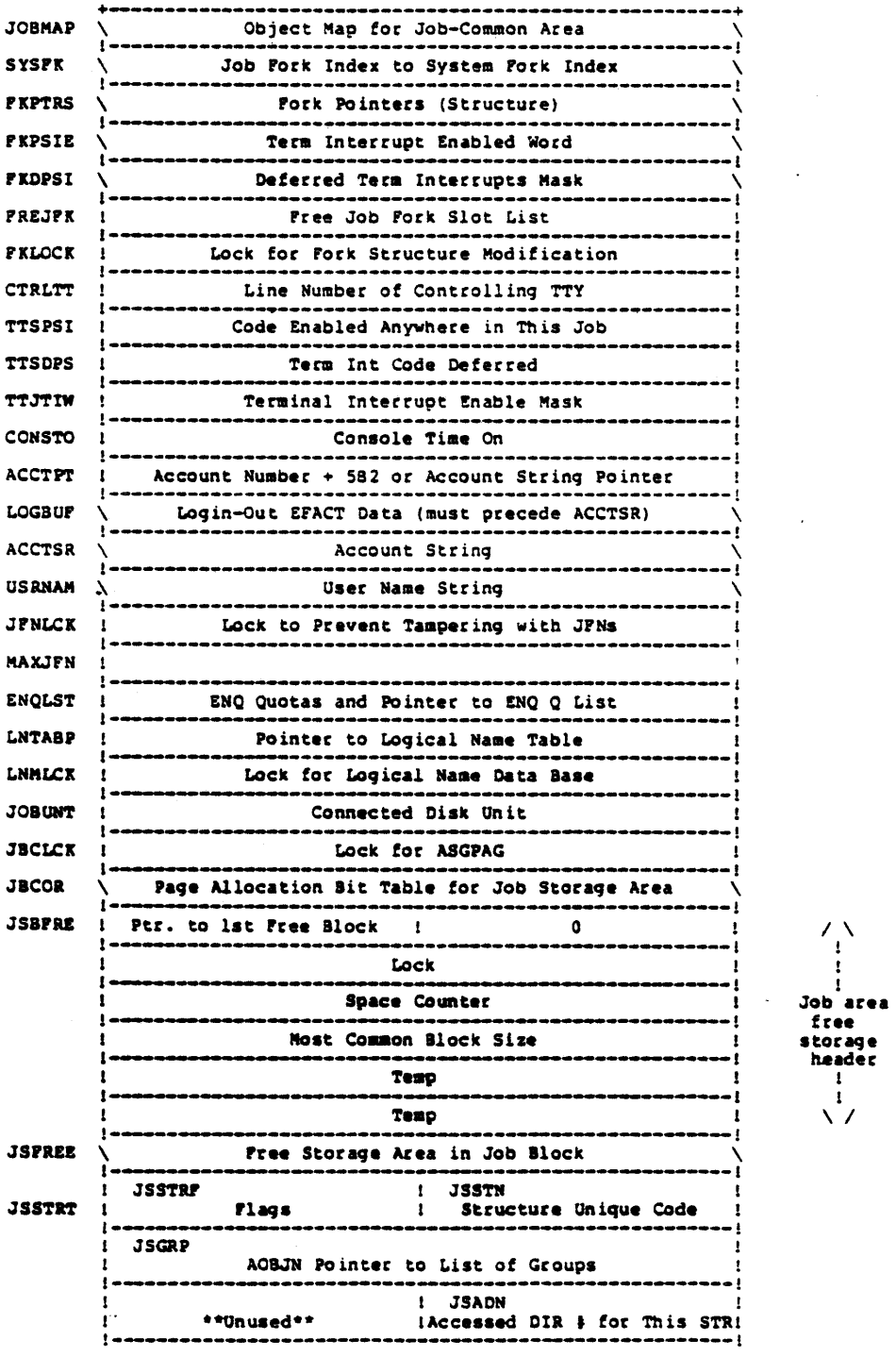
Name: JSB

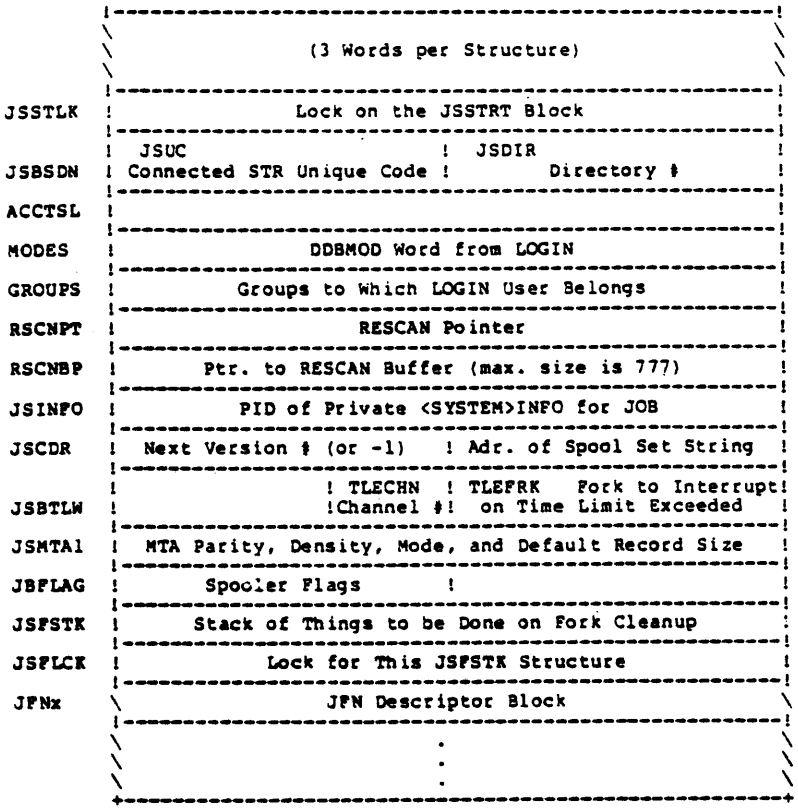
Description: Job Storage Block

Defined in: STG

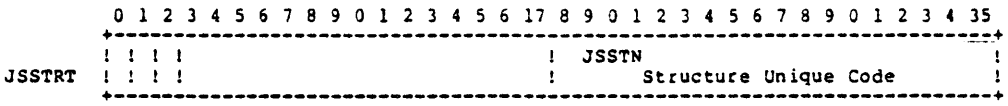
Referenced by: PAGEM, SCHED, FORK

Format

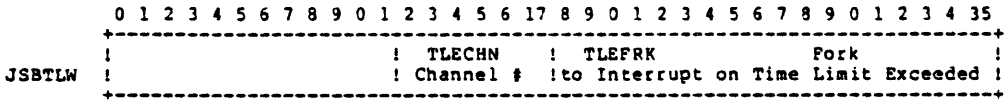




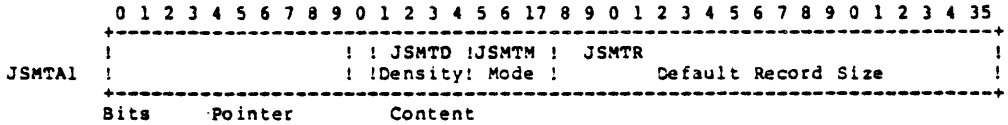
Each JFN uses a block of 18 words. (Since JFNs can grow beyond the end of the JSB into successive pages, the JFN blocks must be the last storage defined in the JSB.)



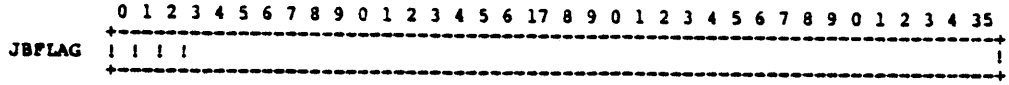
Bit	Pointer	Content
0	JSSDM	Structure is dismounted
1	JSMCI	Mount count has been incremented by structure
2	JSXCL	Structure is mounted exclusively by the structure



Bits	Pointer	Content
12-17	TLECHN	Channel number
17-35	TLEFRK	Fork to interrupt on time limit exceeded

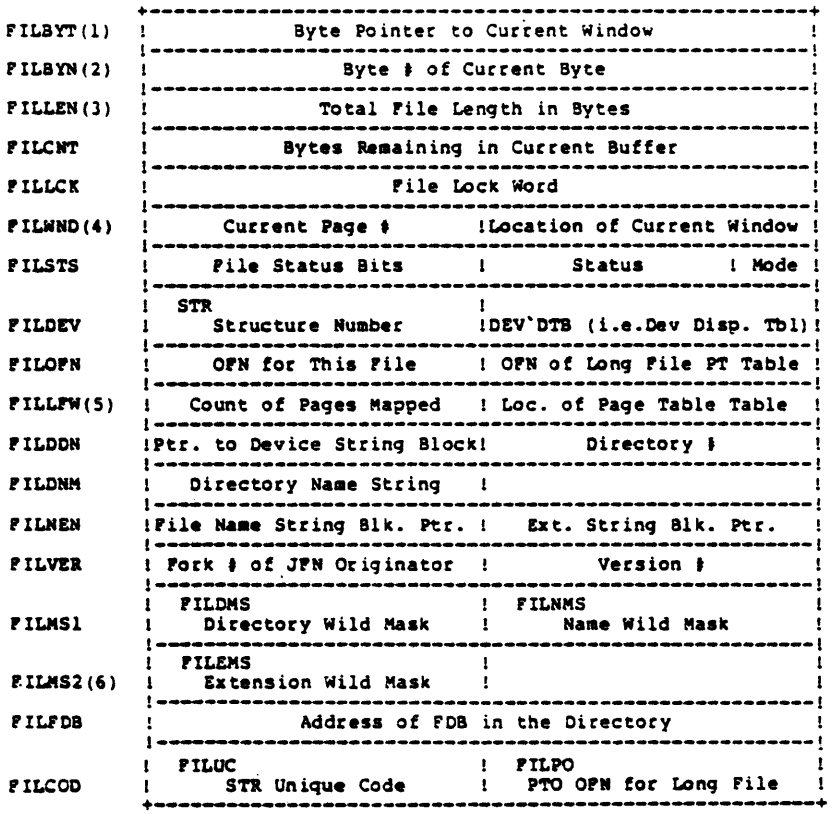


Bits	Pointer	Content
10	JSMTP	Parity
11-14	JSMTD	Density
15-17	JSMTM	Mode
18-35	JSMTR	Default record size



Symbol	Bits	Pointer	Content
SP%BAT	0		
SP%BAT	0		Job is being controlled by BATCH
SP%DFS	1		Spooling is deferred
	18		Job has been in the mini-exec

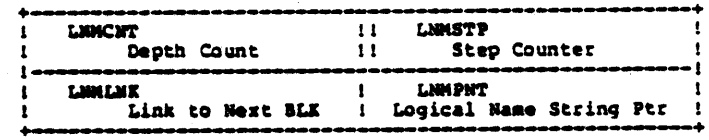
JFN descriptor block format:



These definitions are used in the above positions only during the GTJFN procedure:

(1) FILTMP / Ptr. to temp string block for default ,, Ptr. to temp string block
(2) FILPTR / Ptr. to protection string or protection #
(3) FILACT / Ptr. to account string or acount #
(4) FILOPT / Byte ptr. to store string in GTJFN
(5) FILLNM / Ptr. to RDTEXT buffer ,, Ptr. to logical name chain **
(6) FILIDX / 0 ,, Index into device tables for original devices GTJFNed (i.e., doesn't change during spooling)

** Logical Name Header Format



Bits	Pointer	Content
0-17	LNMCNT	Depth count for logical names
18	LNMIDX	Index into logical name tables
19-35	LNMSTP	Step counter at time of chaining
0-17	LNMLNK	Link to next chain block
18-35	LNMPNT	Pointer to logical name string

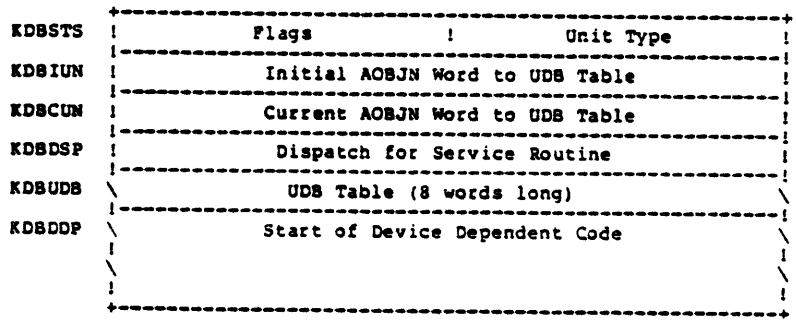
Name: KDB

Description: Kontroller Data Block (TM02 only)

Defined in: PHYPAR

Referenced by:

Format



Name: LPT

Description: Storage area for line printers. Each entry in the resident area is LPTN words long, where LPTN equals the number of line printers on the system.

Defined in: STG

Referenced by: LINPR

Format

LPTSTS	Status Word
LPTST1	Second Status Word
LPTERR	Last Error Word
LPTCNT	Buffer Counter
LPTCLS	LPTCHK Clock Switch
LPTCCW	BLKI/O Pointer
LPTICT	Interrupt Byte Count
LPTCKT	Interval for LPTTIM
LPTLCK	Lock on Opening LPT:
PGDATA	Page Counter to be Sent to -11

The following LPT: storage items are in the nonresident area of the monitor.

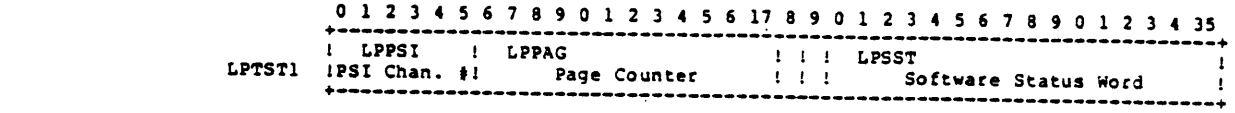
LPTBUF	2 Buffers (each 400 words) for Each LPT:
LPTOFN	VFUOFN ! RAMOFN VFU ! RAM OFN's to Prevent Opens for Write (1 entry/DTE)
VFUFIL	Swappable Storaeg Area for VFU File Names
RAMFIL	Swappable Storage Area for RAM File Names

LPTSTS

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 17 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 35
LPTFE ! LPTMX
Bytes Now in Front End ! Max. Bytes Allowed in FE

Symbol	Bits	Pointer	Content
	0-11	LPTFE	Bytes now in front end
	12-23	LPTMX	Maximum bytes allowed in front end
LP%LHC	24	LPLHC	Loading-has-completed flag for RAM or VFU
LP%HE	25	LPTHE	Hard error on this LPT:
LP%OBF	26	LPOBF	Output is being flushed
LP%MWS	27	LPMWS	MTOPR is waiting for a status to arrive
LP%ER	28	LPTER	LPT had an error
LP%OL	29	LPTOL	LPT on-line
LP%TBL	30	LPTBL	LPT is over allocation
LP%TWT	31	LPTWT	Request on Q.
LP%THN	32	LPTHN	DTEQ failed
LP%OPN	33	LPOPN	LPT is opened
LP%ALI	34	ALTI	Interrupt buffer pointer
LP%ALP	35	ALTP	Buffer Pointer

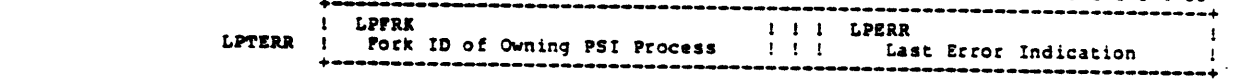
LPTST1



Symbol	Bits	Pointer	Content
	0-5	LPPSI	Channel # on which PSI's are desired
	6-17	LPPAG	Page Counter
LP%LCP	18	LPLCP	Lower case printer
LP%SHA	19	LPSHA	Status has arrived
	20-35	LPSST	Software status word

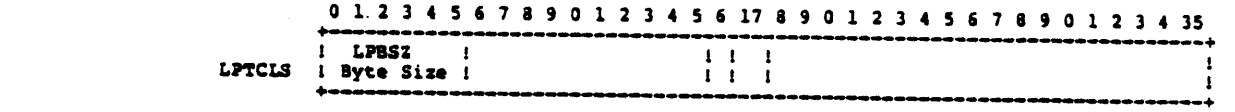
Symbol	Bits	Content
	0-27	
.DVFFE	28	Device has a fatal, unrecoverable error
.DVFLG	29	Error logging infomation follows
.DVFEF	30	EOF
.DVFIP	31	I/O in progress
.DVFSE	32	Software condition
.DVFHE	33	Hardware error
.DVFOL	34	Offline
.DVFNX	35	Nonexistent device

LPTERR



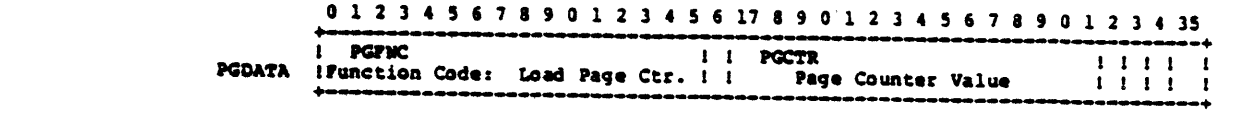
Symbol	Bits	Pointer	Content
	0-17	LPFRK	Fork ID of owning PSI process
LP%MSG	18	LPMSG	If on, suppress standard messages
LP%PCI	19	LPPCI	Page counter has interrupted
	20-35	LPERR	Last error indication

LPTCLS



Symbol	Bits	Pointer	Content
	0-5	LPBSZ	Byte size of OPENF
LP%RLD	16	LPRLD	Front end was reloaded
LP%NOE	17	LPNOE	Note occurrence of EOF

PGDATA



Symbol	Bits	Pointer	Content
	0-15	PGFWC	Function code: load page counter
	16	PGENB	Enable interrupts
	17-31	PGCTR	Page counter value
LP%IRP	32	LPIRP	Interrupt request pending
LP%RBR	33	LPRBR	RAM or VFU being reloaded
LP%LTR	34	LPLTR	Translation RAM requires reloading
LP%LVF	35	LPLVF	VFU requires reloading

Name: MTA

Description: Magtape storage area; each entry (unless otherwise noted) is MTAN words long where MTAN equals the number of magtape units on the system.

Defined in: STG

Referenced by: MAGTAP

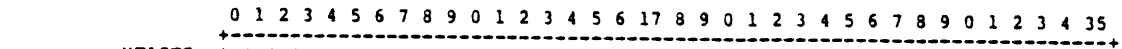
Format

MTALCK	Lock Word
MTASTS	Status of Unit
MTARS1	Resident Storage for Magtape
MTINDX	Number of Real MTAs on System
MTCUTB	CDB Table ! UDB Table
MTAPBF	Space for Buffer Page Pointers
MTIRBF	Space for IORBs
MTIOWD	IOWD for Next Transfer
MTAOLS	Length of last xfer !
MTBIOW	Backup IOWD for Next Transfer
MTARCE	Total Error Count
REWCNT	Number of Rewinding Units
MTERAS	Rewrite Erase Counter
MTPNTR	IOWD During Transfer
MTAUNT	Unit Currently Attached to Controller
MTERRC	Retry Counter
MTERFL	State of Retry
MTACOM	CONO Word of Current Operation
MTDINR	Return Address for Data Interrupt
MTACLS	Clock Routine Switch, 0 for No Clock Wanted
CHCML	DF10 Command List
M10BF	Flag - Non-0 if TM10B

The following MTA storage items are in the nonresident area of the monitor.

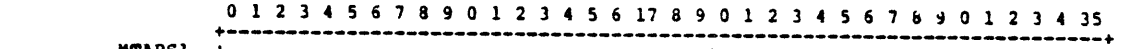
MTANR1	Flags, Density, Mode	MTRS Rec size in Hdw. Bytes
MTANR2	MTBYT Initial LH of FILBYT	Ptr. to Buffer Pages List
MTANR3	MTHBW ! MTUBF ! MTCSB	MTCUB !
MTANR4	MTCIRB Current IORB in Use	MTCUP Current User Page
MTANR5	MTUBB User Bytes per Buffer	MTUBP User Bytes per Page
MTANR6	MTLCTC Last Transfer Count	MTLIRB Last Dump Mode IORB Adr.

MTASTS



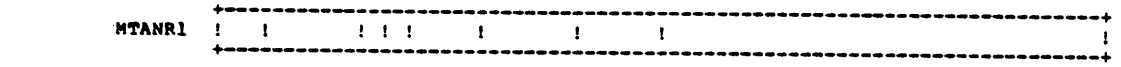
Symbol	Bits	Pointer	Content
	0	OPN	Unit has been opened
	2	OPND	Unit has been opened for dump mode
	3	DMPWT	Waiting for a dump mode to finish
	4	LTERR	Error occurred on last dump mode
	5	BUFA	Buffers have been assigned
	6	CLOF	CLOSF in progress
	7	MTOWT	MTOPR in progress
MT%IEL	8		Inhibit error logging
MT%ILM	18		Illegal mode
MT%WLK	18		Write lock
MT%HER	19		Hardware device error
MT%DER	20		Data error
MT%ERT	21		No error retry
MT%EOF	22		EOF
MT%IRL	23		Illegal record length
MT%BOT	24		Beginning of tape
MT%EOT	25		Physical end of tape

MTARS1



Bits	Pointer	Content
20	ABORTF	An error occurred and IORBs aborted
21-29	MTPPB	Number of pages per buffer
30-35	MTNIR	Number of IORBs queued

MTANR1



Bits	Pointer	Content
0-1	MTNTM	Count of EOFs written
2-5	MTFCN	Last function performed
6	MTPAR	Parity
7	MTRBF	Reading backwards flag
8-10	MSSTC	CLOSF function counter
11-14	MTDN	Density
15-17	MTDM	Data mode
18-35	MTRS	Record size in hardware bytes

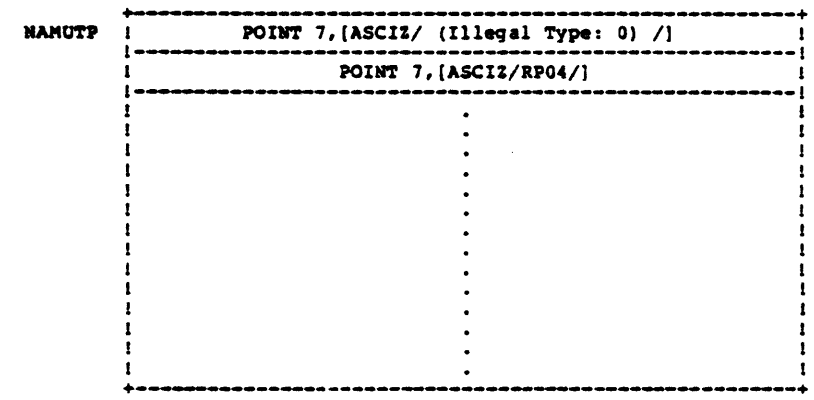
Name: NAMUTP

Description: Name Unit Type Pointers. This table contains pointers to unit type names for disks and mag tapes.

Defined in: PHYSIO

Referenced by: PHYSIO

Format



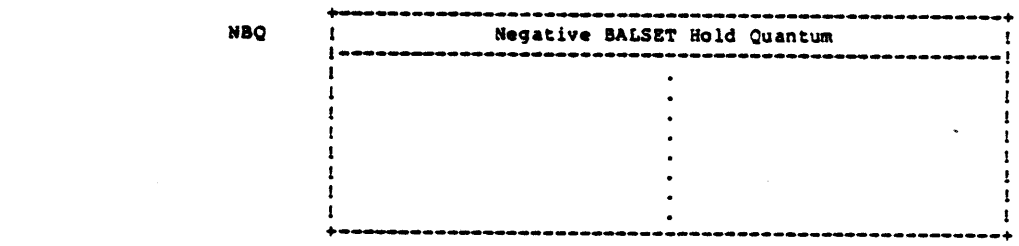
Name: NBQ

Description: Negative Balance Set Hold Quantum. This table contains for each fork in the balance set the minimum hold quantum to be used before the process becomes eligible for removal from the balance set. (Parallel table to BALSET)

Defined in: SCHED

Referenced by: SCHED

Format



Note: Balance Set Hold Time is not used in Release 2.

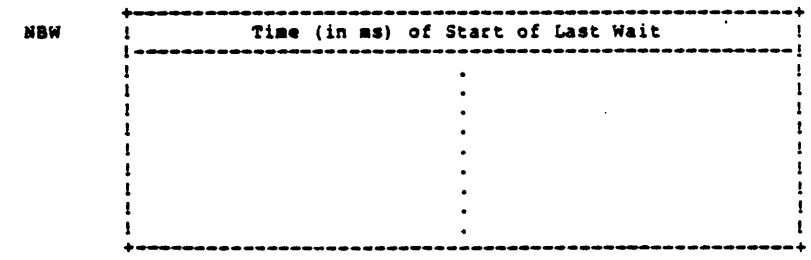
Name: NBW

Description: Balance Set Wait Time. This table records the time of day each fork in the balance set goes into a balance set wait state waiting for disk/drum I/O. When "wait satisfied" occurs, total wait time can be calculated and stored for system statistics. (Parallel table to BALSET)

Defined in: SCHED

Referenced by: SCHED

Format



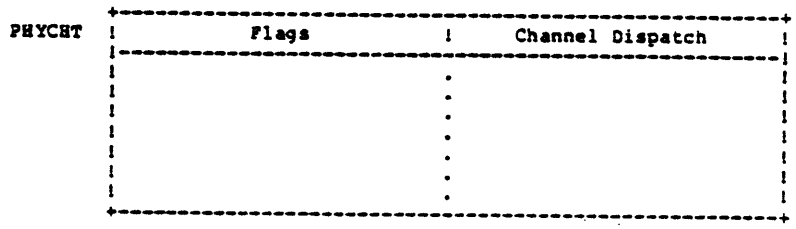
Name: PHYCHT

Description: Table of known channel dispatch routines used by PHYSIO for the different types of devices.

Defined in: STG

Referenced by: PHYSIO

Format



Note: This table currently contains only one entry .CTRH2,,RH2DSP

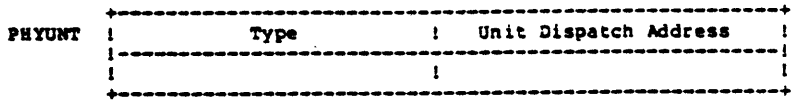
Name: PHYUNT

Description: Table of known unit dispatch routines, (i.e., one for disk and one for magtape).

Defined in: STG

Referenced by: PHYH2

Format



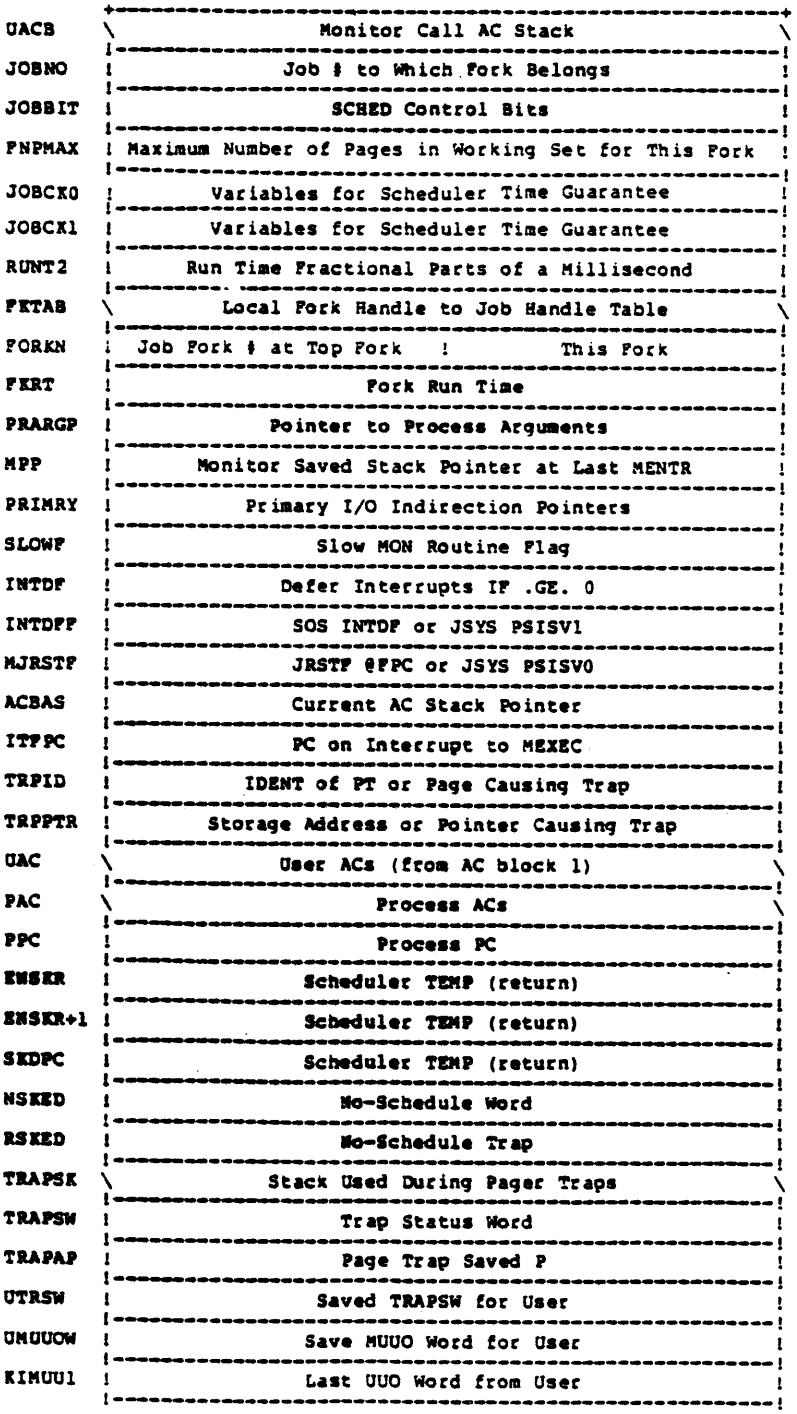
Name: PSB

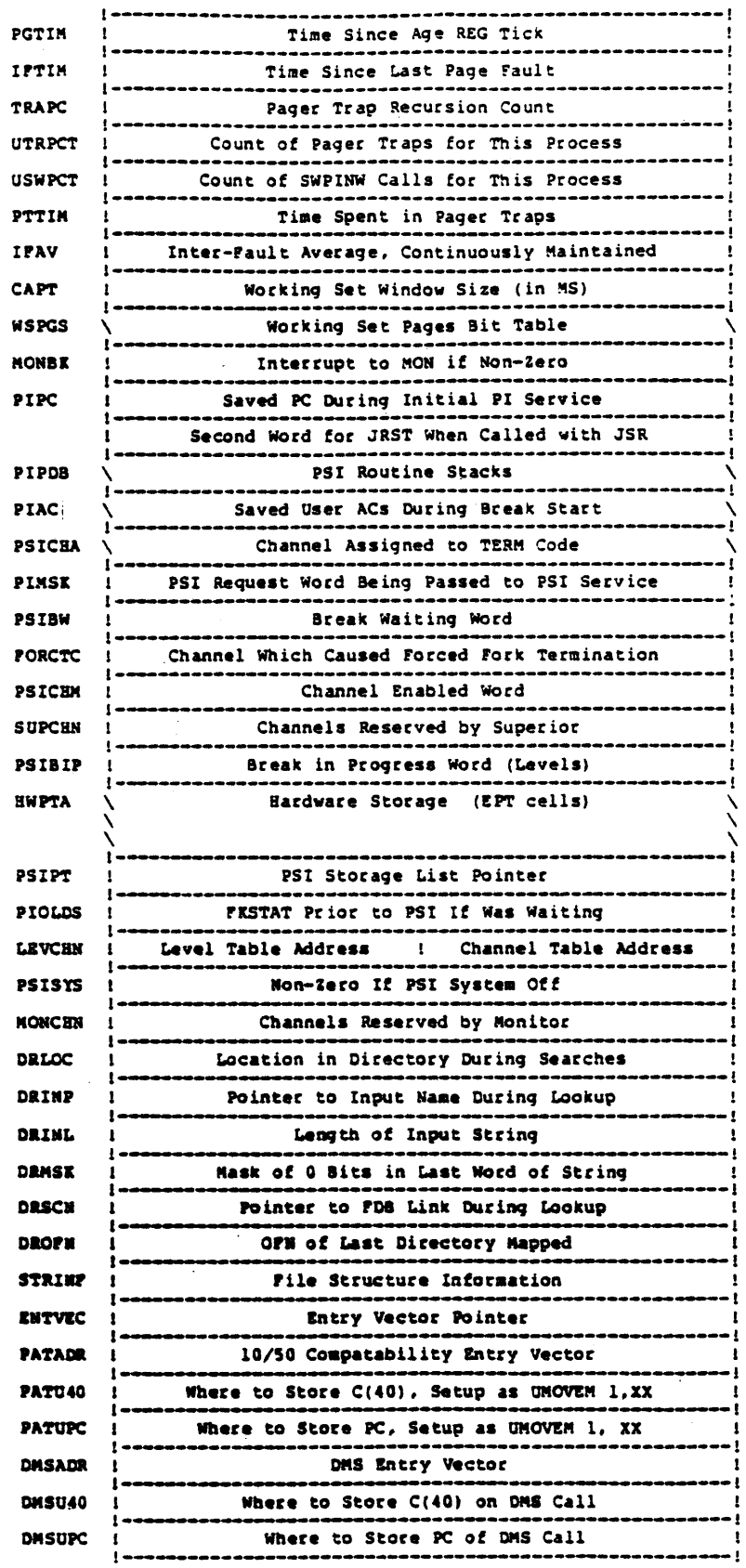
Description: Process Storage Block

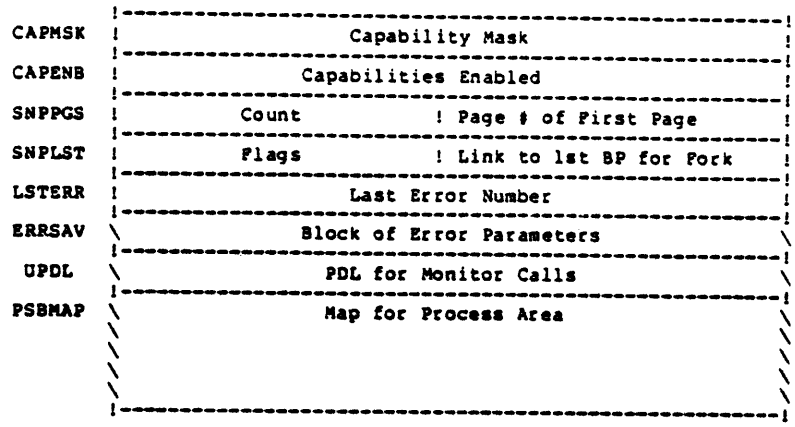
Defined in: PROLOG

Referenced by: APRSRV, DATIME, DIRECT, DISC, DSKALC, DTESRV, ENQ, FESRV, FILINI, FILMSC, FORK, FREE, GTJFN, IO, IPCF, JSYSA, JSYSF, LINEPR, LOGNAM, MAGTAP, MEXEC, PAGEM, PHYSIO, POSTLD, SCHED, SYSERR

Format




PGTIM
Time Since Age REG Tick
IFTIM
Time Since Last Page Fault
TRAPC
Pager Trap Recursion Count
UTRPCT
Count of Pager Traps for This Process
USWPCT
Count of SWPINW Calls for This Process
PTTIM
Time Spent in Pager Traps
IFAV
Inter-Fault Average, Continuously Maintained
CAPT
Working Set Window Size (in MS)
WSPGS
Working Set Pages Bit Table
MONBK
Interrupt to MON if Non-Zero
PIPC
Saved PC During Initial PI Service
Second Word for JRST When Called with JSR
PIPDB
PSI Routine Stacks
PIAC
Saved User ACs During Break Start
PSICHA
Channel Assigned to TERM Code
PIMSK
PSI Request Word Being Passed to PSI Service
PSIBW
Break Waiting Word
FORCTC
Channel Which Caused Forced Fork Termination
PSICHM
Channel Enabled Word
SUPCHN
Channels Reserved by Superior
PSIBIP
Break in Progress Word (Levels)
HWPTA
Hardware Storage (EPT cells)
PSIPT
PSI Storage List Pointer
PIOLDS
FKSTAT Prior to PSI If Was Waiting
LEVCHN
Level Table Address
Channel Table Address
PSISYS
Non-Zero If PSI System Off
MONCHN
Channels Reserved by Monitor
DRLOC
Location in Directory During Searches
DRINP
Pointer to Input Name During Lookup
DRINL
Length of Input String
DRMSK
Mask of 0 Bits in Last Word of String
DRSCN
Pointer to FDB Link During Lookup
DROFN
OFN of Last Directory Mapped
STRINF
File Structure Information
ENTVEC
Entry Vector Pointer
PATADR
10/50 Compatability Entry Vector
PATU40
Where to Store C(40), Setup as UMOVEM 1,XX
PATUPC
Where to Store PC, Setup as UMOVEM 1, XX
DMSADR
DMS Entry Vector
DMSU40
Where to Store C(40) on DMS Call
DMSUPC
Where to Store PC of DMS Call


CAPMSK
Capability Mask
CAPENB
Capabilities Enabled
SNPPGS
Count
Page # of First Page
SNPLST
Flags
Link to 1st BP for Fork
LSTERR
Last Error Number
ERRSAV
Block of Error Parameters
UPDL
PDL for Monitor Calls
PSBMAP
Map for Process Area

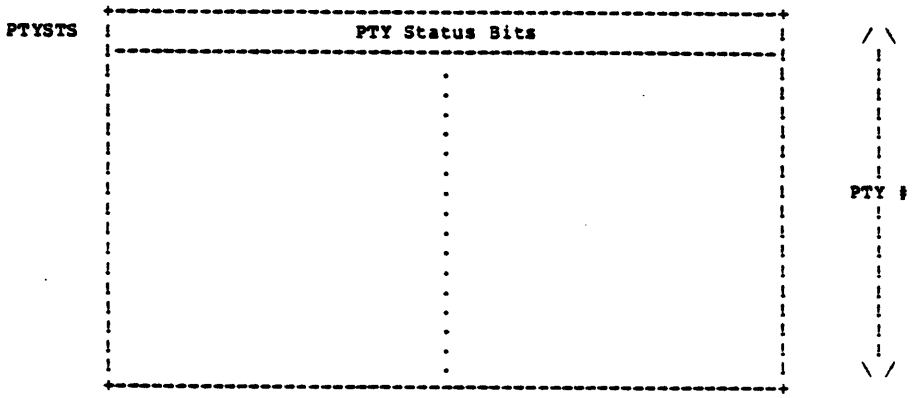
Name: PTYSTS

Description: Pseudo Terminal Status Table. This table, indexed by PTY, contains the PTY's status word.

Defined in: STG

Referenced by: FILMSC

Format



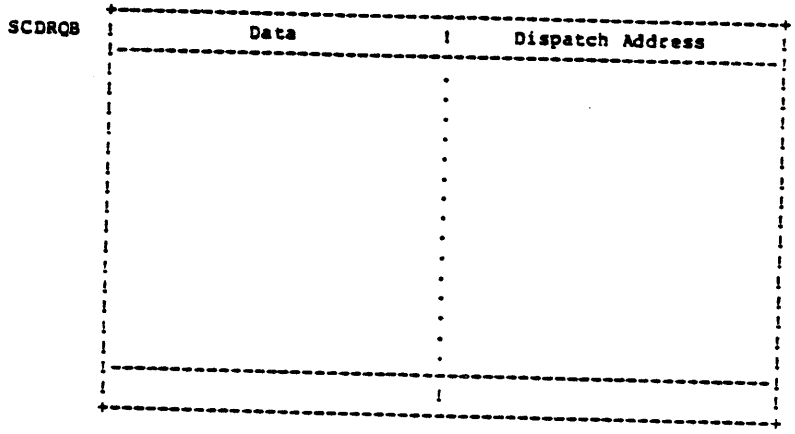
Name: SCDRQB

Description: Scheduler Request Table. During the scheduler's overhead cycle, the initial job startup request is placed in this table when the first CTRL/C is processed. The table is later examined in the same cycle and all entries are processed by dis patching to each entry's dispatch address.

Defined in: SCHED

Referenced by: SCHED, TTYSRV

Format

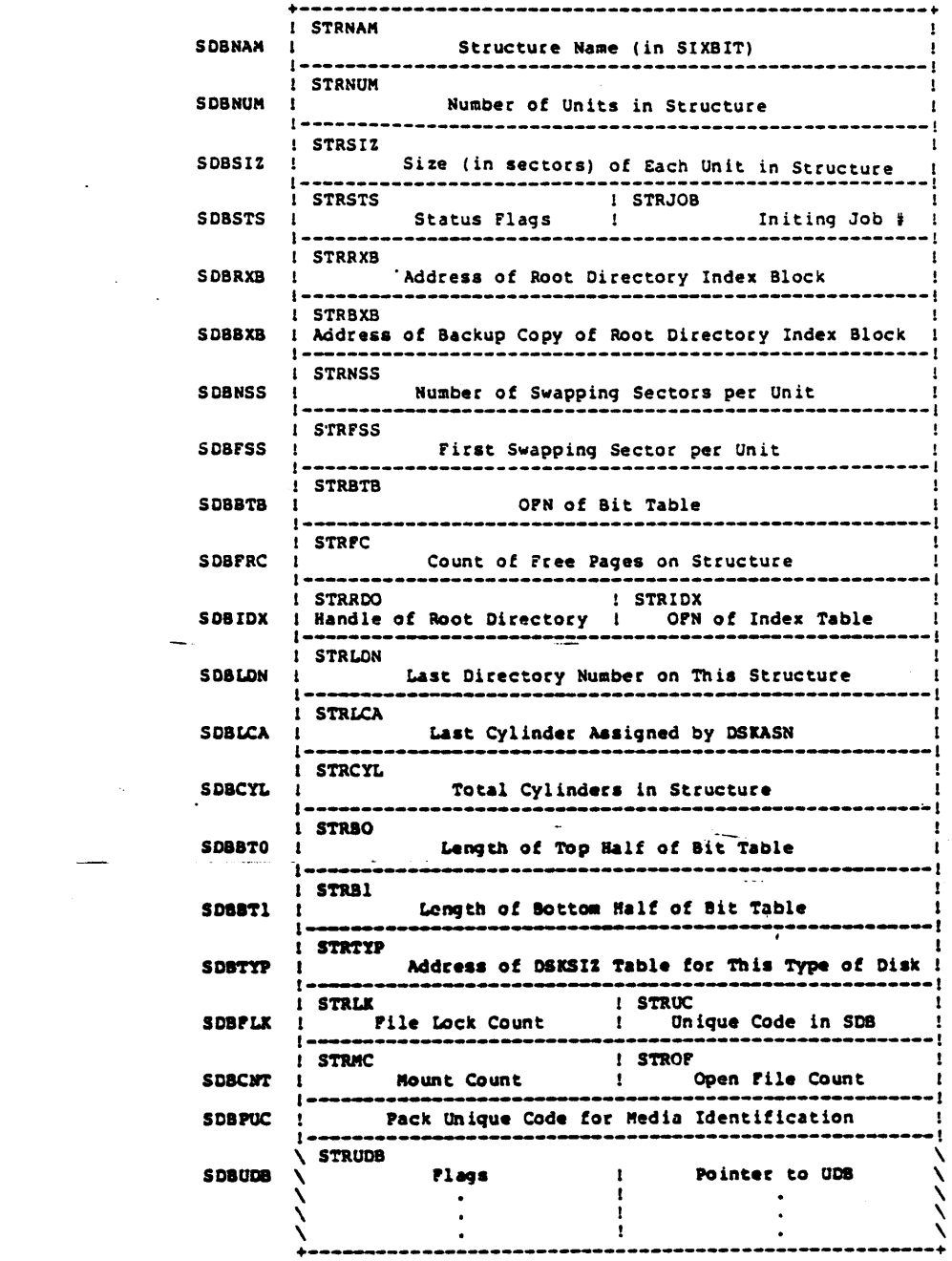


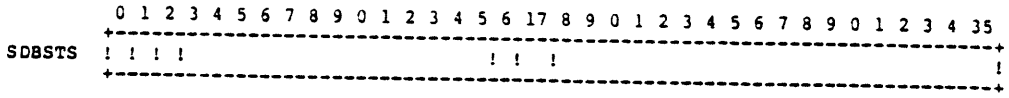
Name: SDB

Description: Structure Data Block. This block, one per structure, contains information about the structure's units, master directory (i.e. Root-Directory), bit map for disk page allocation/deallocation, and assigned swapping area. It also contains mount and open-file information. SDBBLK is the name of the storage area reserved for handling SDBs.

Referenced by: DSKALC, FILINI, PHYSIO

Format





Symbol	Bits	Pointer	Content
MS%PS	0	STPS	Structure is public
MS%DIS	1	STDIS	Structure is being dismounted
MS%DOM	2	STDOM	Structure is domestic
	16	STIDX	Index table file OFN has been set up
	17	STCRD	Creating Root Directory on this Structure
	18-35	STRJB	Initializing job # (only legal user while structure is being initialized)

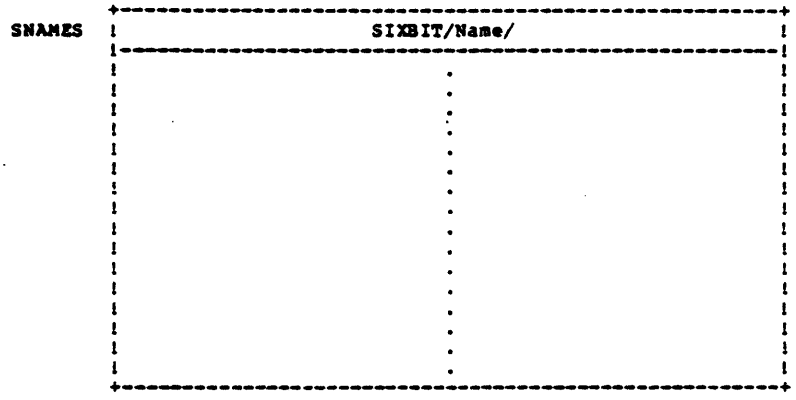
Name: SNAMES

Description: Subsystem Names. Each entry contains a subsystem program name.

Defined in: STG

Referenced by: MEXEC

Format



Name: SNBLKS

Description: Subsystem Blocks. Each entry contains the number of blocks for a subsystem program. This table is parallel to SNAMES.

Defined in: STG

Referenced by: MEXEC

Format

SNBLKS

Number of Blocks
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.

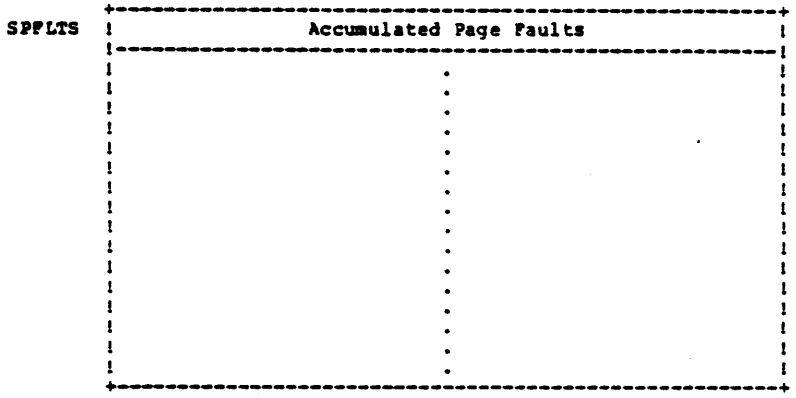
Name: SPFLTS

Description: Subsystem Page Faults. Each entry contains the accumulated number of page faults of a subsystem program. This is a parallel table to SNAMES.

Defined in: STG

Referenced by: MEXEC, PAGEM

Format



Name: SPT

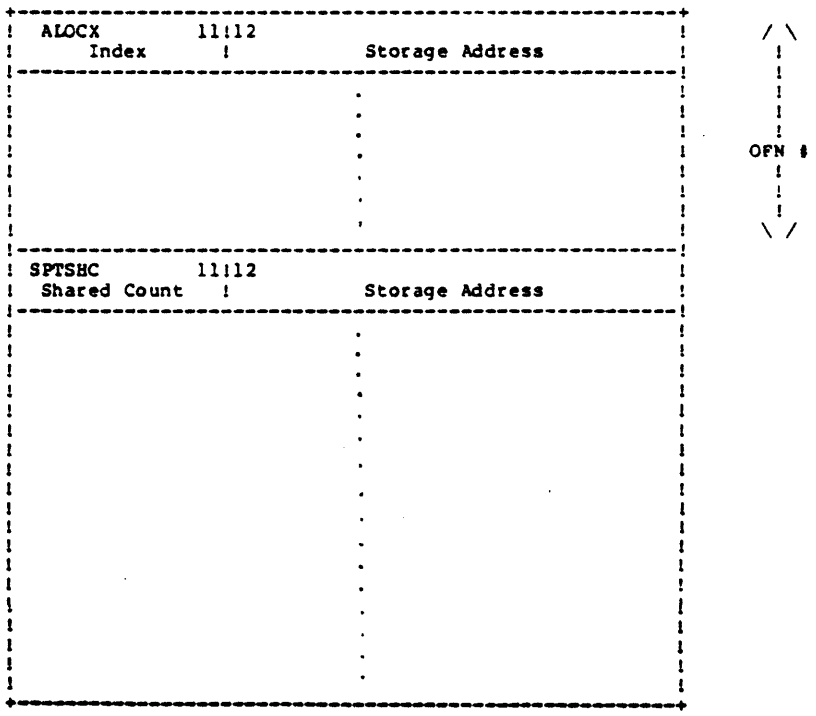
Description: Special Pages Table. This table is referenced directly by the pager. The first part of the table (of length NOFN) is used to point to index blocks in core for open files and an index into this part is often referred to as an OFN (Open File Number). The remaindor of the table is used to point to PSBs, JSBs, UPTs, (User Page Map Tables), and shared file pages.

The ALOCX value in the OFN area is used as an index into the allocation tables (ALOC1 & ACOC2) to obtain information about the directory of the open file, (i.e., pages left in quota). The share count in the non - OFN area is indexed for each sharing of the page.

Defined in: STG

Referenced by: APRSRV, FORK, PAGEM, SCHED

Format



STORAGE ADDRESS

UAAB = 1817 UNASSIGNED BIT IN STORAGE ADR

UAACB = 1817+1835 UNASSIGNED & COPY

DRMAB = 2B17 DRUM BIT IN STORAGE

DRMOB = 1B17 DRUM OVERFLOW (WITH DRMAB) IN STG ADR

DSKAB = 1B14 DSK BIT IN STORAGE ADR

DSKNB = 1B15 WITH DSKAB NEWLY ASSIGNED ADDRESS.

NOFN = NUMBER OF OFN SLOTS

Name: SPTH

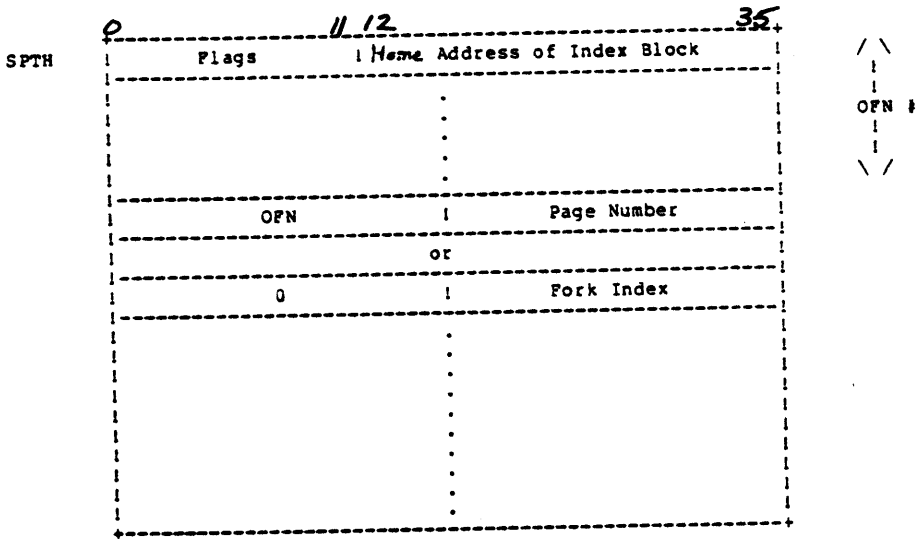
Description: Parallel table to SPT. This table is referenced only by the software and is divided into two parts. The first part, indexed by OFN, is used to point to the home address of each open file (i.e., to its index block) and to hold status information about each OFN.

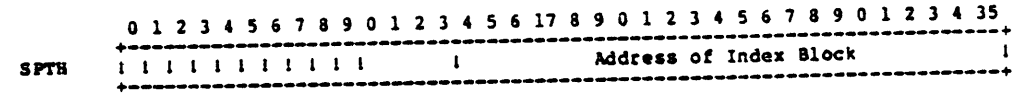
The second part is used mainly to show the page's origin. For a shared file, this is indicated by OFN ,, Page Number , where page number is within open file, OFN. For PSBs, JSBs, and UPTs, the SPTH word contains 0 ,, Fork Index. The free slots in this part are on a list chained through the SPT where the free list pointer resides in FRESPT.

Defined in: STG

Referenced by: DISC, DSKALC, FILINI, PAGEM

Format





Symbol	Bits	Content
FILNB	1	File write bit in SPTH and ASOFN argument
THAWB	2	Thawed bit
FILNB	3	"File new" bit
SPTLKB	4	LH of SPTH(OFN), XB(Index Block) in use by DDMP
OFNWRB	5	OFN has been modified
OFNBAT	6	Index block contains a bad block
OFNERR	7	Error in file (i.e., MPE)
OFNDMO	8	OFN is on a dismounted structure
OFNDUD	9	Suppress DDMP

SPAREN 10-11 SPARE BITS IN SPTH

Note: A file is opened by searching the OFN part of SPTH for the index block address. If the address is found and the write and thawed bits are legal, it is a shared opening and the same index is used. If the address is not found, a new entry is made from one of the free (-1) slots in SPTH.

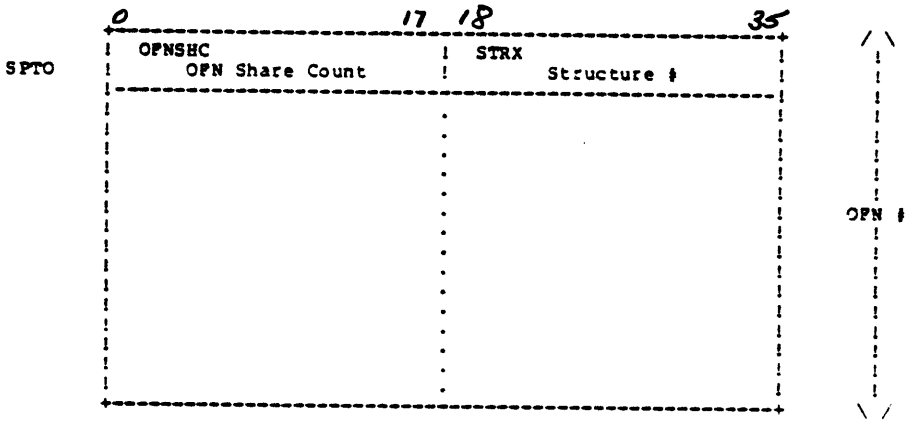
Name: SPTO

Description: Special Pages Table O. This table is parallel to the OFN area of the SPT table and contains the structure number and open file share count for each open file. The OFN share count is indexed for each opening of the file and for each shared page within the open file.

Defined in: STG

Referenced by: PAGEM

Format



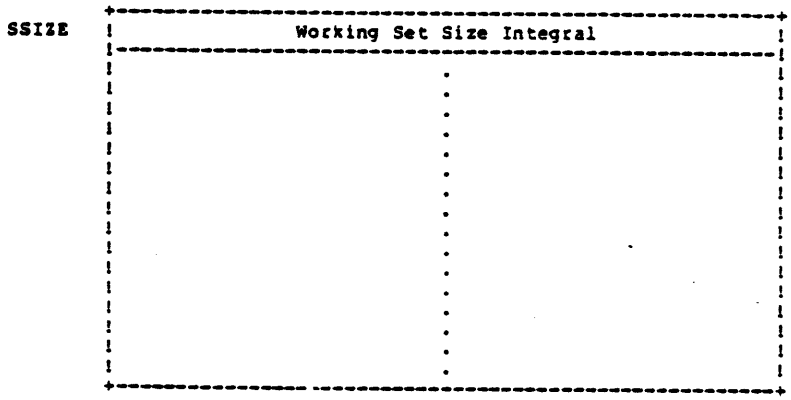
Name: SSIZE

Description: Subsystem Working Set Size. Each entry contains the working set size integral for a subsystem program. This is a parallel table to SNAMES.

Defined in: STG

Referenced by: MEXEC

Format



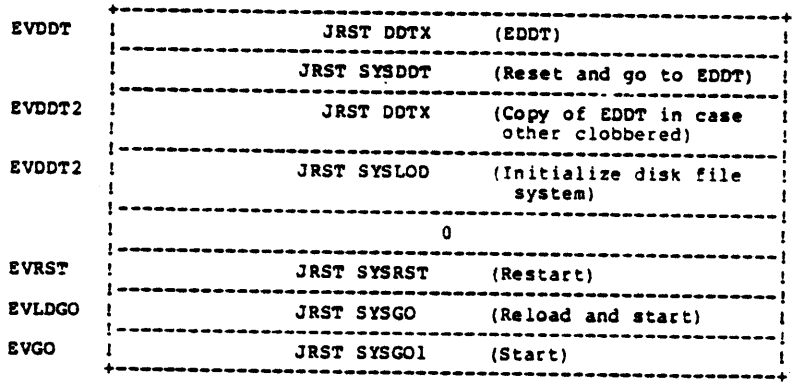
Name: STARTUP

Description: Startup Transfer Vectors. This table, in resident locations 140-147, contains the startup vectors for the monitor as well as vectors to enter EDDT.

Defined in: STG

Referenced by: STG, POSTLD

Format



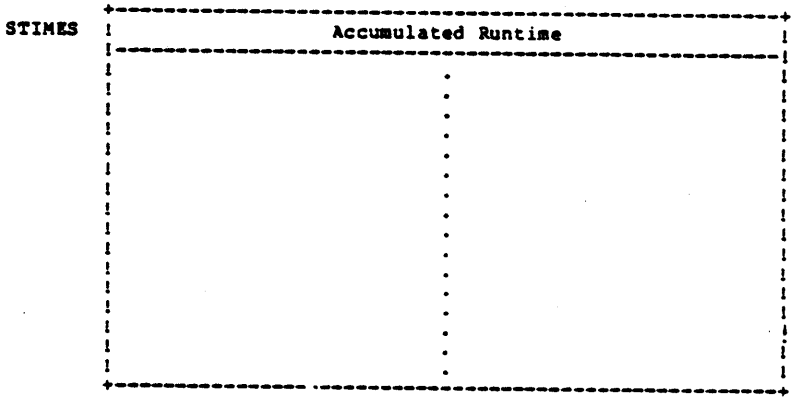
Name: STIMES

Description: Subsystem Runtimes. Each entry contains the accumulated runtime of a subsystem program. This is a parallel table to SNAMES.

Defined in: STG

Referenced by: MEXEC

Format



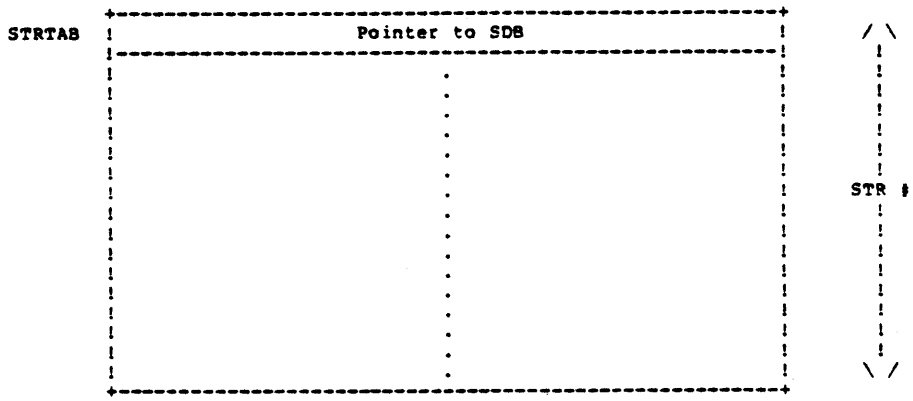
Name: STRTAB

Description: Structure Data Block Table. This table, indexed by structure number, contains pointers to each structure data block in the system.

Defined in: STG

Referenced by: DSKALC, PHYSIO

Format



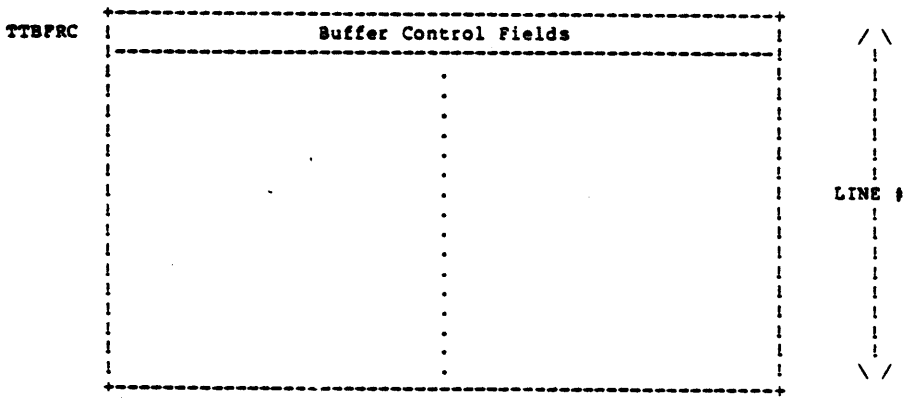
Name: TTBFRC

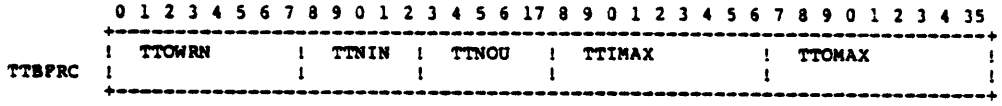
Description: Teletype Buffer Control Table. This table, indexed by line number, contains buffer control fields.

Defined in: STG

Referenced by: TTYSRV

Format





Bits	Pointer	Content
0-7	TTOWRN	Number of characters in output buffer for wakeup after buffer
8-12	TTNIN	Number of input buffers to assign
13-17	TTNOU	Number of output buffers to assign
18-26	TTIMAX	Maximum bytes in input buffer(s)
27-35	TTOMAX	Maximum bytes in ouput buffer(s)

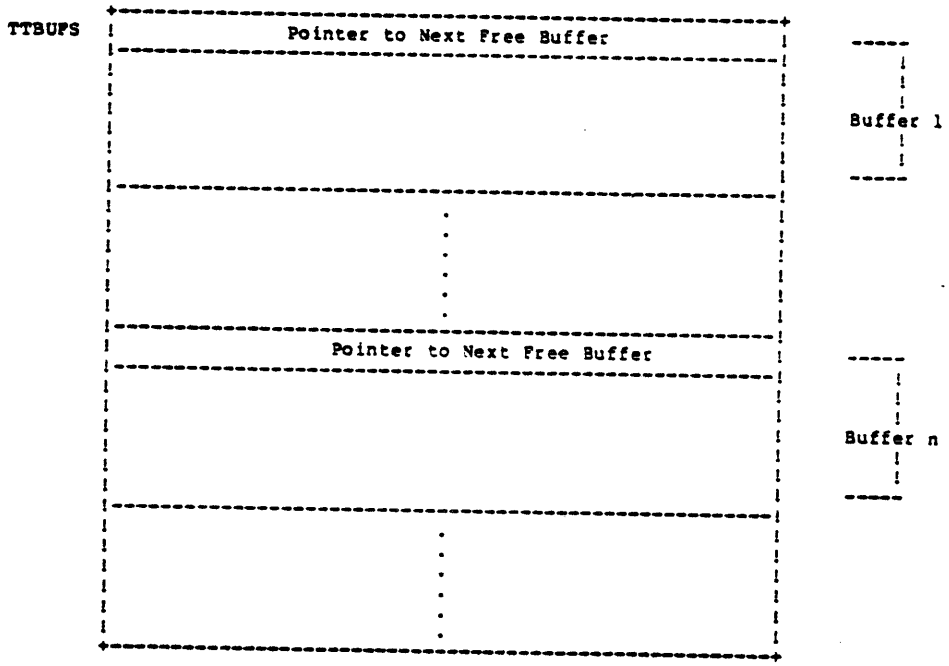
Name: TTBUFS

Description: Teletype Buffers. This storage area contains the input and output buffers for each line (TTY and PTY) on the system. Input and output pointers to each buffer are kept in tables in core. These buffers are fixed length and are assigned on demand. When there is no character activity, the buffers are deassigned.

Defined in: STG

Referenced by: TTYSRV

Format



Note: The free buffers are linked and are pointed to by TTFREB.

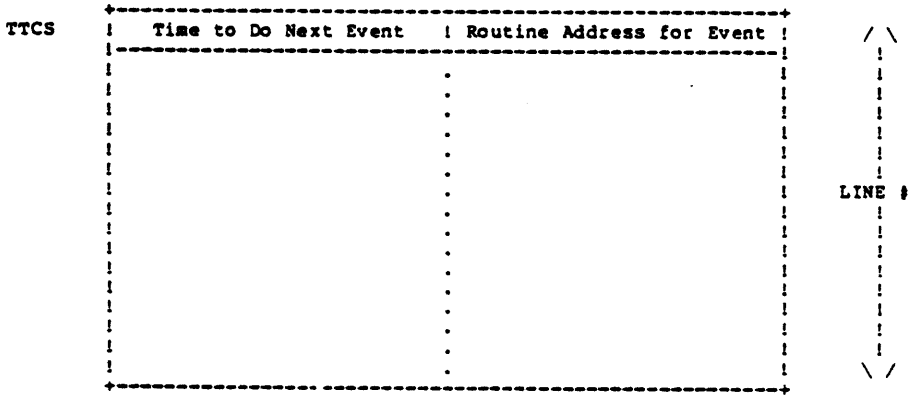
Name: TTCS

Description: Teletype Control State. This table, indexed by line number, contains the control states of each line.

Defined in: STG

Referenced by: TTYSRV

Format



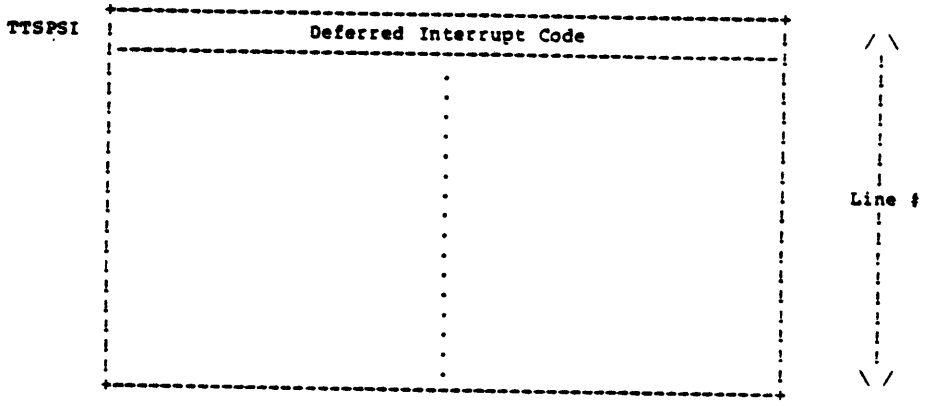
Name: TTDPSI

Description: Teletype Deferred Pseudo Interrupt Table. This table, indexed by line number, contains the deferred pseudo interrupt code for each terminal. The bit for the terminal code is on if a deferred interrupt is assigned. (i.e. Channel interrupt processing on character typed will wait until the character is read by the program.)

Defined in: STG

Referenced by: TTYSRV, FORK, MEXEC

Format



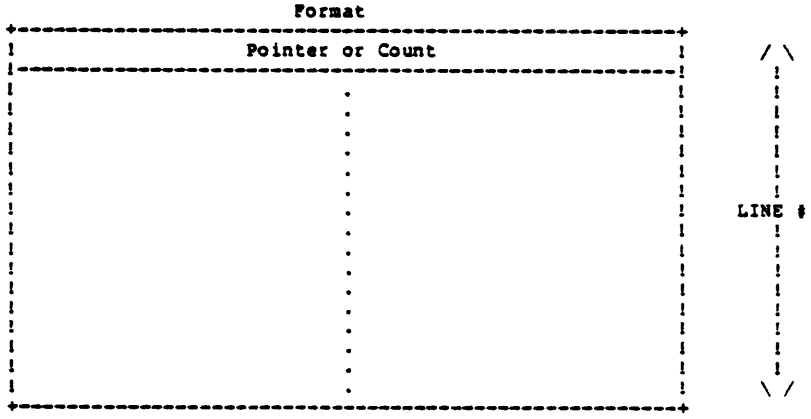
Name: TTExx

Description: Teletype Echo Output Buffer Tables. There are 3 echo output buffer tables (TTECT, TTEIN, TTEOUT), each indexed by line number. Each entry in TTECT contains a character counter. TTEIN and TTEOUT contain input and output pointers to the echo output buffer for each line. The echo buffer is used when the line's output buffer is full.

Defined in: STG

Referenced by: TTYSRV

Format



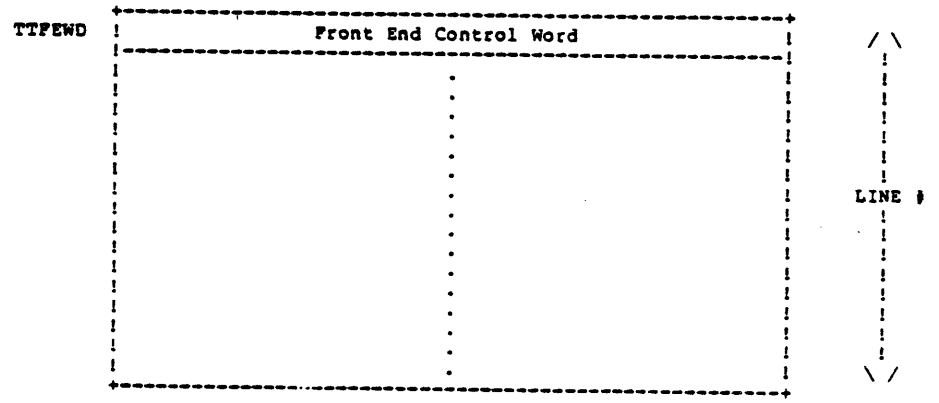
Name: TTFEWD

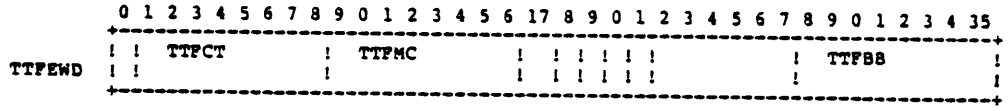
Description: Teletype Front End Word Table. This table, indexed by line number, contains the control word for the front end.

Defined in: STG

Referenced by: TTYSRV

Format





Bits	Pointer	Content
0	TTFPK	Waiting for "to -11 done"
1-8	TTFCT	Bytes now in front end
9-16	TTFMC	Maximum count for front end buffer
17	TTFOW	Line allocation exceeded
18	TTFEM	Line is remote
19	TTFSP	Remote and needs speed set
20	TTFXO	Needs x-on sent
21	TTNTS	Don't send system messages
22-27		Not used
28-35	TTFBB	Number of entries in Big Buffer, TTBBFU

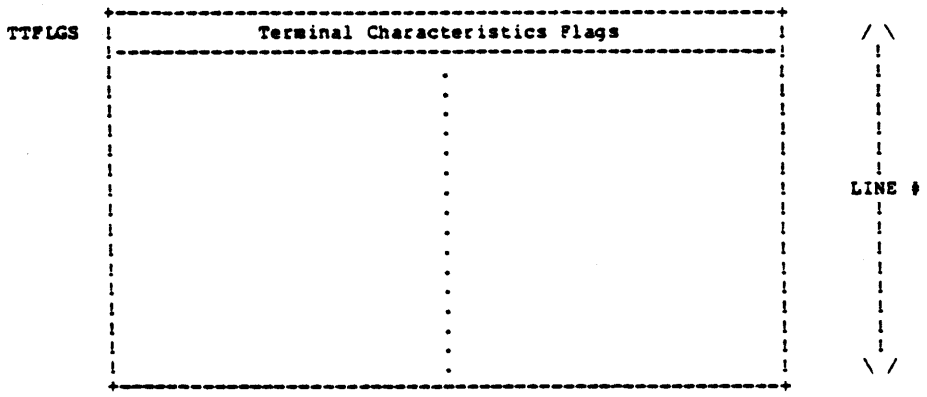
Name: TTFLGS

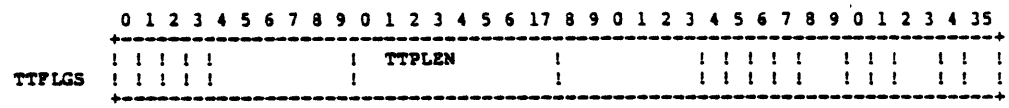
Description: Teletype Flags Table. This table, indexed by line number, contains information about the terminal characteristics.

Defined in: STG

Referenced by: TTYSRV, JSYSA

Format





Symbol	Bits	Pointer	Content
	0		Line is active (interrupt expected from hardware)
TT%MFF	1		Mechanical form feed
TT%TAB	2		Mechanical tab
TT%LCA	3		Lower case
	4-9		Not used
	10-17	TTPLEN	Page Length
TT%WAK	18-23		Wakeup control characters
TT%ECO	24		Echos on
TT%ECM	25		Echo immediate
TT%ALK	26		Accept links
TT%AAD	27		Accept advise
TT%DAM	28-29		Terminal data mode
TT%UOC	30		Flag upper case
TT%LIC	31		Raise lower case
TT%DUM	32-33		Duplex mode
TT%PGM	34		Page mode
TT%CAR	35		Carrier on

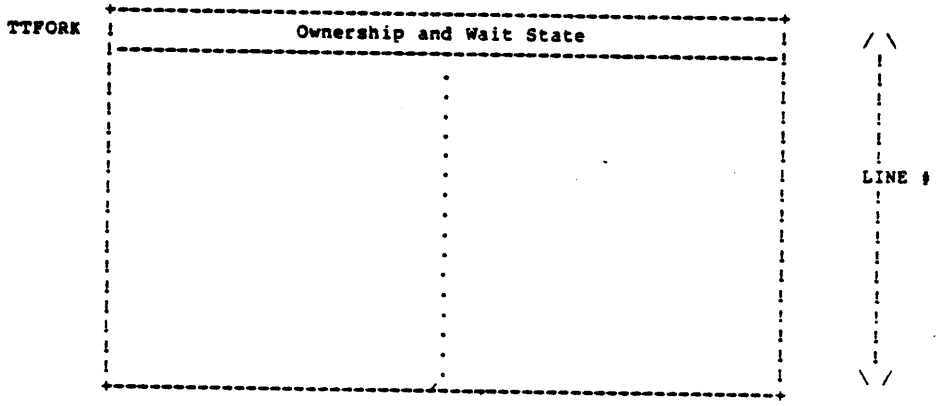
Name: TTFORK

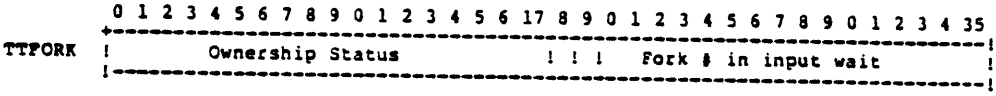
Description: Teletype Fork Table. This table, indexed by line number, contains information about the terminal's ownership and input wait states.

Defined in: STG

Referenced by: TTYSRV, DEVICE, FILMSC, FORK, IO, JSYSA, MEXEC, SCHED

Format





Bits	Content
0-17	Controlling job number, -1 if not a controlling terminal, or -2 if becoming a controlling terminal
18	On if no fork is in input wait
19	Force wakeup because input buffer is full
20-35	Fork number in input wait, or -1 if none

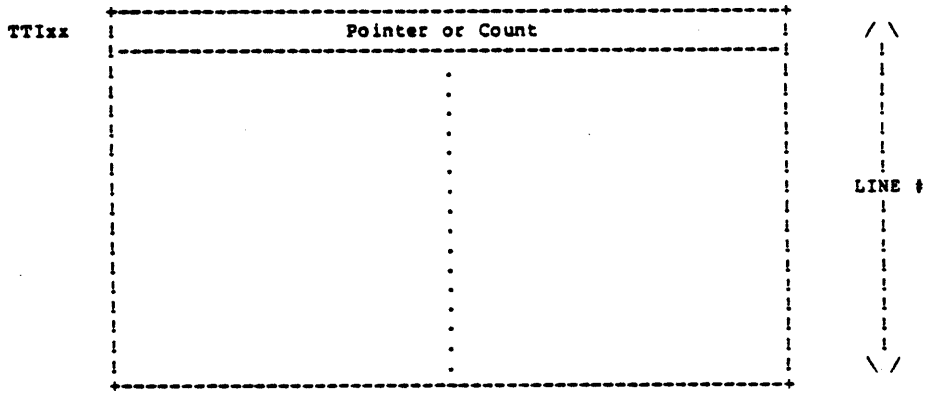
Name: TTIxx

Description: Teletype Input Buffer Tables. There are 3 input buffer tables (TTICT,TTIIN,TTIOUT), each indexed by line number. Each entry in TTICT contains a character count. TTIIN and TTIOUT contain pointer for adding characters to and removing characters from the input line buffers.

Defined in: STG

Referenced by: TTYSRV

Format



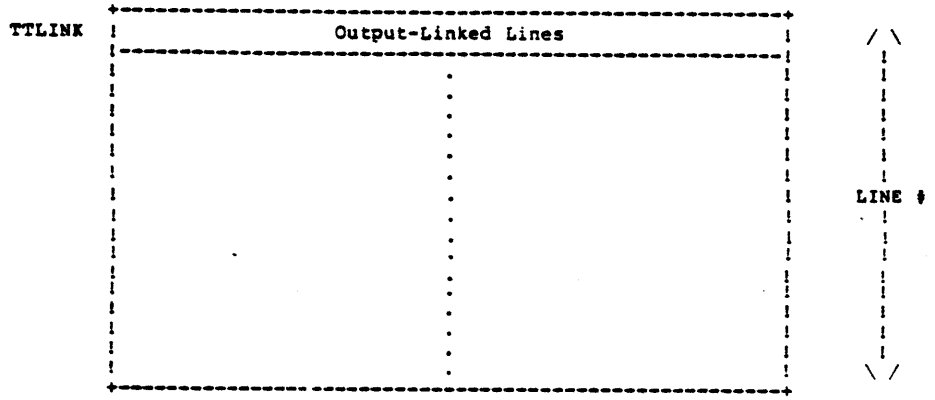
Name: TTLINK

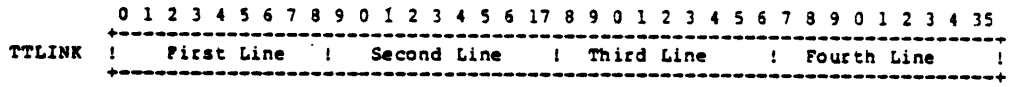
Description: Teletype Linked Lines. This table, indexed by line number, contains the linked lines for each terminal.

Defined in: STG

Referenced by: TTYSRV

Format





Bits	Content
0-9	First line linked to, -1 if none
10-17	Second line linked to, -1 if none
18-26	Third line linked to, -1 if none
27-35	Fourth line linked to, -1 if none

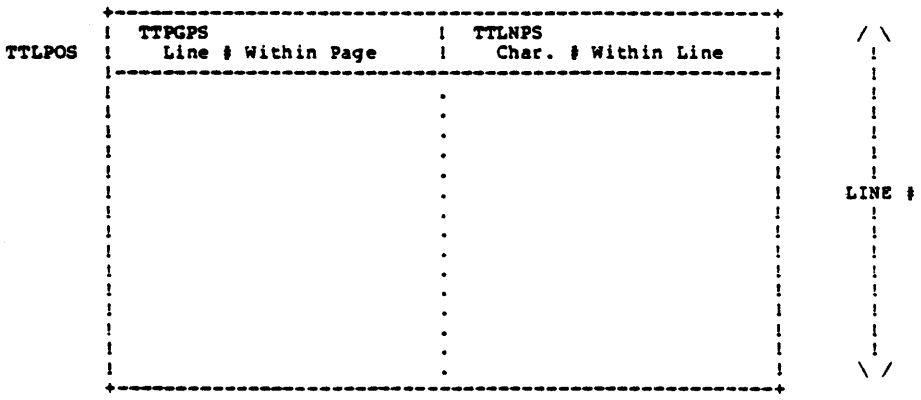
Name: TTLPOS

Description: Teletype Line Position Table. This table, indexed by line number, contains the page and line pointer for each terminal.

Defined in: STG

Referenced by: TTYSRV

Format



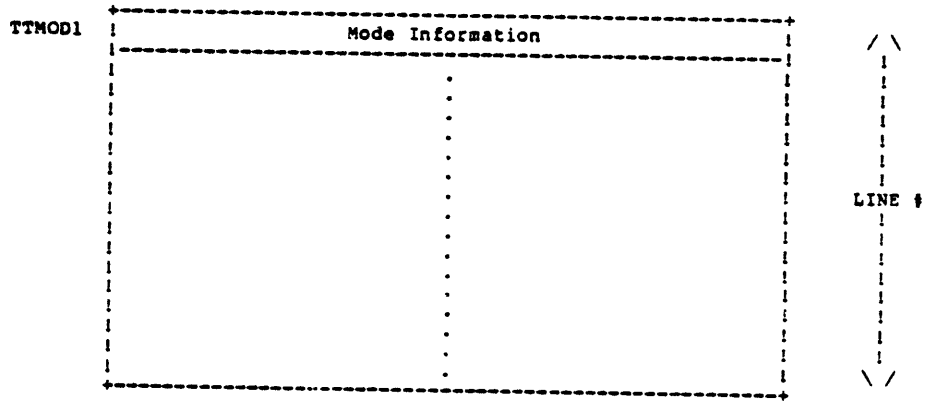
Name: TTMOD1

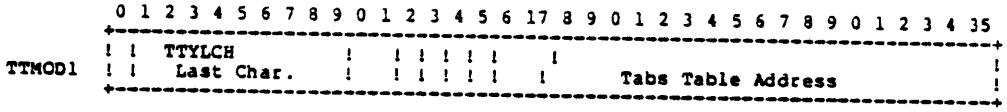
Description: Teletype Mode Table. This table, indexed by line number, contains character and mode information for each terminal.

Defined in: STG

Referenced by: TTYSRV

Format





Symbol	Bits	Pointer	Content
DCXLIN	0		Line is on a DC10
	1-9	TTYLCH	Last character removed from input line buffer
	10-11	TTYLMD	Terminal data mode for last character
TTSFLG	12		Stop Output (CTRL/S was typed)
TTRFLG	13		Repeat last input character (BKJFN)
TTWFLG	14		Wait restarted if 1 at TCITST (Blocked on input)
TTOFLG	15		Flush output (CTRL/O was typed)
	16-17		Not used
	18-35		Tabs table address

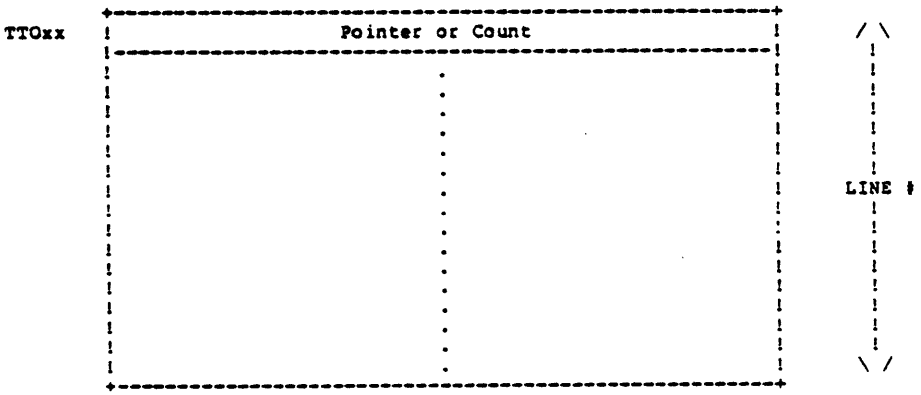
Name: TTOxx

Description: Teletype Output Buffer Tables. There are 3 output buffer tables (TTOCT, TTOOUT), each indexed by line number. Each entry in TTOCT contains a character count. TTOIN and TTOOUT contain pointers for adding characters to and removing characters from the line output buffers.

Defined in: STG

Referenced by: TTYSRV

Format



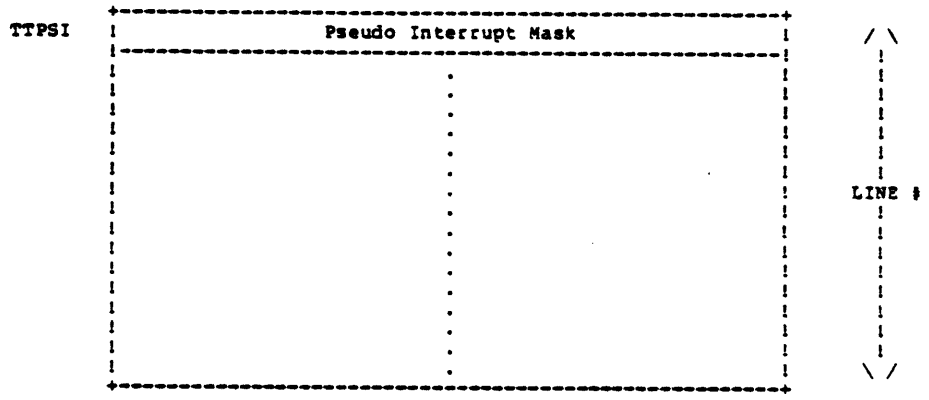
Name: TTPSI

Description: Teletype Pseudo Interrupt Table. This table, indexed by line number, contains the pseudo in terrupt mask for each terminal.

Defined in: STG

Referenced by: TTYSRV, FORK, MEXEC, SCHED

Format



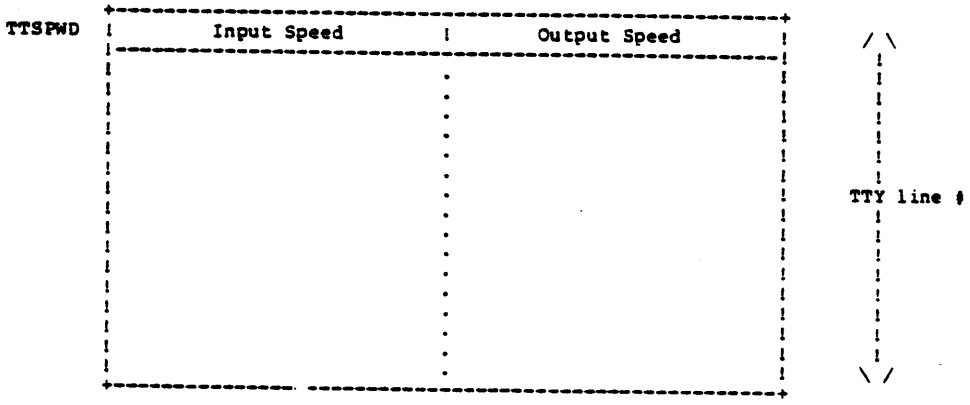
Name: TTSPWD

Description: Terminal Speed Word Table. This table, indexed by line number (TTYS only), contains the terminal speeds for each terminal.

Defined in: STG

Referenced by: TTYSRV

Format



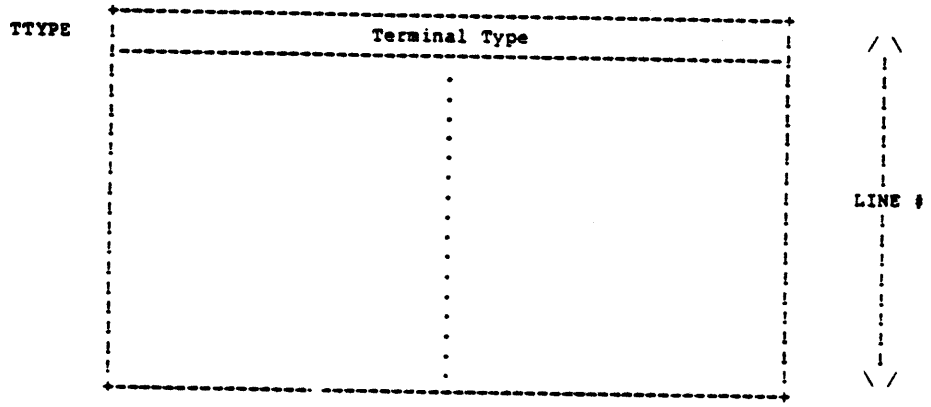
Name: TTYPE

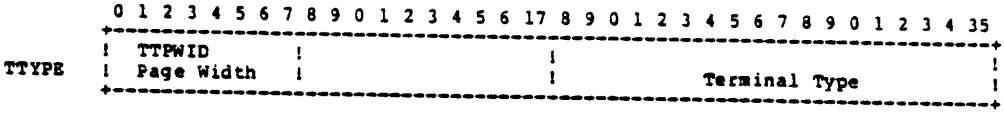
Description: Teletype Type Table. This table, indexed by line number, contains the type of terminal on the line.

Defined in: STG

Referenced by: TTYSRV

Format





Bits	Pointer	Content
0-7	TTPWID	Page width
8-17		Not used

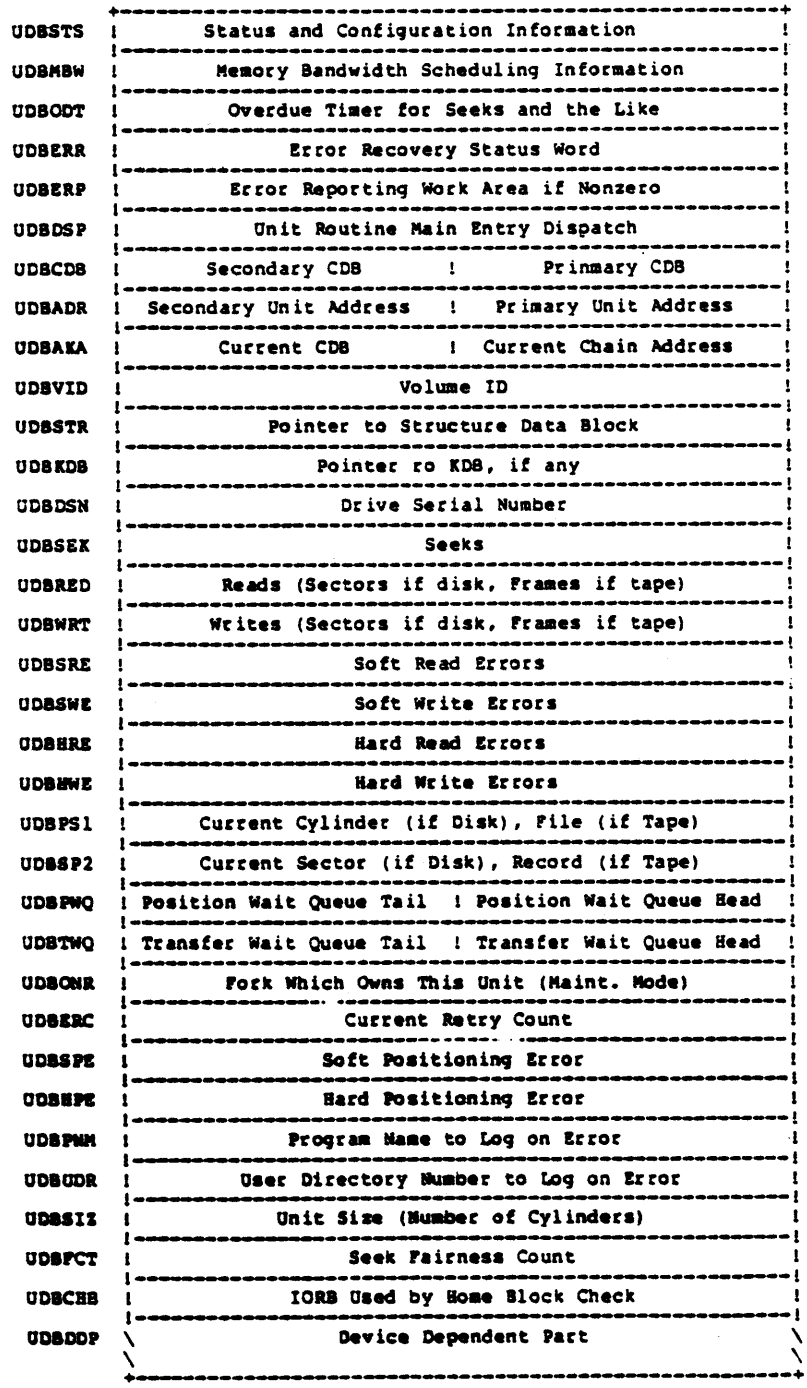
Name: UDB

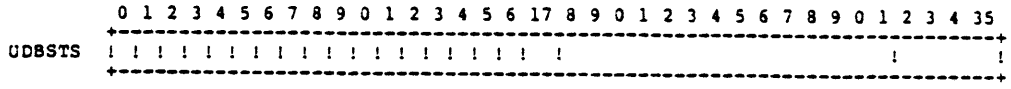
Description: Unit Data Block. This block, one per unit, contains information about the current activity on the unit.

Defined in: PHYPAR

Referenced by: PHYSIO

Format





Symbol	Bits	Pointer	Content
US.OFS	0		Offline or unsafe
US.CHB	1		Check home blocks before any normal I/O
US.POS	2		Positioning in progress
US.ACT	3		Active
US.BAT	4		Bad blocks on this unit
US.BLK	5		Lock bit for this units BAT blocks
US.PGM	6		Dual port switch in (A or B) (RP04,5,6)
US.MAI	7		Unit is in MAINT mode
US.MRQ	8		MAINT mode is requested on this unit
US.BOT	9		Unit is at BOT
US.REW	10		Unit is rewinding
US.WLK	11		Unit is write locked
US.MAL	12		MAINT mode allowed on this unit
US.OIR	13		Operator intervention required. Set at interrupt level, checked at SCHED.
US.OMS	14		Once a minute message to operator. Used in conjunction with US.OIR
US.PRQ	15		Positioning required on this unit
US.TAP	16		Tape type device
US.IDB	17		Tape - IDB seen on previous operation
	32-35	USTYP	Unit Type

Type Code for USTYP

Symbol	Code	Unit
.UTRP4	1	RP04
.UTRS4	2	RS04
.UTT16	3	TU16
.UTTM2	4	TM02 (as unit)
.UTRP5	5	RP05
.UTRP6	6	RP06

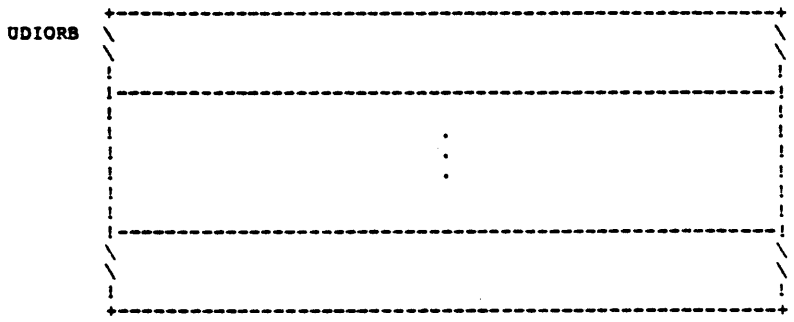
Name: UDIORB

Description: Pool for UDSKIO IORBs. The free IORBS are linked together and this list is pointed to by UIOLST.

Defined in: STG

Referenced by: PHYSIO

Format



Name: UDS

Description: Unit Dispatch Service Routine Table. This table, one per unit type, contains vectored addresses to unit dependent functions, and is given in its generalized form. The specific unit dispatch tables are RP4DSP (in PHYP4) for the disk device, and TM2DSP (in PHYM2) for the magtape device. See PHYPAR for definitions of arguments given and returned on calls to these unit routines.

Defined in: PHYPAR

Referenced by: PHYSIO, PHYH2, PHYM2(MTA), PHYP4(DSK), STG

Format

UDSINI	Initialize
UDSSIO	Start I/O on an IORB, skips if O.K.
UDSINT	Interrupt Routine (called on interrupts for XFER done)
UDSERR	Initiate Error Retry (skips if no more retrys)
UDSHNG	Hung Reset (called from TIMER to reset hung devices)
UDSCNV	Convert Unit Linear Address to CYL, SURF, SEC
UDSLTM	Return Latency or Best Request
UDSPOS	Start Position on IORB (skips if O.K.)
UDSATN	Attention Interrupt