

ASSIGN M:CI,(FILE,MM,DOñCI)

METASYM CI,L0,CN

•SS R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,R11,R12,R13,R14,R15

•SS SR1,SR2,SR3,SR4,D1,D2,D3,D4,§

•END

AJP8	1936/BL	1939/BNEZ	2023-RES				
AJP5	1952/BNE	1956-EQU	1960/B				
AJP6	1954/B	1962-EQU					
AJP9	2006-STD	2012/BIR					
BA	306/EQU	887/LI	887/LI	1039/LI	1039/LI	1060/LI	1060/LI
	1219/CI	1219/CI	1902/LI	2016/LI	2018/MBS	2018/MBS	2022/MBS
	2022/MBS	2067/LI	2067/LI	2091/CI	2091/CI		
BSCREECH	2316/BAL	2337/B	2358/B	2390/BE	2411/BAL	2438/BAL	2457-EQU
BUFFEXIT	2162-EQU	2198/BCR	2202/B				
BUFFXIT2	2165-PULL	2305/B					
BUFFXIT3	2273/BLZ	2293/B	2304-PULL				
CCS	319-EQU	979/LCF					
CORATX	690-EQU	691/LI					
DA	910/CLM	1108/CLM	1699/CLM				
DIC	1873/CI						
DOUBLEZERO	154/REF						
DV2	1040-CB	1047/B					
DV4	1041/BE	1048-STW					
EIDPA	262/REF	1631/LI					
EINC							

E:ND	260/REF	436/LI	481/LI	1977/LI	2054/LI	2176/LI	2
EIQMF	261/REF	843/LI	1916/LI				
ENBSR4	1022/REF	1023/LI					
FPMC	1303/REF	1317/BEZ	1326/BLZ	1328/B	1356/BLZ	1361/BGE	1363/B
	265/REF	570/LI	621/LI	998/LI			
GAJ6	2285/CI	2335/CI	2389/CI	2415/LI	2146/LI	2230/CI	2268/LI
GAJ8	1945/BE	1971-EQU					
GETCP00L	1967/LI	1982-EQU					
GETFP00L	2138-LCW						
GETSBUF	2137/BL	2158/B	2203-EQU				
GETSPARE	187/REF	2156/BAL					
GNV2	2146-LI	2214/B					
GNV3	422-EQU	1920/B					
GNV4	424-EQU	433/BDR					
GNV5	429/BCR	431-EQU	478/BE				
GNV7	439-EQU	484/B					
G0TCP00L	428/B	476-EQU					
G0TFP00L	2180-MTW						
G0TSPARE	2179/BL	2187-MTW					

	2148/BE	2168•EQU		
GPP1				
	1243/BEZ	1249•EQU	1290/B	
GVPA				
	954/BAL	963/BAZ	967/BNF	1761•EQU
GVPB				
	976•RES	1122/BLE	1745/LI	
GVPB				
	623/BNE	981•RES		
GVP10				
	940/BEZ	942/BL E	959/LI	974•FQU
GVP2				
	624•EQU	2171/BAL		
GVP23				
	689/BNEZ	696•EQU		
GVP230				
	700•EQU	714/BLE		
GVP232				
	704/BEZ	709•EQU		
GVP235				
	702/BNE	706/BCR	708/BCR	711•FQU
GVP24				
	744/BNE	747/BEZ	751/BGE	754•EQU
GVP240				
	756/BL	762•LH		
GVP241				
	764/BAZ	770•LB		
GVP242				
	771/BEZ	779•EQU		
GVP25				
	716/BL	718/BL E	721/BEZ	728•FQU
GVP250				
	768/BANZ	790•LB		
GVP255				
	776/BANZ	781/BCR	784/BANZ	792•EQU
GVP256				
	791/B	794•LB		

GVP26				
GVP27	627/LI	815=B		
GVP3	682/BE	734= EQU		
GVP4	833= EQU	845/B		
GVP45	841/BCR	846= EQU		
GVP5	829/BGEZ	868= EQU		
GVP51	874/BNE	882= EQU		
GVP52	870/BL	873= AI		
GVP53	871/BE	876= RES		
GVP6	884/BCR	886= LI		
GVP7	910= CLM	912/ BIR		
GVP75	911/BCR	915/ BLE	917/ BG	920= AI
GVP8	927= EQU			
GVP90	926/BE	931= EQU		
GZPRIV	733/ BLE	742/ BGE	745= LI	
HIGH	457/ REF	464/ CI		
I	575/ REF	576/ CW		
IA	472= D0	473/ STM		
IACU2	311/ IPSD			

IACU6	369/BGE	377-EQU					
IACU7	367/B	371/B	386/BDR	388/BAZ	390/BANZ	400-EQU	
INCSTL	405-LC	521/B					
INH	1301/DEF	1359-EQU					
IV10	311/PSD						
IV3	901-STORE	1216/BE	1226/B				
IV31	1162-EQU	1196/BL					
IV32	896/B	1173-LB					
IV4	1174/BNEZ	1177-RES					
IV6	1160-CLM	1186/BCR					
IV8	1161/BCR	1194-EQU					
IV8 ²	1176/B	1190/BCR	1197-EQU				
IV8 ⁵	1205/BCR	1227-CLM					
IV88	1208-CW	1228/BCR					
IV89	1203/BE	1207/BCR	1210-STW				
IV9	1209/BL	1211-LW	1229/B				
J	897-RES	1222/BNEZ					
JTABUF	306-EQU	687/LI	825/LI	834/LI	933/LI	1091/LI	1103/LI
	198/REF	2227/STW					

JIAJ	210/REF 2062/LW	850/LW 2109/LW	861/LW	890/LW	1034/LW	1938/LW	1988/STW
JIAJIT	211/REF 2018/MBS	852/LI 2022/MBS	892/LI 2061/LI	1037/LI	2006/STD	2009/STD	2016/LI
JIBUP	190/REF	568/CLM					
JICL	223/REF 2064/LI	849/LI	889/LI	1033/LI	2003/LM	2018/MBS	2022/MBS
JICLE	222/REF 1031/MTW	853/AW 1038/AW	854/LW 1098/MTW	865/MTW 1099/LW	893/AW 1910/AW	894/MTW 2014/LW	932/LW 2065/AW
JICLPA	224/REF	1990/STW					
JIDLL	203/REF	678/SW	707/CLM				
JIDUL	204/REF	677/AW					
JIEUP	202/REF	713/CW	916/CW	1114/CW	1705/CW	1864/LW	
JIJAC	219/REF 1879/STM	316/DATA	381/LW	952/LI	1689/LB	1717/STB	1749/LI
JIJIT	199/REF 2185/STB 2352/LB	306/EQU 2201/STB	693/LC 2206/LB	738/LB 2278/CB	749/LB 2287/STB	780/LC 2297/LB	2182/LB 2298/STB
JIPLL	200/REF 1185/CLM	676/SW 1189/CLM	705/CLM 1204/CLM	773/SW 1206/CLM	910/CLM 1227/CLM	1108/CLM 1699/CLM	1160/CLM
JIPUL	201/REF	675/AW	772/AW	914/CW	1112/CW	1703/CW	
JIRNST	153/REF	720/LS					
JITELFLGS							

JIVLCS	193/REF	783/CW					
JAJITVP	216/REF	1201/LW	1210/STW	2044/CW	2046/STW	2047/MTW	
JBICBUC	259/REF	1943/LI					
JBIFBUC	181/REF	2180/MTW	2276/MTW				
JBIFBUL	178/REF	2187/MTW	2289/MTW				
JBILMAP	177/REF	732/CB	735/LB	2133/LB	2213/LB	2222/LB	2226/STH
	2236/MTB	2274/CB	2434/LB				
JBINRG	220/REF	573/CB	579/STB	701/CB	887/LI	1002/LB	1010/MTB
	1039/LI	1040/CB	1042/LB	1051/LB	1052/STB	1059/STB	1172/LB
	1173/LB	1199/STB	1200/STB	1219/CI	2067/LI	2071/CB	2073/LB
	2089/LB	2091/CI	2101/LB				
JBIPCP	195/REF	1902/LI					
JBIPPC	205/REF	666/LB	669/LB	684/LB	699/LB	730/LB	921/MTB
	1120/MTB	1985/MTB					
JBIPPEAK	207/REF	818/LI					
JBIPPC	213/REF	1070/MTB	1218/MTB	1964/MTB	2043/MTB		
JBIPRIV	218/REF	463/LB					
JBIVLH	214/REF	887/LI	1039/LI	1219/CI	2067/LI	2091/CI	
JBIVLT	215/REF	1003/CB	1057/STB	1175/STB			
JBCBLL	183/REF	2181/LI	2277/LI				
JBCBUC	182/REF	748/LI					
JBFBFP							

JBFBUC	180/REF	2200/LI	2204/LI	2295/LI	2351/LI		
JBMNPA	179/REF	737/LI					
JBNASP	206/REF	687/LI					
JBNRG	209/REF	834/LI	1103/LI				
JBUPVP	228/REF	825/LI	933/LI	1091/LI			
JCCL	192/REF	697/LI	741/CI				
JCMAP	226/REF	1935/CI					
JC0VP	257/REF	1815/AI					
JC02VPA	174/REF	2136/CI	2178/CI	2264/LI	2408/LI		
JEUPVP	186/REF	312/DATA					
JHIDA	191/REF	1865/CI					
JHILDCF	225/REF	856/STH	1101/LH				
JJAC	175/REF	746/AND					
JJITVP	227/REF	1821/AI					
J0VVP	258/REF	964/LI	1949/LI				
JSBUF2VP	173/REF 1817/LI	313/DATA 1823/LI	368/CI 1824/LI	717/CI 2139/AI	941/CI 2143/LI	1121/CI 2282/CI	1816/LI 2313/LI
JX1CMAP	188/REF	2152/CI					
	221/REF 999/COMPARE	571/COMPARE 1011/STORE	578/STORE 1049/LOAD	622/COMPARE 1062/LOAD	901/STORE 1072/STORE	938/LOAD 1132/LI	1167/LOAD

	1947/STORE 2103/LOAD 2299/LOAD 2393/STORE	1950/LOAD 2147/COMPARE 2334/LOAD 2414/STORE	1973/STORE 2191/LOAD 2338/COMPARE 2416/STORE	1987/LOAD 2193/STORE 2359/LOAD 2441/LOAD	2082/STORE 2229/LOAD 2388/LOAD 2443/STORE	2090/LOAD 2269/STORE 2391/COMPARE	2095 ⁹ /LOAD 2284/LOAD
JX1PPH	212/REF 209 ³ /STORE	1060/LI	1223/LOAD	1225/STORE	1951/COMPARE	1953/STORE	1955/LOAD
JXBUFVP	184/REF 2260/AI	313/DATA 2385/AI	736/AI 2407/AI	2134/AI 2436/AI	2139/AI	2190/AI	2223/CI
LKIMG	347/DEF	348=RES					
MIADRINCR	256/REF	1999/AW					
MIDWT	1384=EQU	1448/LB	1458/CB				
MIFPPC	254/REF	1248/MTW	1289/MTW	1322/MTW	1351/MTW		
MIFPPH	252/REF	1242/LW	1245/STW	1284/STW	1316/LW	1319/STW	1347/STW
MIFPPT	253/REF	1247/STW	1282/LW	1288/STW	1321/STW	1345/LW	1350/STW
MIFREE#GRAN	168/REF	1531/MTW	1623/MTW	1914/CW			
MIGAM1	1378=EQU						
MIGAM2	1379=EQU	1513/LB					
MIGAM3	1380=EQU	1514/LB					
MIGAM4	1381=EQU	1519/LB					
MIGAM5	1382=EQU	1581/LB	1600/LB				
MIGAM6	1383=EQU	1589/LB	1611/LB				
MIGASLIM							

MIGATLIM	166/REF	1427/CW	1595/CW	1997/CS
MIGPT	167/REF	1584/CW		
MISBAND	1385=EQU			
MISGP	321/DEF	322=DATA		
MISWAPD	251/REF	1449/LW	1506/LW	1617/LW
MISWAPS	321/DEF	323=DATA		
MAP	1386=EQU	1445/LB	1613/LB	
MAPSPARE	229/REF	1846/BAL	2303/BAL	2364/BAL
MASKHAF	2184/BG	2186/B	2188=EQU	2210/B
MBIDWT	331=GEN	1499/CW		
MBIGAM1	162/REF	1384/EQU		
MBIGAM2	155/REF	1378/EQU		
MBIGAM3	156/REF	1379/EQU		
MBIGAM4	157/REF	1380/EQU		
MBIGAM5	158/REF	1381/EQU		
MBIGAM6	159/REF	1382/EQU		
MBIGAM7	160/REF	1383/EQU	1528/LB	1627/LB
MBIGPT	161/REF	1433/MTB	1525/CB	1592/CB
	163/REF	1385/EQU		1996/LB

MBISPT						
MBISWAPS	164/REF	1426/LB	1586/LB			
MINUS2	165/REF	1386/EQU				
MM	170/REF	1580/AND				
MMCC	275/DEF	308=EQU	2459/EQU			
MMCI	1140/B	1760=EQU				
MMSZ	1775/EXU	1799=EQU				
MMUMI	2459=EQU					
MN2	311=IPSD	2300/LPSD				
MXIPPUT	326=DATA	392/AND				
	255/REF	898/LOAD	899/STORE	900/STORE	1060/LI	1068/LOAD
	1069/STORE	1224/STORE	1244/LOAD	1280/STORE	1287/STORE	1318/LOAD
	1343/STORE	1349/STORE	1957/COMPARE	1959/LOAD	1961/STORE	1963/STORE
	2096/STORE	2105/STORE				
M13						
M2	189/REF	566/AND				
M22	325/REF	401/AND	1683/AND			
M5	1772/LW	1773/REF				
M8	171/REF	1601/AND				
NOBUF	325/REF	364/AND	567/AND			
NOBUF2	2151=EQU					
	2153/BG	2160=LI				

N0BUF3	2159=PULL	2172/B				
N0NEFREE	2207/BEZ	2211=EQU				
N00LAY	667/BEZ	680=EQU				
NPMC	303/DEF	304=EQU	879/LI	939/AI	1064/CI	1080/CI
	1215/CI	1972/LI	2042/LI	2087/CI		1168/CI
NREADY	341=EQU	880/AND	1975/AND	2052/AND		
P:AC	270/REF	1877/LD				
P:NAME	268/REF	787/CD				
P:SA	250/REF	726/CW	767/CW			
PAC4	1871/BANZ	1874/BAZ	1876=EQU			
PAGEZAP	302/DEF	463=LB				
PAGEZAPT	456/DEF	461=LI				
PAGEZAPO	456/DEF	458=LI				
PAGEZAP1	468=AI	475/B				
PBIPSZ	241/REF	790/LB	795/LB	797/CB	799/LB	
PGPZE	460/B	466=SLS				
RNAMEND	267/REF	785/LI				
POSITION	328=EQU	1501/AH				
PXAC	316=DATA	1880/LD				

RBUF2	2281=AI	2286/BE		
RBUF3	2283/BE	2287=STB		
RLSBUF	2266/B	2271=LW		
RLSBUF2	2275/BLE	2289=MTW		
RLSBUF3	2279/BNE	2288/B	2290=LW	
RLSP00L	2272/BGZ	2294=EGU		
RLSP2	311/IPSD	2301=SL S		
RNVP1	1081/BE	1083/BE	1087/BCR	1089=FGU
RPP2	1283/BNEZ	1285/B	1286=EGU	
RSPARE	2261=EGU			
RTCHK	240/REF	883/BAL	1086/BAL	
RUNDRLS	2235/BAL	2263=LI		
RVNP	1067/B	1071=EGU		
RVP2	1001=EGU			
RVP3	980=RES	1000/BE		
RVP42	1009/BNE	1013=LW		
RVP43	1014=LB	1028/B		
RVP44	1015/BEZ	1029=EGU		
RVP5				

RVP6	1095/BL	1106=EGU					
RVP7	1108=CLM	1110/BIR					
RVP8	1109/BCR	1113/BLE	1118=EGU				
RVP9	1128=LI	1639/LI					
RVP91	1012/B	1127/B	1638=LI				
RWINDOW	1128/LI	1745=LI					
S:ACORE	2259/BCR	2307=EGU					
S:CUN	1302/REF	1327/MTW	1357/MTW				
	242/REF	625/LW	857/LW	877/LW	903/LW	960/LW	1013/LW
	1074/LW	1416/LW	1569/LW	1777/LW	1867/LW	1912/LW	1940/LW
	2049/LW	2111/LW					
S:DP	169/REF	1421/MTW	1573/MTW	1624/MTW			
S:PCORE	238/REF	804/LW					
S:RTCORE	239/REF	885/MTW	1088/MTW				
S:SJACCW	343/DEF	345=DATA					
S:STL#	1302/REF	1307/LW	1323/MTW	1352/LW	1354/STW		
S:STLC	1302/REF	1310/CW	1358/LW	1362/MTW			
SACACC	338=DATA	1692/LB	1696/CB	1710/LB	1711/CB	1715/LB	
SACBP	337=DATA	1686/LB					
SAC2	1692=LB	1695/BIR					

SAC3	1700/BCR	1704/BLE	1706/BG	1708=AI	
SAC4	1690/BIR	1715=LB			
SAC4M	1697/BLE	1713=LW			
SAC6	1712/BG	1718=B			
SAD0B	565=EQU				
SAD0D	581/BE	584=EQU			
SAD0R	569/BCS	574/BNE	577/BG	596=EQU	
SAD1	583/B	590=BAL			
SAD1M1	589=AI				
SAD5	441/B	520=PULL			
SB:HQ	264/REF				
SB:RBMX	235/REF	692/LB			
SB:ROMX	236/REF	695/LB			
SBUF1VPA	185/REF	312/DATA			
SEARCH	2265/BAL	2315/BAL	2333=EQU	2410/BAL	2437/BAL
SEARCH2	2336/BNE	2338=COMPARE	2343/BDR		
SEARCH3	2339/BNE	2342=AI			
SGA1	1444=EQU				
SGA10					

SGA12	1422/BNEZ	1530=EQU		
SGA ²	1447=EQU	1465/BNE		
SGA ³	1454=EQU	1462/BDR	1478/B	
SGA ⁴	1459/BL	1461=EQU		
SGA ⁵	1456/BNE	1470=EQU		
SGA ⁶	1473/BANZ	1476/BANZ	1481=LW	
SGA ⁷	1492=EQU			
SGA ⁸	1479=STS	1485/BEZ		
SGA ⁹	1498=EQU	1503/BDR		
SGR ^{BAD}	1500/BCS	1502=EQU		
SG ^{R0}	1466/B	1585/BG	1596/BG	1620=LCI
SGR ^{0A}	1574/BNEZ	1619/BAZ	1622=EQU	
SGR ¹	1625/BNEZ	1629=EQU		
SIZERR	1532/B	1599/BANZ	1634=EQU	
SJAC	603=PULL	815/B		
SLICORE	962/CI	1779/CI	1830/CI	
SLIRSV ^P	237/REF	755/CW		
SLISTLM	1302/REF	1308/SW	1355/SW	
	1302/REF	1360/CW		

SNAC ₁					
	536=	EQU	543/BDR		
SNAC ₂					
	537/LI		541=RES		
SPDBASE					
	249/REF		317/GEN		
SPGV ₁					
	2049=LW		2098/CI	2099/BE	
SPJACTRL					
	346=GEN		969/LW	1784/LW	1834/LW
SPJITAC					
	342=DATA		345/DATA	968/LI	1783/LI
SPRLIMS					
	313=DATA				1833/LI
STL					
	1309/BLZ		1311/BL	1315=	EQU
SYSGEN					
	5=SET		320/D0		
S69PROC					
	1=SET				
T:FPF					
	278/DEF		1085/BAL	1264=	EQU
T:GAJP					
	937/BAL		1911/BAL	1933=	EQU
T:GBUF					
	281/DEF		2130=	EQU	
T:GNVNP ₁					
	299/DEF		1897=	PUSH	
T:GNVPI					
	298/DEF		419=	EQU	
T:GPP					
	277/DEF		872/BAL	1240=	EQU
T:GVGPI					
	287/DEF		514=	EQU	
T:GVPI					
	288/DEF		427/BAL	517/BAL	619=
T:IACU					
					2037/BAL

T:MBUF	297/DEF	359=EQU					
T:IPAC	283/DEF	2383=EQU					
T:RBUF	290/DEF	1863=EQU					
T:IRE	282/DEF	2254=EQU					
T:IREG	263/REF	1632/BAL					
	339/REF	437/BAL	482/BAL	844/BAL	1025/BAL	1917/BAL	1978/BAL
	2055/BAL	2177/BAL					
T:RSPP	1301/DEF	1338=EQU					
T:RSPPPEA	1301/DEF	1337=LW					
T:RVPI	289/DEF	500/B	996=EQU	2291/BAL			
T:RVSPI	300/DEF	496=EQU					
T:SAC	296/DEF	590/BAL	929/BAL	1680=EQU			
T:SACP1	292/DEF	539/BL	1682=RES				
T:SADO	276/DEF	561=EQU					
T:SAD1	305/DEF	564=EQU					
T:SBUF	286/DEF	2405=EQU					
T:SGA	840/BAL	1415=EQU					
T:SGAJIT	295/DEF	1419=EQU					
T:SGR	279/DEF	1102/BAL	1568=EQU				
T:SGRNU							

	1571-EQU	1572/DEF					
T:SMC							
	1762-EQU						
T:SNAC							
	301/DEF	535-EQU					
T:SPGVI							
	2036/DEF	2037-BAL					
T:STLPP							
	1301/DEF	1305-EQU					
T:SXAC							
	1757-EQU	1761/EQU	1825/BAL				
T:SXACM3							
	291/DEF	592/BAL	1749-LI				
T:SXMAP							
	1138-LI	1818/BAL					
T:SXMAPM2							
	1132-LI	1968/B					
T:SXMAPM3							
	293/DEF	594/BAL	947/BAL	1131-LW	2195/BAL	2270/BAL	2394/BAL
	2417/BAL	2419/BAL	2447/BAL				
T:TOTESZ							
	280/DEF	664-EQU					
T:XBUFF							
	285/DEF	2431-PUSH					
T:XMMC							
	294/DEF	1807-EQU	1845/BAL				
T:XMMC1							
	1842/DEF	1843-PUSH					
T:XPVVI							
	2039/BNE	2048/BAL	2061-LI	2445/BAL			
T:ZBUF							
	284/DEF	2220-EQU					
T:ZPUP							
	374-PUSH						
TEL							
	314-TEXTC	786/LD					
TESTREL							

TESTREL3	2262/BAL	2318/BAL	2350-EQU			
TFPP1	2355-AI	2367/BNE				
TIC	1277-SCREECH	1340/BLE				
T0PR00T	763/CI	775/CI	1870/CI			
TSTACK	176/REF	1265/LI	1339/CI			
UACT	266/REF	440/STB*	626/PSW	822/PLW		
UBIACP	309-DATA	928/LB	1709/LB			
UBIAPR	232/REF	765/LB	1869/LB			
UBIASP	233/REF	722/LB	770/LB			
UBIDB	231/REF	724/LB	794/LB	1872/LB		
UBIMF	269/REF	796/LB	1875/LB			
UBIPCT	246/REF	1014/LB				
UBISWAPI	243/REF	665/LB	904/MTR	1075/MTB	1941/MTB	
UFLAGS	172/REF	1417/LB	1570/LB	1913/LB	1994/LB	
UHIAJIT	3-SET					
UHIFLG	248/REF	1993/STH				
UHIFLG2	245/REF	762/LH	878/LH	881/STH	905/LH	907/STH
	1076/LH	1078/STH	1778/LH	1822/LH	1868/LH	1974/LH
	1980/STH	2050/LH	2053/STH			961/LH
						1976/STH
	1016/REF	1017/LH	1019/STH			

UHIJIT	247/REF	1992/LH	2000/STH	
UNMAP	230/REF	1844/BAL	2361/BAL	
UNMAPBUF	2267=EQU	2312/B	2319/B	
UTSPROC	2=SET			
UXIJIT	244/REF	858/LOAD	1811/LOAD	2112/LOAD
WA	910/CLM	1108/CLM	1699/CLM	
WK	311/IPSD			
WNDLIMS	312=DATA	2258/CLM		
XA	272/REF	855/DW	1100/DW	
XFFF800	271/REF	863/LW		
XMMC5	1831/BAZ	1837=EQU		
XN2	327=DATA	341/EQU		
XPGV1	2068=AI	2074/BNEZ		
XPGV2	2072/BE	2082=STORE		
XPGV3	2104=EQU	2110/BNEZ	2113/B	
XPGV4	2102/BEZ	2108=EQU		
XPPSWAP	340=EQU	906/BR	1077/BR	2051/BR
X1	325/REF	1979/BR		
X10				

	325/REF	340/EQU	380/DW	
X4	197/REF	1907/DW		
X8	1018/BR	1020/REF		
YFF	318/REF	319/EQU	2004/AND	2021/AW
Y002	194/REF	782/LW		
Y003E	208/REF	719/LW		
Y08	325/REF	725/LW		
Y4	234/REF	766/LW		
Y8	325/REF	2208/LW		
ZBUF2	2222=LB	2237/B		
ZBUF3	2224/BGE	2229=LOAD		
ZBUF4	2231/BE	2232/LI	2236=MTR	
1A1	1268/BL	1278=EQU		
7H3	499/LI	518=RES	1082/CI	
1BIG	273/REF	859/LI	1061/SLS	1139/LI

H01 17145 SEP 08, 1975
1 00000001
2 00000001
3 00000001
4
5 00000001

S69PRBC SET 1
UTSPRBC SET 1
UFLAGS SET 1
SYSTEM UTS
SYSGEN SET 1

6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42

PAGE
MM
NAME: MM

PURPOSE: TO PERFORM ALLOCATION OF VIRTUAL AND PHYSICAL MEMORY AND SWAP STORAGE. THIS INCLUDES SETTING UP THE USERS VIRTUAL MEMORY MAP AND ACCESS PROTECTION TABLE IN THE JIT AND LOADING THEIR RESPECTIVE HARDWARE REGISTERS.

DESCRIPTION: WHEN A VIRTUAL PAGE IS TO BE ALLOCATED A PHYSICAL PAGE AND SWAP GRANULE ARE OBTAINED AND AN I/O COMMAND ENTRY BUILT IN THE USER SWAP COMMAND LIST IN JIT/AJIT. ALLOCATED VIRTUAL PAGES ARE LINKED IN JB:LMAP IN THE SAME ORDER AS THEIR COMMAND LIST ENTRIES. THE PHYSICAL PAGE NUMBER ASSIGNED TO THE VIRTUAL PAGE IS PLACED IN JX:ICMAP.

SWAP GRANULES ARE ALLOCATED IN GROUPS OF FOUR, ON RAD SWAPPING SYSTEMS, AND THE SEEK ADDRESS OF THE FIRST SECTOR OF THE GROUP IS PLACED IN THE DISC ADDRESS TABLE, JH:IDA. A SEEK COMMAND, ADDRESSING THE DISC ADDRESS TABLE ENTRY, IS BUILT IN THE COMMAND LIST PRECEDING EACH GROUP OF FOUR I/O COMMANDS.

UNALLOCATED VIRTUAL PAGES ARE MAPPED INTO A MONITOR PAGE (KNOWN AS THE FPMC PAGE), WHICH IS WRITE LOCKED AND HAS AN ACCESS PROTECTION CODE OF 11. ALLOCATED PAGES HAVE THEIR ACCESS PROTECTION CODES STORED IN J:JAC BASED ON THEIR TYPE OF USAGE.

OTHER JIT TABLES DESCRIBE VIRTUAL MEMORY BOUNDARIES FOR EACH TYPE OF MEMORY, (DATA DCB, PROCEDURE, ETC.) AND THE NUMBER OF PAGES

H01 17:45 SEP 08, 195
43

P

ALLOCATED FOR EACH TYPE.

25

44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73

PAGE	
JIJIT	USERS TEMP STACK
*	:
*	:
*	:
*	:
*	:
*	BATCH JIT
*	:
*	:
*	JIBUP
*	JIEUP
*	:
*	JIPLL
*	JIPUL
*	JIDLL
*	JIDUL
*	JIDDLL
*	JIDDUL
*	JICLL
*	JICUL
*	JB:PCP JB:PCD JB:PCDD JB:PC
*	JB:TDP JB:BCP JB:MNPA JB:NASP

74
 75
 76
 77
 78
 79
 80
 81
 82
 83
 84
 85
 86
 87
 88
 89
 90
 91
 92
 93
 94
 95
 96
 97
 98
 99
 100
 101
 102

*
 *
 *
 *
 *
 *
 *
 *
 *
 *
 *
 *
 *
 *
 *
 *
 *
 *
 *
 *
 *
 *
 *
 *
 *
 *

PAGE

JB:PPH
JB:PPT
JB:PPC
JB:VLH
JB:VLT
J:VLCS
J:AJ
J:CLPA
J:CLE
J:CLP
J:CLS
J:JAC

HO1 17:45 SEP 08, 1975

130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151

*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*

PAGE

J:CL

:
:
:

	PAGE		
152			
153	REF	J:RNST	CHECK IF MONITOR RUNNING
154	REF	DOUBLEZERO	CONSTANT
155	REF	MBIGAM1	GRANULE ALLOCATION
156	REF	MBIGAM2	GRANULE ALLOCATION
157	REF	MBIGAM3	GRANULE ALLOCATION
158	REF	MBIGAM4	GRANULE ALLOCATION
159	REF	MBIGAM5	GRANULE ALLOCATION
160	REF	MBIGAM6	GRANULE ALLOCATION
161	REF	MBIGAM7	GRANULE ALLOCATION
162	REF	MBIDWT	GRANULE ALLOCATION
163	REF	MBIGPT	GRANULE ALLOCATION
164	REF	MBISPT	GRANULE ALLOCATION
165	REF	MBISWAPS	GRANULE ALLOCATION
166	REF	M:GASLIM	GRANULE ALLOCATION
167	REF	MIGATLIM	GRANULE ALLOCATION
168	REF	MIFREE#GRAN	GRANULE ALLOCATION
169	REF	S:DP	CHECK FOR PACK SWAPPING
170	REF	MINUS?	CONSTANT
171	REF	M5	CONSTANT
172	REF	UB:SWAP1	SWAP INDEX NR.
173	REF	JOVVP	OVERLAY VIRT PAGE NR.
174	REF	JCOVP	CONTEXT VIRT PAGE NR.
175	REF	JHILDCF	CHECK IF SYMB ACCESS OK
176	REF	TOPROOT	MONITOR TOP
177	REF	JBIFBUL	FILE BUFFER UPPER LIMIT
178	REF	JBIFBUC	FILE BUFFER USE COUNT
179	REF	JBFBUC	FILE BUFFER USE COUNT
180	REF	JBFBFP	FREE FILE BUFFER HEAD
181	REF	JB:CBUC	COOP BUFFER USE COUNT
182	REF	JBCBUC	COOP BUFFER USE COUNT
183	REF	JBCBLI	COOP BUFFER LOWER LIMIT
184	REF	JXBUFVP	SPARE BUFFER VIRT PAGE NR.
185	REF	SBUF1VPA	SPEC WINDOW BUFFER #1
186	REF	JCO2VPA	2ND. COOP BUFFER PAGE ADDR
187	REF	GETSBUF	FILE MANG. TRUNCATION
188	REF	JSBUF2VP	SPEC WINDOW BUFFER #2

189	REF	M13	CONSTANT
190	REF	J:BUSP	BEGIN USER PAGE
191	REF	JEUPVP	END USER PAGE VIRT PAGE
192	REF	JBUPVP	BEGIN USER PAGE VIRT PAGE
193	REF	J:TELFLGS	EXTENDED MEMORY MODE BIT
194	REF	Y002	CONSTANT
195	REF	JBINRG	NR. GRANULES REMAINING
196			IN GROUP OF FOUR
197			CONSTANT
198	REF	X4	RESET TO ZERO
199	REF	J:ABUF	JOB INFORMATION TABLE
200	REF	J:JIT	PROCEDURE LOWER LIMIT
201	REF	J:PLL	PROCEDURE UPPER LIMIT
202	REF	J:IPUL	END USER PAGE
203	REF	J:IEUP	DATA LOWER LIMIT
204	REF	J:DLL	DATA UPPER LIMIT
205	REF	J:IDUL	PROCEDURE PAGE COUNT
206	REF	JB:PCP	MAX NR. PAGES ALLOWED
207	REF, 1	JB:PEAK	MAX PAGES USED
208	REF	Y003E	CONSTANT
209	REF	JB:NASP	NEXT GRAN POSITION ON RAD
210	REF	J:IAJ	AJIT PHYS PAGE NR.
211	REF	J:IAJIT	VIRT ADDRESS OF AJIT
212	REF	JX:IPPH	USERS PHYSICAL PAGE CHAIN HD
213	REF	JB:PPC	USERS PAGE COUNT
214	REF	JB:IVLH	VIRT PAGE CHAIN HD
215	REF	JB:IVLT	VIRT PAGE CHAIN TAIL
216	REF	J:IVLCS	VIRT PAGE WHERE MEMORY
217			AND RAD NO LONGER COINCIDE.
218	REF	JB:PRIV	USERS PRIVILEGE
219	REF	J:JAC	ACCESS PROTECTION TABLE
220	REF	JB:IMAP	VIRTUAL PAGE TABLE
221	REF	JX:CMAP	PHYSICAL PAGE TABLE
222	REF	J:ICL	CLIST LENGTH
223	REF	J:ICL	COMMAND LIST IN JIT
224	REF	J:ICLPA	CLIST PHYSICAL ADDRESS
225	REF	JH:IDA	DISC ADDRESS TABLE

226	REF	JCCL	MAX CLIST LENGHT IN JIT
227	REF	JJAC	ACCESS PROTECTION CODE TABLE
228	REF	JBNRG	NR. REMAINING GRANULES
229	REF	MAP	ROUTINE TO MAP USER
230	REF	UNMAP	ROUTINE TO UNMAP USER
231	REF	UBI:ASP	USERS SPECIAL PROCESSOR
232	REF	UBI:ACP	USERS COMMAND PROCESSOR
233	REF	UBI:APR	USERS PROCESSOR
234	REF	Y4	CONSTANT
235	REF	SB:RBMX	CORE LIMIT
236	REF	SB:ROMX	CORE LIMIT
237	REF	SLICORE	CHECK USER SIZE
238	REF	S:PCORE	CHECK USER SIZE
239	REF	S:RTCORE	COUNT REAL TIME PAGES
240	REF	RTCHK	CHECK FOR REAL TIME LOCKED IN
241	REF	PBIPSZ	PROCESSOR PROCEDURE SIZE
242	REF	S:CUN	CURRENT USER NUMBER
243	REF	UBIPCT	USERS PAGE COUNT
244	REF	UX:JIT	USERS JIT PAGE NR.
245	REF	UHIFLG	USERS FLAGS
246	REF	UBIMF	USERS MASTER FUNCTION COUNT
247	REF	UH:JIT	JIT DISC ADDRESS
248	REF	UH:AJIT	AJIT DISC ADDRESS
249	REF	SPDBASE	SPEC PROCESSOR DATA ADDRESS
250	REF	P:SA	PROCESSOR START ADDRESS
251	REF	MISGR	SWAP GRANULE TABLE ADDRESS
252	REF	M:FPFH	FREE PAGE CHAIN HEAD
253	REF	M:FPPT	FREE PAGE CHAIN TAIL
254	REF	M:FPFC	FREE PAGE CHAIN COUNT
255	REF	MX:IPPUT	RESIDENT PHYSICAL PAGE TABLE
256	REF	MIADRINCR	INC FROM END OF TRACK TO NEXT
257	REF	JCMAP	PHYSICAL PAGE TABLE IN JIT
258	REF	JJITVP	JIT VIRT PAGE NR.
259	REF	JAJITVP	AJIT VIRT PAGE NR.
260	REF	E:INC	NO CORE EVENT
261	REF	E:IND	NO DISC EVENT
262	REF	E:IDPA	DISC AVAILABLE EVENT

263	REF	T:RE	REPORT EVENTSS
264	REF	SB:HQ	HEAD OF STATE QUEUES
265	REF	FPMC	FREE PAGE MAP CONSTANT
266	REF	TSTACK	TEMP STACK
267	REF	PNAMEND	PROCESSOR NAME TBL END
268	REF	P:NAME	TABLE OF PROCESSOR NAMES
269	REF	UB:DB	USERS DEBUGGER
270	REF	P:AC	PROCESSOR ACCES CODES
271	REF	XFFF800	CONSTANT
272	REF	XA	CONSTANT
273	REF	:BIG	BIG MEMORY FLAG
274			
275	DEF	MM	DEF FOR PATCHING
276	DEF	T:ISAD0	MAP VIRT TO PHYS
277	DEF	T:GPP	GET PHYS PAGE
278	DEF	T:FPP	FREE PHYS PAGE
279	DEF	T:SGR	RELEASE SWAP GRANULE
280	DEF	T:TOTESZ	CALCULATE USERS SIZE
281	DEF	T:GBUF	GET BUFFER PAGE
282	DEF	T:RBUF	RELEASE BUFFER PAGE
283	DEF	T:MBUF	MAP BUFFER PAGE
284	DEF	T:ZBUF	RESET BUFFERS
285	DEF	T:XBUF	EXCHANGE BUFFERS
286	DEF	T:SBUF	SWITCH BUFFERS
287	DEF	T:GVGPI	GET VIRT GIVEN PHYS PAGE
288	DEF	T:GVPI	GET VIRT AND PHYS PAGE
289	DEF	T:RVPI	RELEASE VIRT AND PHYS PAGE
290	DEF	T:PAC	LOAD PROCESSOR ACCESS CODE
291	DEF	T:ISXACM3	LOAD ACCES PROTECTION
292	DEF	T:ISACP1	SET ACCESS CODE IN JAC
293	DEF	T:ISXMAPM3	LOAD MAP REGISTERS
294	DEF	T:IXMMC	SETUP MAP AND ACCESS REGISTERS
295	DEF	T:ISGALIT	JIT GRANULE ALLOCATION
296	DEF	T:ISAC	SET ACCESS CODE IN JAC
297	DEF	T:IIACU	OBTAIN ACCESS CODE
298	DEF	T:IGNVPI	GET N VIRT AND PHYS PAGES
299	DEF	T:IGNVNPI	GET N VIRT, NB PHYS, PAGES

H01 17:45 SEP 08, 175

300
301
302
303
304
305
306

00000022

EXT

NPMC

J

DEF TIRVSP1
DEF TISNAC
DEF PAGEZAP
DEF NPMC
EQU X1221
DEF TISAD1
EQU BA(JI,JIT)

RELEASE VIRT, NOT PHYS, PAGES
SET ACCESS ON N PAGES
INITIALZE PAGES
UNALLOCATED PAGE MAP CONSTANT
CVM ENTRY FOR SAD

Line	Code	Value	Unit	Page	Page	Page
307				PAGE		
308		01 00000	MM	EQU	8	
309	01	00000 01	UACT	DATA, 1	1,0,0,3,2	PRBG,DATA,DYN DATA,CONTEXT,DCBS
	01	00000 1 00				
	01	00000 2 00				
	01	00000 3 03				
	01	00001 02				
310				BUND	8	
311	01	00002 00004B7 N	MMUMI	IPSD	(IA,RI,SP2),(WK,1),INH	
		17000000				
312	01	00004 00000000 N	WDLIMS	DATA	SBUF1VP,JC02VP	
	01	00005 00000000 N				
313	01	00006 00000000 N	SPRLIMS	DATA	JXBUFVP,J0VVP=1	
	01	00007 FFFFFFFF N				
314	01	00008 03E3C5D3 A	TEL	TEXTC	'TEL'	
315	01	00009 40404040 A		TEXT	' '	
316	01	0000A 0000000E N	PXAC	DATA	J:JAC+14	
317	01	0000B 02000000 N		GEN,8,24	2,SPDBASE	
318				REF	YFF	
319		EXT	CCS	EQU	YFF	
320		00000000		DB	1-SYSGEN	
321			*S*	DEF	MISBAND,MISWAPD	
322			*S* MISBAND	DATA	X'1000'	
323			*S* MISWAPD	DATA	X'1F0'	
324				FIN		
325				REF	X1,X10,Y8,Y08,M8,M2	
326	01	0000C FFFFFFFC A	MN2	DATA	X'FFFFFFFC'	
327	01	0000D FFFFFFFE A	XN2	DATA	=2	
328		01 0000E	POSITION	EQU	8	
329	01	0000E 00000001 A		GEN,16,16	0,1	
330	01	0000F 00020004 A		GEN,16,16	2,4	
331	01	00010 00080010 A	MASKHAF	GEN,16,16	8,16	
332	01	00011 55555555 A		DATA	X'55555555'	
333	01	00012 33333333 A		DATA	X'33333333'	
334	01	00013 0F0F0F0F A		DATA	X'0F0F0F0F'	
335	01	00014 00FF00FF A		DATA	X'00FF00FF'	
336	01	00015 0000FFFF A		DATA	X'0000FFFF'	

HO1

17145 SEP 08, '75

337	01 00016	00300C03 A	SACBP	DATA	X'0300C03'
338	01 00017	0055AAFF A	SACACC	DATA	X'0055AAFF'
339				REF	T:REG
340	EXT		XPPSWAP	EQU	X10
341	01 0000D		NREADY	EQU	XN2
342	01 00018	FFF30FFF A	SPJITAC	DATA	X'FFF30FFF'
343				DEF	SISJACW
344				BOUND	8
345	01 0001A	00000018	SISJACW	DATA	SPJITAC
346	01 0001B	01008000 A	SPJACTRL	GEN,8,7,8,9 1,0,X'40',0	
347				DEF	LKIMG
348	01 0001C		LKIMG	RES	8

WI IMAGE FOR XDLT

```

349
350
351
352
353
354
355
356
357
358
359      01 00024
360 01 00024      02200080 A
      01 00025      08000000 N
      00000000
361
362
363
364 01 00026      48700000 X
365 01 00027      6930002A
366 01 00028      22600002 A
367 01 00029      68000035
368 01 0002A      21700000 N
369 01 0002B      6810002E
370 01 0002C      22600003 A
371 01 0002D      68000035
372
373      00000000
374
375
376
377      01 0002E
378 01 0002E      22200000 A
379 01 0002F      32300007 A
380 01 00030      36200000 X
381 01 00031      32760000 X
382 01 00032      22600000 A
383 01 00033      25200001 A
384 01 00034      25640102 A
    
```

```

PAGE
*F*
*F* NAME: T:IACU
*F* PURPOSE: TO OBTAIN THE ACCESS PROTECTION CODE FOR A GIVEN
*F* VIRTUAL PAGE.
*F* DESCRIPTION: RETURNS ACCESS = 2 FOR PAGE ZERO, ACCESS
*F* = 3 FOR PAGES BELOW THE FIRST OVERLAY PAGE (JOVVP)
*F* AND THE VALUE FROM JIJAC FOR OTHERS.
*
* 7 = VP #
* CC3&4 = AC
T:IACU EQU $
      PUSH 8,0
      DB 0 T:ZPUP
      LI,5 2
      FIN T:ZpUp
      AND,7 M8
      BNEZ $+3 NOT PAGE ZERO
      LI,6 2 AC=2 FOR PAGE
      B IACU6 ZERO
      CI,7 JOVVP
      BGE IACU2
      LI,6 3
      B IACU6
* UNUSED ROUTINE T:ZPUP REMOVED (17 WORDS RJR).
      DB 0 T:ZPUP
      PUSH 8,0
      LI,5 0
      FIN T:ZPUP
      EQU $ IACU2
      LI,2 0
      LW,3 7
      DW,2 X10 2= DB BIT POS 3= WD OF IMAGE
      LW,7 JIJAC,3 WD FRM AC IMAGE
      LI,6 0
      SLS,2 1 DB BIT POS TO BIT POS
      SLD,6 2,2 APPR8 AC LOW 2 BITS OF 6
    
```

H01 17:45 SEP 08, 175
00000000

38

T:ZPUP

385				DR	0
386			*S*	BDR,5	IACU6
387			*S*	CI,6	1
388			*S*	BAZ	IACU6
389			*S*	CI,6	2
390			*S*	BANZ	IACU6
391			*S*	LD,0	6
392			*S*	AND,0	MN2
393			*S*	LCW,2	2
394			*S*	SLD,0	*2,2
395			*S*	LI,2	1
396			*S*	AI,3	X18001
397			*S*	SLS,3	13
398			*S*	LPC,2	0

ZERO AC

MOVE WD BACK TO ORIG & APPRO AC TO
IMAGE ADR
CNT OF 1 WHEN SHIFTED

T:ZPUP

400		01 00035		IACU6	FIN	
401	01	00035	4B600000 X		EQU	*
402	01	00036	25600004 A		AND,6	M2
403	01	00037	75600008 A		SLS,6	4
404	01	00038	02200080 A		STB,6	11
					PULL	8,0
405	01	0003A	0A000000 N			
406	01	0003B	70200008 A	IACU7	LC	11
			E8000008 A		B	*11

GET APPRO AC
TO 3&4TH BITS OF BYTE

```

407
408
409
410
411
412
413
414
415
416
417
418
419      01 0003C
420      01 0003C      02200070 A
          01 0003D      0B500000 N
421      01 0003E      22500000 A
422          01 0003F
423      01 0003F      22AFFFFF A
424          01 00040
425      01 00040      22400000 A
426
427      01 00041      6AB000BD
428      01 00042      6800007D
429      01 00043      68F00045
430      01 00044      3550000A A
431          01 00045
432      01 00045      20700001 A
433      01 00046      64600040
434      01 00047      21AFFFFF A
435      01 00048      6830004B
436      01 00049      22600000 N
437      01 0004A      6AB00000 X
438      01 0004B      22A00000 A
439          01 0004C
440      01 0004C      F5A00000 X
441      01 0004D      6800008F
    
```

```

PAGE
NAME:   TIGNVPI
PURPOSE: TO ALLOCATE A GIVEN NUMBER OF PAGES.
DESCRIPTION: ALLOCATES N VIRTUAL PAGES AND ASSIGNS
              PHYSICAL PAGES TO THEM BY CALLING TIGVPI.
              THIS ROUTINE IS CALLED BY STEP, LNKTRC AND
              SEGLD.
* I     6 = # OF PGS
* I     7 = 1ST VP #
*       5 = *1 = INDICATION TO GVP TO GET NO PP
*       * 0 = INDICATION TO GVP TO GET PP
TIGNVPI EQU $
        PUSH 7,5
        LI,5 0
        GNV2 EQU $
          LI,10 *1
        GNV3 EQU $
          LI,4 0
        *
        BAL,11 TIGVPI
        B      GNV7
        BCR,15 GNV4
        STW,5 10
        GNV4 EQU $
          AI,7 1
          BDR,6 GNV3
          CI,10 *1
          BE   $+3
          LI,6 EINC
          BAL,11 TIREG
          LI,10 0
        GNV5 EQU $
          STB,10 *TSTACK
          B      SAD5
    
```

* IS NOT AFFECTED BY FRMC TEST BUT IS AFFECTED BY MAX PGS TESTS

REQUESTED TOO MANY PGS GOT PG

INDICATES ALL OK

DONT CLEAN FOR GENERAL CASE


```

442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458 01 0004E 22A00000 A
459 01 0004F 22600001 A
460 01 00050 48000056
461 01 00051 22AFFFFFF A
462 01 00052 207FFFFFF A
463 01 00053 72B00000 X
464 01 00054 21B00000 N
465 01 00055 484A0000 A
466 01 00056 25700009 A
467 01 00057 25A00009 A
468 01 00058 206FFFFFF A
469 01 00059 691A0000 A
470 01 0005A 3870000A A
471 01 0005B 02200000 A
472 01 0005C 00000020 I
473 01 0005C 2B7E0000 A
474
01 0005D 2B7E0010 A
01 0005E 2B7E0020 A
01 0005F 2B7E0030 A
01 00060 2B7E0040 A
    
```

```

PAGE
NAME: PAGEZAP, PAGEZAPT, PAGEZAP0
PURPOSE: TO INITIALIZE PAGES ALLOCATED BY MM ROUTINES.
DESCRIPTION: EACH PAGE IS INITIALIZED WITH THE CONTENTS
OF THE CURRENT REGISTER SET.
PAGEZAP0 INITIALIZES ONE PAGE.
PAGEZAPT INITIALIZES N PAGES IN DESCENDING
ORDER
PAGEZAP INITIALIZES N PAGES IN EITHER ASCENDING
OR DESCENDING ORDER AS SPECIFIED.
*
7=PAGE BEFORE FIRST TO CLEAN
10=0,1,-1 DIRECTION TO CLEAN
6=#PAGES TO CLEAN
*
    
```

```

DEF PAGEZAPT, PAGEZAP0
REF GZPRIV
PAGEZAP0 LI,10 0 CLEAN ONE PAGE
LI,6 1
B PGPZE
PAGEZAPT LI,10 =1 ALL PAGES
AI,7 =1 R7 BELOW 1ST PAGE
PAGEZAP LB,11 JBIPRIV PAGEZAP0 BYPASSES
CI,11 GZPRIV IS HE SECURE
BAZ 0,5
PGPZE SLS,7 9
SLS,10 9
PAGEZAP1 AI,6 =1
BLZ 0,5
SW,7 10
LCI 0
DB 32
STM,7 I*16=16,7
FIN
    
```

H01 17:45 SEP 08, 1975

	01	00061	287E0050	A
	01	00062	287E0060	A
	01	00063	287E0070	A
	01	00064	287E0080	A
	01	00065	287E0090	A
	01	00066	287E00A0	A
	01	00067	287E00B0	A
	01	00068	287E00C0	A
	01	00069	287E00D0	A
	01	0006A	287E00E0	A
	01	0006B	287E00F0	A
	01	0006C	287E0100	A
	01	0006D	287E0110	A
	01	0006E	287E0120	A
	01	0006F	287E0130	A
	01	00070	287E0140	A
	01	00071	287E0150	A
	01	00072	287E0160	A
	01	00073	287E0170	A
	01	00074	287E0180	A
	01	00075	287E0190	A
	01	00076	287E01A0	A
	01	00077	287E01B0	A
	01	00078	287E01C0	A
	01	00079	287E01D0	A
	01	0007A	287E01E0	A
	01	0007B	287E01F0	A
475	01	0007C	68000058	
476		01 0007D		
477	01	0007D	20400000	A
478	01	0007E	68300045	
479	01	0007F	21AFFFFFF	A
480	01	00080	68300083	
481	01	00081	22600000	N
482	01	00082	6AB00000	X
483	01	00083	22A000FF	A
484	01	00084	6800004C	

GNV7

B	PAGEZAP1
EQU	\$
AI,4	0
BE	GNV4
CI,10	=1
BE	\$+3
LI,6	EINC
BAL,11	T:REG
LI,10	X'FF'
B	GNV5

INDICATION OF ERROR

```

485
486
487
488
489
490
491
492
493
494
495
496      01 00085
497      01 00085      02200070 A
          01 00086      0B500000 N
498      01 00087      225FFFFFF A
499      01 00088      22B0008E
500      01 00089      680001B2
    
```

```

PAGE
NAME: TIRVSP1
PURPOSE: TO RELEASE A VIRTUAL PAGE WITHOUT RELEASING
        ITS ASSIGNED PHYSICAL PAGE.
DESCRIPTION: THIS ROUTINE CALLS T:RVPI TO RELEASE THE
        VIRTUAL PAGE, COMMAND LIST ENTRY AND SWAP
        GRANULE. IT SETS A FLAG TO INDICATE THAT
        T:RVPI IS NOT TO RETURN THE PHYSICAL PAGE TO
        THE MONITOR FREE PAGE LIST. THE FPMC CONSTANT
        IS PLACED IN JX:CMAP ENTRY. THE CALLER MUST
        KEEP TRACK OF THE PHYSICAL PAGE.

TIRVSP1 EQU *
        PUSH 7,5

        LI,5 *1
        LI,11 7H3
        B TIRVPI
    
```

```

501
502
503
504
505
506
507
508
509
510
511
512
513
514      01 0008A
515      01 0008A      02200070 A
          01 0008B      0B500000 N
516      01 0008C      225FFFFE A
517      01 0008D      6AB000BD
518      01 0008E
519      01 0008E      02000000 A
520      01 0008F      02200070 A
          01 00090      0A500000 N
521      01 00091      6800003A
    
```

```

PAGE
NAME:      T;GVGPI
PURPOSE:   TO ALLOCATE A VIRTUAL PAGE AND MAP IT INTO
           A GIVEN PHYSICAL PAGE.
DESCRIPTION: THIS ROUTINE CALLS T;GVPI TO ALLOCATE
           THE VIRTUAL PAGE, SWAPPER GRANULE AND COMMAND
           LIST ENTRY. IT SETS A FLAG TO INDICATE
           TO T;GVPI TO USE THE SPECIFIED PHYSICAL
           PAGE INSTEAD OF GETTING ONE FROM THE
           MONITOR FREE PAGE LIST.

*
*       7 = VP
*       3 = PP
*       5 = *2 = INDICATION TO GVP TO USE GIVEN PP
T;GVGPI EQU $
          PUSH      7,5

          LI,5      *2
          BAL,11    T;GVPI
          RES       0
          NBP
          SADS     PULL      7,5

          B        IACU7
    
```

```

522
523
524
525
526
527
528
529
530
531
532
533
534
535      01 00092
536      01 00092
537 01 00092 22280097 N
538 01 00093 21700100 A
539 01 00094 69100338
540 01 00095 0F000000 X
      01 00096 00210100 A
541 01 00097
542 01 00097 20700001 A
543 01 00098 64600092
544 01 00099 F800000B A
545

```

```

PAGE
NAME: T:SNAC
PURPOSE: TO SET ACCESS PROTECTION ON N VIRTUAL PAGES.
DESCRIPTION: CALLS T:SNAC N TIMES.
SCREECH CODE: 21 SUA
REPORTED BY: MM
MESSAGE: ATTEMPT TO SET ACCESS CONTROLS ON NON-
EXISTENT VIRTUAL PAGE.
REGISTERS: 6=NUMBER OF PAGES TO SET
           7=VIRTUAL PAGE NUMBER.
           11=LINK REGISTER.
* I 7 = VP #
* I 6 = # OF PGS
T:SNAC EQU $
SNAC1 EQU $
      LI,2 SNAC2=X'80000'
      CI,7 X'100'
      BL T:SACP1
      SUA X'21'
SNAC2 RES 0
      AI,7 1
      BDR,6 SNAC1
      B *11
*

```

```

546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561      01 0009A
562 01 0009A 02200070 A
      01 0009B 0B500000 N
563 01 0009C 22400002 A
564      01 0009D
565      01 0009D
566 01 0009D 4B600000 X
567 01 0009E 4B700000 X
568 01 0009F 19700000 X
569 01 000A0 699000B6
570 01 000A1 22D00000 N
571 01 000A2 71DE0000 N
572 01 000A3 683000A6
573 01 000A4 714E0000 X
574 01 000A5 693000B6
575
576 01 000A6 31600000 X
577 01 000A7 692000B6
578 01 000A8 756E0000 N
579 01 000A9 754E0000 X
580 01 000AA 21400001 A
581 01 000AB 683000AE
    
```

```

PAGE
NAME: T:ISADO,T:SAD1
PURPOSE: TO MAP A SPECIFIED VIRTUAL PAGE ONTO
A SPECIFIED PHYSICAL PAGE.
DESCRIPTION: THIS ROUTINE IS USED BY BOTH
APWP AND CVM CALS. IT PLACES THE
GIVEN PHYSICAL PAGE NUMBER INTO JX:CMAP
AND SETS JB:LMAP TO EITHER 01 OR 02
DEPENDING ON THE CAL. IT SETS THE ACCESS
PROTECTION AND LOADS THE HARDWARE MAP
AND PROTECTION REGISTERS.
*
6 = PHYSICAL PAGE NUMBER
*
7 = VIRTUAL PAGE NUMBER
*
8 = ACCESS CODE FOR PAGE
    
```

```

T:ISADO EQU $
PUSH 7,5

LI,4 2 INDICATE PWP ENTRY AND LMAP CONSTANT
T:ISAD1 EQU $
SAD0B EQU $
AND,6 M13
AND,7 M8
CLM,7 JIBUP IS VP WITHIN USER PROG LIMITS
BCS,9 SADOR ERROR
LI,13 FPMC
COMPARE,13 JX:CMAP,7
BE $+3
CB,4 JB:LMAP,7 ENABLE SAD OVER SAD....
BNE SADOR ERROR
REF HIGH
CW,6 HIGH
BG SADOR DON'T ALLOW MORE THAN PHYS
STORE,6 JX:CMAP,7
STB,4 JB:LMAP,7
CI,4 1 WAS ENTRY FOR SAD
BE SAD0D YES
    
```

NO1 17:45 SEP 08, 1975

582 01 000AC 32400008 A
583 01 000AD 680000B2
584 01 000AE
585 01 000AE 22400000 A
586 01 000AF 208FFFD0 A
587 01 000B0 681000B2
588 01 000B1 20400002 A
589 01 000B2 6A200337
590 01 000B3
591 01 000B3 6AB0035D
592 01 000B4 6AB00222
593 01 000B5 22C00000 A
594 01 000B6
595 01 000B6 02200070 A
596 01 000B7 0A500000 N
597 01 000B8 7020000C A
598 01 000B9 F800000B A
599
600
601
602
603 01 000BA 08C00000 N
604 01 000BB 08300000 N
605 01 000BC F800000B A

LW,4 8
B SAD,1
SAD0D EQU *
* SET AC
LI,4 0
AI,8 X'80'-X'B0'
BGE *+2
SAD1M1 AI,4 2
SAD1 BAL,2 T;SAC
* EXECUTE AC
BAL,11 T;SXACM3
* EXECUTE MAP
BAL,11 T;SXMAPM3
LI,12 0
SAD0R EQU \$
PULL 7,5
LC 12
B *11
*
* ERROR RETURN FROM TITOTESZ
*
SIZERR PULL 12
PULL 3
B *11

PWP ENTRY, AC INPUT IN R8
GO SET ACCESS
AC
SET CC AS NO ERROR
RESTORE REGS
SET CC
RETURN

```

606
607
608
609
610
611
612
613
614
615
616
617
618
619      01 000RD
620
621      01 000BD      22C00000 N
622      01 000BE      71CE0000 N
623      01 000BF      693001B1
624              01 000P0
625      01 000C0      32400000 X
626      01 000C1      09300000 X
627      01 000C2      22000143
628
629
630
    
```

```

PAGE
NAME:      TIGVPI
PURPOSE:   TO ALLOCATE A GIVEN VIRTUAL PAGE.
DESCRIPTION: THIS ROUTINE CHECKS THE USERS SIZE
            BY CALLING TITOTESZ. IT ASSIGNS A
            PHYSICAL PAGE, UNLESS CALLED BY TIGNVPI,
            BUILD A COMMAND LIST ENTRY AND OBTAINS A
            SWAPPER GRANULE. IT MAINTAINS THE COUNTS
            FOR EACH TYPE OF MEMORY (DATA, PROCEDURE
            DCB, ETC.) AND SETS THE APPROPRIATE ACCESS
            PROTECTION CODE. IT LOADS THE HARDWARE
            MAP AND ACCESS REGISTERS BY CALLING TISXMAPM3
            AND TISAC.
TIGVPI     EQU      $
* TEST WHETHER THE VP IS AVAILABLE
            LI,12    FPMC      FREE PG MAP CONSTANT
            COMPARE,12 JXICMAP,7  IS REQUESTED VP FREE
            BNE      GVP0
GVP2       EQU      $
            LW,4     SICUN     GET CURRENT USER #
            PSW,3    TSTACK
            LI,0     GVP26
*
*          B          TITOTESZ
*
    
```



```

631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667

```

	01	000C3	72C80000	X
	01	000C4	72D00000	X
	01	000C5	683000CF	

```

PAGE
NAME: TITOTESZ
PURPOSE: TO COMPUTE THE USERS SIZE AND DETERMINE
DESCRIPTION: IF IT IS WITHIN VARIOUS LIMITS.
              THERE ARE THREE SIZES COMPUTED. THE
              USERS VIRTUAL SIZE IS FOUND BY:
              UB:PCT # PAGES HE NOW OWNS
              *(PUL=PLL)+PDP*1 # PROC PGS POTENTIAL
              *(DUL=DLL)+PCD*1 # DATA PGS POTENTIAL
              *PCC # CONTEXT PGS
              *APWP # ASSIGN PHYS WORK PGS

              THIS SIZE CANNOT EXCEED USERS AUTHORIZATION
              OR SYSTEM LIMIT UNLESS IT IS A SHARED
              PROCESSOR ALLOWED TO OBTAIN ALL MEMORY.

              THE NUMBER OF CONTEXT PAGES, PLUS THE
              NUMBER OF POTENTIAL SPARE BUFFER PAGES
              ARE ADDED TO THE USERS SIZE. THIS SIZE
              CANNOT EXCEED THE MAXIMUM VIRTUAL SWAP
              SPACE (S:I:CORE).

              THE NUMBER OF PAGES NEEDED FOR SHARED
              PROCESSORS, INCLUDING SPECIAL SHARED, CORE
              LIBRARY, TEL, ETC. ARE ADDED TO THE TOTAL.
              THIS SUM IS TESTED AGAINST S:PCORE TO
              INSURE THE USER DOES NOT EXCEED MAXIMUM
              AVAILABLE PHYSICAL MEMORY.
* BAL,0 TITOTESZ R4=CUN
*
* R7=VIRTUAL PAGE#
* R7=0 INDICATES THIS IS A TEST FOR PCORE EXCEEDED
* DESTROYS R1,R2,R3,R12,R13
*
TITOTESZ EQU $
                LB,12 UB:PCT,4 USER PROC + DATA + CONTEXT + DY DAT
                LB,13 JB:PCP PAGE COUNT OF USERS PROCEDURE
                BEZ N00LAY SHARED PROCESSOR RUNNING OR NO TREE

```

H01 17:45 SEP 08, '75

668	01	000C6	22100001	A
669	01	000C7	72120000	X
670	01	000C8	30D00001	A
671	01	000C9	38C0000D	A
672				*
673				*
674				*
675	01	000CA	30C00000	X
676	01	000CB	38C00000	X
677	01	000CC	30C00000	X
678	01	000CD	38C00000	X
679	01	000CE	20C00002	A
680	01	000CF		
681	01	000CF	21700000	A
682	01	000D0	683000FE	
683	01	000D1	22200003	A
684	01	000D2	72240000	X
685	01	000D3	38C00002	A
686	01	000D4	69C00000	N
687	01	000D5	22100000	N
688	01	000D6	72D20000	A
689	01	000D7	693000DD	
690		00000001		
691	01	000D8	22200001	A
692	01	000D9	72D40000	X
693	01	000DA	70200000	X
694	01	000DB	688000DD	
695	01	000DC	72D40000	X
696	01	000DD		
697	01	000DD	22100000	N
698	01	000DE	22200002	A
699	01	000DF	72300000	X
700	01	000E0		
701	01	000E0	71220000	X
702	01	000E1	693000E9	
703	01	000E2	20300000	A
704	01	000E3	683000E8	

*
*
*

N06LAY

C0RATX

GVP23

GVP230

LI,1	1
LB,1	JBIPCP,1
AW,13	1
SW,12	13
AW,12	JIPUL
SW,12	JIPLL
AW,12	JIDUL
SW,12	JIDLL
AI,12	2
EQU	\$
CI,7	0
BF	GVP27
LI,2	3
LB,2	JBIPCP,2
SW,12	2
PUSH	12
LI,1	J+JBMNPA
LB,13	0,1
BNEZ	GVP23
EQU	1
LI,2	C0RATX
LB,13	SB;RBMX,2
LC	JIJIT
BCR,8	\$+2
LB,13	SBIRBMX,2
EQU	\$
LI,1	JBUPVP
LI,2	2
LB,3	JBIPCP
EQU	\$
CB,2	JB;LMAP,1
BNE	GVP235
AI,3	0
BFZ	GVP232

49
USER RUNNING SO FIND HIS TRUE SIZE
PAGE COUNT OF DATA
CURRENT PR0CD PGS + CURRENT DATA PG
TAKEN OUT OF TOTAL PAGE COUNT
NOW ADD PROCEDURE AND DATA BACK
IN BASED ON JIT LIMITS TO ALLOW
FOR POSSIBLE SEGMENT LOADING

BECAUSE LIMITS ARE INCLUSIVE

IS THIS A PCORE TEST
YES-SKIP USER CORE CHECKS

PAGE COUNT CONTEXT

SAVE CURRENT USER SIZE

GET JIT LIMIT
IF NOT 0, USE IT

CORE RAT INDEX
SYSTEM LIMIT FOR BATCH CORE

SYSTEM LIMIT FOR ONLINE CORE

START AT BEGINNING OF USER
LOOKING FOR APWPAGES AND
RETAIN PROCEDURE COUNT

AND ADDING THEM TO USER TALLY
IS IT AN APWP
YES, HAS IT ALREADY BEEN COUNTED
NO, SINCE PROCEDURE IS ZERO

H01 17:45 SEP 08, '75

```

705 01 000E4 19100000 X
706 01 000E5 689000E9
707 01 000E6 19100000 X
708 01 000E7 689000E9
709 01 000F8 GVP232
710 01 000E8 20C00001 A
711 01 000F9 GVP235
712 01 000E9 20100001 A
713 01 000EA 31100000 X
714 01 000EB 682000E0
715 01 000EC 31C0000D A
716 01 000ED 691000F9
717 01 000EE 217FFFFFF N
718 01 000EF 682000F9
719 01 000F0 32100000 X
720 01 000F1 4A100000 X
721 01 000F2 683000F9
722 01 000F3 72280000 X
723 01 000F4 693000F6
724 01 000F5 72280000 X
725 01 000F6 32100000 X
726 01 000F7 31140000 X
727 01 000F8 F8400000 A
728 01 000F9 GVP25
729 01 000F9 22200003 A
730 01 000FA 72240000 X
731 01 000FB 30C00002 A
732 01 000FC 71700000 X
733 01 000FD 68200108
734 01 000FE GVP27
735 01 000FE 72100000 X
736 01 000FF 20100001 N
737 01 00100 22200000 N
738 01 00101 72240000 X
739 01 00102 38100002 A
740 01 00103 30C00001 A
741 01 00104 21700000 N

```

```

CLM,1 J;PLL
BCR,9 GVP235
CLM,1 J;DLL
BCR,9 GVP235
EQU $
AI,12 1
EQU $
AI,1 1
CW,1 J;EUP
BLE GVP230
CW,12 13
BL GVP25
CI,7 J;VVP,1
BLE GVP25
LW,1 Y003E
LS,1 J;RNST
BEZ GVP25
LB,2 UB;APR,4
BNEZ $+2
LB,2 UB;ASP,4
LW,1 Y08
CW,1 P;SA,2
BAZ *0
EQU $
LI,2 3
LB,2 JB;PCP,2
AW,12 2
CB,7 JB;FBUL
BLE GVP90
EQU $
LB,1 JB;FBUL
AI,1 *JXBUFVP+1
LI,2 JBFBUC
LB,2 J;JIT,2
SW,1 2
AW,12 1
CI,7 JBUPVP

```

```

IF IN PROCEDURE OR STATIC DATA
THEN THE PAGE HAS ALREADY BEEN
ACCOUNTED FOR, OTHERWISE
ADD IT TO USER TOTAL

```

INCREMENT TOTAL SIZE

```

MOVE TO NEXT VIRT PAGE
CONTINUE TILL DONE

```

```

HAS MAX BEEN ALLOC
NO

```

```

TRYING TO GET A SPARE
YES=CONTINUE

```

```

IF ASP, ALLOW EVERYTHING
YES, CHECK FOR SPEC PROC

```

```

IS PROCESSOR ALLOWED MAX CORE
NO=ABORT HIM

```

COUNT OF CONTEXT

```

GETTING A FILE MGMT BUFFER
YES=DONT HOLD BACK FOR IT

```

```

HOLD BACK CONTEXT NOT YET ALLOCATED
MAX # FPOOLS

```

```

# FPOOLS ALLOCATED
# FPOOL LEFT TO ALLOCATE
HOLD THEM IN RESERVE
GETTING A USER PAGE

```

401 17:45 SEP 08, '75

742 01 00105 68100108
 743 01 00106 21700000 A
 744 01 00107 69300111
 745 01 00108 2220FFFF A
 746 01 00109 4B200000 X
 747 01 0010A 68300111
 748 01 0010B 22100000 N
 749 01 0010C 72120000 X
 750 01 0010D 21100002 A
 751 01 0010E 68100111
 752 01 0010F 20C00002 A
 753 01 00110 38C00001 A
 754 01 00111 31C00000 X
 755 01 00112 69100115
 756 01 00113 21700000 A
 757 01 00114 69300000 A
 759
 760
 761
 762 01 00115 52280000 X
 763 01 00116 21200080 A
 764 01 00117 68400110
 765 01 00118 72180000 X
 766 01 00119 32300000 X
 767 01 0011A 31320000 X
 768 01 0011B 6940012E
 769 01 0011C 6800011F
 770 01 0011D 72180000 X
 771 01 0011E 68300124
 772 01 0011F 30C00000 X
 773 01 00120 38C00000 X
 774 01 00121 20C00001 A
 775 01 00122 21200080 A
 776 01 00123 69400130
 777
 778

GVP90
 BGE GVP90
 CI,7 0
 BNE GVP24
 LI,2 X'FFFF'
 AND,2 JH;LDCF
 BEZ GVP24
 LI,1 JBCBUC
 LB,1 JIJIT,1
 CI,1 2
 BGE GVP24
 AI,12 2
 SW,12 1
 EQU 8
 CW,12 SLICORE
 BL GVP240
 CI,7 0
 BNE *0
 *
 * USER SWAP SIZE OK-NOW CHECK FOR PCORE EXCEEDED
 *
 GVP240
 LH,2 UHIFLG,4
 CI,2 TIC
 BAZ GVP241
 LB,1 UBIACP,4
 LW,3 Y4
 CW,3 PISA,1
 BANZ GVP250
 B *+3
 GVP241
 LB,1 UBIAPR,4
 BEZ GVP242
 AW,12 JIPUL
 SW,12 JIPLL
 AI,12 1
 CI,2 TIC
 BANZ GVP255
 *
 *

51
 YES-NOW HOLD OUT UNALLOCATED CPBOOLS
 PERHAPS IT IS A TEST FOR PCORE
 NO-GETTING A CPBOL
 ALLOWED SYMBIONT ACCESS
 NO
 # CPBOOLS USED
 DEFAULT OR MORE
 YES-DONT HOLD BACK ANY MORE
 HOLD BACK THOSE NOT
 NOT YET ALLOCATED
 USER PAGES : MAX SWAP SIZE
 USER SIZE IS OK
 IS THIS A CHECK FOR PCORE
 NO-GETTING USER PAGE-ABORT HIM
 *
 * USER SWAP SIZE OK-NOW CHECK FOR PCORE EXCEEDED
 *
 GET STANDARD SHARED PROC. SIZE
 IS COMMAND PROCESSOR IN CONTROL
 NO-SEE IF THERE IS APR
 GET COMMAND PROCESSOR NUMBER
 IS CP A SPECIAL SHARED PROCESSOR
 YES-GO COMPUTE SPEC. SHRD. PROC. SIZE
 NO-CP SIZE IS INCLUDED IN STD. SHRD.
 IS STD. SHRD. PROC. ASSOCIATED
 NO-SEE IF WE SHOULD HOLD BACK FOR TEL
 SIZE OF APR OR ACP(NOT SPECIAL)
 FROM JIT LIMITS (ALLOWS FOR SEG LOAD
 BECAUSE LIMITS ARE INCLUSIVE
 DID WE JUST ADD IN ACP
 YES-DONT CHECK FOR TEL
 GET SPECIAL SHARED PROCESSOR SIZE
 LARGEST OF ACP/ASP/DB

H01

17145 SEP 08, 175

779 01 00124 70200000 X
780 01 00124 68800130
781 01 00125 32200000 X
782 01 00126 31200000 X
783 01 00127 69400130
784 01 00128 22100000 N
785 01 00129 12200008
786 01 0012A 11220000 X
787 01 0012B 6830012E
788 01 0012C 6410012B
789 01 0012D 72220000 X
790 01 0012E 68000131
791 01 0012F 01 00130
792 01 00130 22200000 A
793 01 00131 72180000 X
794 01 00132 72D20000 X
795 01 00133 72180000 X
796 01 00134 71D20000 X
797 01 00135 68100137
798 01 00136 72D20000 X
799 01 00137 31D00002 A
800 01 00138 6810013A
801 01 00139 32D00002 A
802 01 0013A 30C0000D A
803 01 0013B 32D00000 X
804 01 0013C 21700000 A
805 01 0013D 6930013F
806 01 0013E 20D00001 A
807 01 0013F 31C0000D A
808 01 00140 F8100000 A
809 01 00141 20000001 A
810 01 00142 F8000000 A
811
812
813
814
815 01 00143 680000BA

GVP242

EQU *
LC JIJIT
BCR,8 GVP255
LW,2 Y002
CW,2 JIJTELFLGS
BANZ GVP255
LI,1 PNAMEND
LD,2 TEL
CD,2 PINAME,1
BE \$+2
BDR,1 \$=2
LB,2 PBIPSZ,1
B GVP256
EQU *
LI,2 0
LB,1 UBASP,4
LB,13 PBIPSZ,1
LB,1 UBIDB,4
CB,13 PBIPSZ,1
BGE \$+2
LB,13 PBIPSZ,1
CW,13 2
BGE \$+2
LW,13 2
AW,12 13
LW,13 S;PCORE
CI,7 0
BNE \$+2
AI,13 1
CW,12 13
BGE *0
AI,0 1
B *0

GVP250

GVP255

GVP256

*
*
*

GVP26

B

SIZERR

ON-LINE USER
NO
DBES ON-LINE USER WANT
SPACE HELD IN RESERVE FOR TEL
NO
FIND TELS NUMBER

FIND TELS SIZE

FIND # PGS REQUIRED FOR SHARED
NO SP. SHARED CP
SP.SHARED PROC OR CORE LIBRARY
ASP SIZE
DEBUGGER
ASP I DB

ASP/DB : SP.SHARED ACP

GREATEST POSSIBLE SIZE

IS THIS A PCORE TEST

NO

ADJUST FOR BGE BELOW

IS USER REQUIREMENT > PHY CORE

TITOTESZ TEST FAILED

17145 SEP 08, '75

816	01	00144	08C00000	N
817	01	00145	20C00001	A
818	01	00146	22200000	N
819	01	00147	71C40000	A
820	01	00148	6820014A	
821	01	00149	73140000	A
822	01	0014A	08300000	X
823	01	0014B	09B00000	N
824	01	0014C	09600000	N
825	01	0014D	22400000	N
826	01	0014E	72180000	A
827	01	0014F	201FFFFFF	A
828	01	00150	75180000	A
829	01	00151	6810016E	
830	01	00152	22100003	A
831	01	00153	75180000	A
832				
833		01 00154		
834	01	00154	22400000	N
835	01	00155	72180000	A
836				
837				
838				
839				
840	01	00156	6AB002A9	
841	01	00157	6810015B	
842				
843	01	00158	22600000	N
844	01	00159	6AB00000	X
845	01	0015A	68000154	
846		01 0015B		
847	01	0015B	20100008	A
848	01	0015C	75180000	A
849	01	0015D	22400000	N
850	01	0015E	32C00000	X
851	01	0015F	68300161	
852	01	00160	22400000	N

PULL	12	GET CURRENT SIZE
AI,12	1	FOR NEW PAGE
LI,2	JB,PEAK	
CR,12	0,2	IS THIS A NEW PEAK
BLE	*+2	NO
MTB,1	0,2	YES-INCREMENT COUNT BY ONE
PLW,3	TSTACK	
PUSH	11	
PUSH	6	
LI,4	J+JBNRG	
LB,1	0,4	
AI,1	=1	
STB,1	0,4	STORE REMAINING GR. COUNT
BGEZ	GVP45	BR. IF MORE GRANULES REMAIN
LI,1	3	
STB,1	0,4	ELSE GET ANOTHER GROUP.
* GET FOUR GRANULES.		
GVP3	EQU \$	
LI,4	J+JBNASP	
LB,1	0,4	GET NEXT GRANULE POSITION
* I 1 = NEXT GRANULE POSITION		
* B 15 = DISC ADR ALLOCATED		
* B CC4 = 0 A GRANULE WAS AVAILABLE		
* B 1 = PRES GRAN POSITION		
BAL,11	T;SGA	SWAPPER GRANULE ALLOCATION
BCR,1	GVP4	A GRAN WAS AVAILABLE
* NO DISC GRANULE WAS AVAILABLE - REG - UPON RETURN, GET GRAN AGAIN		
LI,6	E;ND	NO DISC EVENT #
BAL,11	T;REG	
B	GVP3	TRY FOR DISC AGAIN
GVP4	EQU \$	
AI,1	8	FOUR GRANULES PER CLUSTER*2
STB,1	0,4	SAVE IT IN JIT FOR NEXT REQUEST
LI,4	J;CL	
LW,12	J;AJ	GET ADDRESS OF CLIST IN JIT
BEZ	*+2	
LI,4	J;AJIT	OR AJIT, IF USED.

```

853 01 00161 30400000 X
854 01 00162 32100000 X
855 01 00163 36100000 X
856 01 00164 55F20000 X
857 01 00165 32200000 X
858 01 00166 72C40000 N
859 01 00167 22D00000 N
860 01 00168 6820016A
861 01 00169 32C00000 X
862 01 0016A 25C0000B A
863 01 0016B 32D00000 X
864 01 0016C 47C80000 A
865 01 0016D 33200000 X
866
867
868 01 0016E
869 01 0016E 215FFFFFF A
870 01 0016F 69100172
871 01 00170 68300174
872 01 00171 6A20025F
873 01 00172 20300000 A
874 01 00173 69300179
875
876 01 00174
877 01 00174 32400000 X
878 01 00175 52080000 X
879 01 00176 22300022 A
880 01 00177 4B00000D
881 01 00178 55080000 X
882 01 00179
883 01 00179 6A100000 X
884 01 0017A 6840017C
885 01 0017B 33100000 X
886
887 01 0017C 22200000 N
888 01 0017D 22000000 A
889 01 0017E 22400000 N
    
```

```

AW,4 JICL
LW,1 JICL CLIST LENGTH
DW,1 XA
STH,15 JHIDA,1 ADDRESS TABLE, STORE DA
LW,2 SICUN
LOAD,12 UXIJIT,2 PHYSICAL PAGE NR OF JIT
LI,13 ,BIG
BLEZ $+2
LW,12 JIAJ ON BIG9 THE DA IS IN AJIT
SLS,12 11 BA(JIT PAGE)
LW,13 XFFF800
STS,12 0,4 STORE PHYS PG ADDRESS IN SEEKI0CD
MTW,2 JICL INC LENGTH OF CLIST,
* GET PP # FROM FREE PG POOL UNLESS THIS CAL IS GET VP, NO PP
* B 3 = PHY PG # OBTAINED
GVP45 EQU $
CI,5 =1
BL GVP51
BE GVP52
BAL,2 TIGPP
GVP51 AI,3 0
BNE GVP5
* RESET READY FLG IF NO PP
GVP52 RFS 0
LW,4 SICUN
LH,0 UHIFLG,4
LI,3 NPMC
AND,0 NREADY RESET READY TO RUN BIT
STH,0 UHIFLG,4
GVP5 EQU $
BAL,1 RTCHK IS THIS USER LOCKED IN CORE
BCR,4 GVP53 NO
MTW,1 SIRT0RE YES...ACCOUNT FOR NEW CONTEXT PGE
GVP53 ,
LI,2 BA(JB:VLH)=BA(JB:ILMAP) IX TO 1ST LMAP LINK
LI,0 0
LI,4 JICL
    
```

H01 17:45 SEP 08 175

890	01	0017F	32C00000	X
891	01	00180	68300182	
892	01	00181	22400000	N
893	01	00182	30400000	X
894	01	00183	33200000	X
895	01	00184	22D00003	A
896	01	00185	68000232	
897	01	00186		
898	01	00186	72C80000	N
899	01	00187	75C60000	N
900	01	00188	75380000	N
901	01	00189	753E0000	N
902				
903	01	0018A	32400000	X
904	01	0018B	73180000	X
905	01	0018C	52080000	X
906	01	0018D	49000000	X
907	01	0018E	55080000	X
908				
909	01	0018F	221FFFFB	A
910	01	00190	19720000	F
911	01	00191	68900199	
912	01	00192	65100190	
913	01	00193	221FFFFE	A
914	01	00194	31700000	X
915	01	00195	68200199	
916	01	00196	31700000	X
917	01	00197	69200199	
918	01	00198	221FFFFD	A
919				
920	01	00199	20100005	A
921	01	0019A	73120000	X
922				
923				
924				
925	01	0019B	21100003	A
926	01	0019C	6830019F	

LW,12	JIAJ
BEZ	*+2
LI,4	JIAJIT
AW,4	JICLE
MTW,2	JICLE
LI,13	3
B	IV3,1
IV9	RES 0
LOAD,12	MXIPPUT,4
STORE,12	MXIPPUT,3
STORE,3	MXIPPUT,4
IV10	STORE,3 JX,CMAP,7
* SET SWAP PURE PROCEDURE FLAG BIT	
LW,4	SICUN
MTB,1	UBIPCT,4
LW,0	UHIFLG,4
OR,0	XPPSWAP
STH,0	UHIFLG,4
* TEST WHICH AREA PG IS IN	
LI,1	=5
GVP6	CLM,7 WA(DA(J,PLL)+5),1
BCR,9	GVP7
BIR,1	GVP6
LI,1	=2
CW,7	JIPUL
BLE	GVP7
CW,7	JIEUP
BG	GVP7
LI,1	=3
* UPDATE PG CNT FOR APPROPRIATE AREA	
GVP7	AI,1 5
MTB,1	JBIPCP,1
* SET UP FOR TISAC	
* I 13 = AC FOR JAC	
* I 4 = AC FOR DAC	
CI,1	3
BE	GVP8

INC CLIST LENGTH FOR NEW WRITE IBCD
SEEK ORDER

WHAT ITS LINKED TO
LINK NEW PP
LINK PREV TO NEW PP
PUT PP INTO CMAP

UPDATE USERS PG CNT TOTAL NEEDED

SET PPSWAP BIT

WHICH PAGE IS VP IN
FOUND AREA

ASSUME CONTEXT

DYNAMIC DATA

CONVERT INDEX
INCR APPRO PG COUNT

IS PAGE IN CONTEXT AREA
YES=NO CHANGE TO JIJAC

H01 17145 SEP 08, 175

927 01 0019D 0019D
 928 01 0019D 72420000
 929 01 0019E 6A200337
 930
 931 01 0019F 0019F
 932 01 0019F 32C00000 X
 933 01 001A0 22400000 N
 934 01 001A1 72480000 A
 935 01 001A2 693001A4
 936 01 001A3 20C00002 A
 937 01 001A4 6A0003BC
 938 01 001A5 723E0000 N
 939 01 001A6 203FFFDE A
 940 01 001A7 683001AD
 941 01 001A8 217FFFFFF N
 942 01 001A9 682001AD
 943
 944
 945
 946
 947 01 001AA 6AB00222
 948
 949
 950
 951
 952 01 001AB 22C00000 N
 953 00000001
 954 01 001AC 6AB0035E
 955
 956
 957
 958
 959 *S*
 960 *S*
 961 *S*
 962 *S*
 963 *S*

GVP75 EQU *
 LB,4 UACT,1
 BAL,2 TISAC
 * TEST WHETHER AJIT NEEDED
 GVP8 EQU *
 LW,12 J,CLE
 LI,4 J+JBNRG
 LB,4 0,4
 BNEZ \$+2
 AI,12 2
 BAL,0 TIGAJP
 LOAD,3 JXICMAP,7
 AI,3 =NPMC
 BFZ GVP10
 CI,7 J0VVP,1
 BLE GVP10
 * SET UP & EXECUTE MMC FOR MAP
 * I 12 = IMAGE ADR
 * I 14 = # OF PGS
 * I 15 = STARTING VP #
 BAL,11 T: SXMAPM3
 * SET UP & EXECUTE MMC FOR AC
 * I 12 = IMAGE ADR
 * I 14 = # OF PGS
 * I 15 = STARTING VP #
 LI,12 JIJAC
 DB 1
 BAL,11 GVPA
 *
 * GVPA(T: SXAC) NOW DOES SJAC STUFF (13 WORDS, RJR)
 *
 ELSE T: SXAC(SJAC)
 LI,11 GVP10
 LW,4 S: CUN
 LW,0 UH: FLG,4
 CI,0 SJAC
 BAZ GVPA

AC FOR APPRO AREA
SET UP USER & DELTA AC IN IMAGES

IF THE NR OF REMAINING GRANULES
IS 0
ADD TWO FOR NEXT SEEK.
GET AJIT PP
GET PP #

WAS A SPARE BUFFER OBTAINED
YES-DONT LOAD MAP NOR ACCESS

USER RUNNING = IMAGE ADR
T: SXAC(SJAC)
*SET HIS ACCESS ON THE PAGE RIGHT

HO1 17145 SEP 08, '75

964			*S*	LI,14	JJITVP
965			*S*	LI,15	XIFOI
966			*S*	CS,14	7
967			*S*	BNE	GVPA
968			*S*	LI,14	SPJITAC
969			*S*	LW,15	SPJACTRL
970			*S*	LPC,14	0
971			*S*	LI,3	1
972				FIN	
973				* TISXAC RETURNS WITH REG 3 = 1	
974		01 001AD		GVP10	EQU *
975	01	001AD 08600000	N		PULL 6
976	01	001AE		GVPB	RES 0
977	01	001AE 08B00000	N		PULL 11
978	01	001AF 20B00001	A		AI,11 1
979	01	001B0 70360000	X		LCF CCS,3
980	01	001B1		RVP3	RES 0
981	01	001B1		GVPB	RES 0
982	01	001B1 F800000B	A		B *11

START AND COUNT CONTROL

TISXAC(SJAC)

3=0=>NB PP, =1=>NORMAL EXIT.

```

983
984
985
986
987
988
989
990
991
992
993
994
995
996      01 001B2
997
998      01 001B2      22C00000 N
999      01 001B3      71CE0000 N
1000     01 001B4      683001B1
1001           01 001B5
1002     01 001B5      724E0000 X
1003     01 001B6      71700000 X
1004     01 001B7      683001BA
1005     01 001B8      21400000 A
1006     01 001B9      F830000B A
1007     01 001BA     09B00000 N
1008     01 001BB     21400001 A
1009     01 001BC     693001C0
1010     01 001BD     73FE0000 X
1011     01 001BE     75CE0000 N
1012     01 001BF     68000335
1013     01 001C0     32200000 X
1014     01 001C1     72440000 X
1015     01 001C2     683001CD
1016
1017     01 001C3     52440000 X
1018     01 001C4     49400000 X
1019     01 001C5     55440000 X
    
```

```

PAGE
NAME: TIRVPI
PURPOSE: TO RELEASE A GIVEN VIRTUAL PAGE.
DESCRIPTION: THE VIRTUAL PAGE IS REMOVED FROM THE
JBILMAP LIST AND THE COMMAND LIST.
IT IS REMAPPED INTO THE MONITOR FPMC PAGE.
THE ACCESS PROTECTION SET NO NO ACCESS AND ITS
PHYSICAL PAGE RETURNED TO THE MONITOR FREE
PAGE LIST, UNLESS IT WAS CALLED BY TIRVSP1.
ITS SWAP GRANULE IS RELEASED.
THE COUNT FOR ITS MEMORY TYPE IS DECREMENTED,
(DATA, DCB, ETC.) AND THE ACCESS PROTECTION
AND MAP REGISTERS ARE LOADED.
TIRVPI EQU $
* TEST WHETHER THE VP IS IN USE
      LI,12 FPMC FREE PG MAP CONST
      COMPARE,12 JXICMAP,7 IS VIRT PG TO RELEASE IN USE
      BE RVP3
RVP2 EQU $
      LB,4 JBILMAP,7
      CB,7 JBIVLT
      BE $+3 DONT TEST TAIL
      CI,4 0 DONT RELEASE A CMAP PAGE
      BEZ *11 NOT REPRESENTED IN LMAP
      PUSH 11
      CI,4 1
      BNE RVP42 IF NOT A 'CVMI' PAGE
      MTB,1 JBILMAP,7 GET RID OF 'CVMI' FLAG
      STORE,12 JXICMAP,7 ELIMINATE VIRTUAL PAGE
      B RVP9 SET ACCESS AND GET OUT
RVP42 LW,2 S;CUN
RVP43 LB,4 UBIMF,2 IF USERS FUNCTION COUNT IS
      BEZ RVP44 NON-ZERO PUT HIM TO SLEEP.
      REF UHIFLG2
      LH,4 UHIFLG2,2 GET SECOND FLAGS
      BR,4 X8 SET DONT SWAP
      STH,4 UHIFLG2,2 AND REPLACE
    
```

1020				REF	X8	
1021	01	001C6	35600004	STW,6	4	
1022				RFF	EIQMF	
1023	01	001C7	22600000	LI,6	EIQMF	QUEUE FOR MASTER FUNCTION COUNT
1024	01	001C8	05800000	PUSH	11	
1025	01	001C9	6AB00000	BAL,11	TIREG	
1026	01	001CA	08B00000	PULL	11	
1027	01	001CB	32600004	LW,6	4	
1028	01	001CC	680001C1	B	RVP43	
1029	01	001CD		RVP44 EQU	*	
1030				* DELETE VP & PP		
1031	01	001CD	33E00000	MTW,=2	JICL	REDUCE CLIST LENGTH BY 1 IBCD
1032	01	001CE	22200003	LI,2	3	SETUP SEEK ORDER
1033	01	001CF	22400000	LI,4	JICL	SETUP END CLIST
1034	01	001D0	32D00000	LW,13	JIAJ	
1035	01	001D1	683001D4	BEZ	*+3	
1036	01	001D2	22D00000	LI,13	0	
1037	01	001D3	22400000	LI,4	JIAJIT	
1038	01	001D4	30400000	AW,4	JICL	
1039	01	001D5	22100000	LI,1	BA(JB:VLH)=BA(JB:LMAP)	IX TO 1ST LMAP LINK
1040	01	001D6	71720000	CB,7	JB:LMAP,1	IS THIS THE VP # TO RELEASE
1041	01	001D7	683001DE	BE	DV4	YES
1042	01	001D8	72120000	LB,1	JB:LMAP,1	NEXT LINK
1043	01	001D9	46D80000	XW,13	0,4	GET THE NEXT ADDRESS
1044	01	001DA	204FFFFE	AI,4	=2	AND RIPPLE PREVIOUS UP,
1045	01	001DB	F1200004	CB,2	*4	IS NEXT A SEEK IBCD
1046	01	001DC	682001DA	BLE	*=2	YES. SKIP IT.
1047	01	001DD	680001D6	B	DV2	
1048	01	001DE	35D80000	STW,13	0,4	MOVE PREV ENTRY UP
1049	01	001DF	72220000	LOAD,2	JXICMAP,1	
1050	01	001E0	22300000	LI,3	0	
1051	01	001E1	724E0000	LB,4	JB:LMAP,7	NXT LINK
1052	01	001E2	75420000	STB,4	JB:LMAP,1	
1053	01	001E3	753E0000	STB,3	JB:LMAP,7	
1054	01	001E4	693001E8	BNEZ	*+4	
1055	01	001E5	20100000	AI,1	0	
1056	01	001E6	691001EA	BLZ	*+4	

W01 17145 SEP 08, 175

1057	01	001E7	75100000	X	STB,1	JBIVLT	
1058	01	001E8	20100000	A	AI,1	0	
1059	01	001E9	681001EC		BGEZ	\$+3	
1060	01	001EA	22200000	N	LI,2	BA(JX;PPH)*BA(MX;PPUT)	
1061	01	001EB	25200000	N	SLS,2	=:BIG	
1062	01	001EC	723E0000	N	LOAD,3	JXICMAP,7	
1063	01	001ED	6D000037	A	DISABLE		
1064	01	001EE	21300022	A	CI,3	NPMC	
1065	01	001EF	693001F2		BNE	\$+3	
1066	01	001FO	225FFFFFF	A	LI,5	=1	
1067	01	001F1	680001F5		B	RVNP	RELEASE VIRTUAL, NO PHYSICAL
1068	01	001F2	72160000	N	LOAD,1	MX;PPUT,3	
1069	01	001F3	75140000	N	STORE,1	MX;PPUT,2	UNLINK
1070	01	001F4	73F00000	X	MTB,=1	JBIPPC	DECR USER PG CNT
1071		01 001F5			EQU	\$	
1072	01	001F5	75CE0000	N	STORE,12	JXICMAP,7	FREE PAGE
1073							
1074	01	001F6	32400000	X	* SET SWAP PURE PROCEDURE FLAG BIT		
1075	01	001F7	73F80000	X	LW,4	\$;CUN	
1076	01	001F8	52080000	X	MTB,=1	UB;PCT,4	
1077	01	001F9	49000000	X	LH,0	UH;FLG,4	
1078	01	001FA	55080000	X	BR,0	XPPSWAP	
1079	01	001FB	6D000027	A	STH,0	UH;FLG,4	
1080	01	001FC	21300022	A	ENABLE		
1081	01	001FD	68300204		CI,3	NPMC	DONT RELEASE IF WE DIDN'T GET
1082	01	001FE	21B0008E		BE	RNVP1	PAGE
1083	01	001FF	68300204		CI,11	7H3	ALSO, DONT RELEASE IF CALLED
1084					BF	RNVP1	BY T;RVSP1
1085	01	00200	6A200269		* RELEASE PP # TO FREE PG P06L		
1086	01	00201	6A100000	X	BAL,2	T;FPP	
1087	01	00202	68400204		BAL,1	RTCHK	IS THIS USER LOCKED IN CORE
1088	01	00203	33F00000	X	BCR,4	RNVP1	NA
1089		01 00204			MTW,=1	S;RTCORE	YES...COUNT IT DOWN
1090					RNVP1	EQU	
1091	01	00204	22400000	N	* RELEASE DISC GRANULE		
1092	01	00205	73180000	A	LI,4	J+JBNRG	INCR NR GRANULES REMAINING IN GROUP
1093	01	00206	72280000	A	MTB,1	0,4	
					LB,2	0,4	

H01 17:45 SEP 08, '78

1094	01	00207	21200004	A
1095	01	00208	69100212	
1096	01	00209	22200000	A
1097	01	0020A	75280000	A
1098	01	0020B	33E00000	X
1099	01	0020C	32100000	X
1100	01	0020D	36100000	X
1101	01	0020E	52F20000	X
1102	01	0020F	6AB002FD	
1103	01	00210	22400000	N
1104	01	00211	75180000	A
1105				
1106		01 00212		
1107	01	00212	221FFFFB	A
1108	01	00213	19720000	F
1109	01	00214	6890021C	
1110	01	00215	65100213	
1111	01	00216	221FFFFE	A
1112	01	00217	31700000	X
1113	01	00218	6820021C	
1114	01	00219	31700000	X
1115	01	0021A	6920021C	
1116	01	0021B	221FFFFD	A
1117				
1118		01 0021C		
1119	01	0021C	20100005	A
1120	01	0021D	73F20000	X
1121	01	0021E	217FFFFF	N
1122	01	0021F	682001AE	
1123				
1124				
1125				
1126				
1127	01	00220	68000335	
1128	01	00221	22B0035C	
1129				
1130				

CI,2	4	
BL	RVP5	
LI,2	0	
STB,2	0,4	CLEAR NR, REMAINING GR.
MTW,=2	JICLE	REMOVE SEEK FROM CLIST AND
LW,1	JICLE	RETURN GROUP OF FOUR GRANULES
DW,1	XA	COMPUTE NR OF SEEKS TO GET
LH,15	JHIDA,1	DISC SEEK ADDRESS TO RETURN.
BAL,11	TISGR	SWAPPER GRANULE RELEASE
LI,4	J+JBNASP	
STB,1	0,4	SET NEXT AVAIL GRAN POSITION
* TEST WHICH AREA PG IS IN		
RVP5	EQU	\$
	LI,1	=5
RVP6	CLM,7	WA(DA(JIPLL)+5),1 WHICH AREA IS VP IN
	BCR,9	RVP7
	BIR,1	RVP6
	LI,1	=2 ASSUME CONTEXT
	CW,7	JIPUL
	BLE	RVP7
	CW,7	JIEUP
	BG	\$+2
	LI,1	=3 DYNAMIC DATA
* UPDATE PG CNT FOR APPROPRIATE AREA		
RVP7	EQU	\$
	AI,1	5 CONVERT INDEX
	MTB,=1	JBIPCP,1 DECR APPR PG COUNT
	CI,7	JBVVP=1 WAS A SPARE BUFFER RELEASED
	BLE	GVPB YES=DBNT LOAD MAP NOR ACCESS
* UPDATE PG CNT TOTAL NEEDED		
* SET UP FOR TISAC		
* I 13 = AC FOR JAC		
* I 4 = AC FOR DAC		
	B	RVP9
RVP8	LI,11	RVP91
F	NAMEI	TISXMAPM3
F	PURPOSE:	TO LOAD THE MAP REGISTERS FOR A SINGLE PAGE.

MO1 17145 SEP 08, '75

1131 01 00222 32F00007 A
 1132 01 00223 22C00000 N
 1133 01 00224 22E00001 A
 1134
 1135
 1136
 1137
 1138 01 00225 22300000 A
 1139 01 00226 224FFFFFFE N
 1140 01 00227 68000362

TISXMAPM3 LW,15 7
 TISXMAPM2 LI,12 JXICMAP
 LI,14 1
 F NAME: TISXMAP
 F PURPOSE: TO LOAD THE MAP REGISTERS FROM THE USERS
 F IMAGE IN JXICMAP STARTING AT A SPECIFIED
 F PAGE FOR A SPECIFIED NUMBER OF PAGES.
 TISXMAP LI,3 0
 LI,4 IBIG=2
 B MMCC

VP #
 IMAGE ADDRESS
 # OF PGS

INST CODE
 SHIFT CODE

```

1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160 01 00228 19700000 X
1161 01 00229 6890023F
1162 01 0022A 0022A
1163 01 0022A 204FFFFE A
1164 01 0022B F1D00004 A
1165 01 0022C 6820022A
1166
1167 01 0022D 72C20000 N
1168 01 0022E 21C00022 A
1169 01 0022F 68300231
1170 01 00230 3200000C A
1171
1172 01 00231 72240000 X
1173 01 00232 72140000 X
1174 01 00233 69300236
1175 01 00234 75700000 X
1176 01 00235 68000241
1177 01 00236
    
```

```

PAGE
*****
*          SET VIRTUAL & PHYSICAL          *
*
*          LINK VP INTO LMAP                *
*          LINK PP INTO PPUT                *
*          SET PP IN CMAP                   *
*          SET UP NEW CL ENTRY &           *
*          SET UP NEW DA ENTRY             *
*          RIPPLE PP MEM ADR DOWN          *
*****
*          1 = NEXT LINK IN LMAP           *
*          2 = PREV ' ' ' '               *
*          I 3 = PP #                       *
*          4 = INDEX ADR FOR CL & DA TABLE *
*          I 7 = VP #                       *
*          12 = ACCUM                       *
*          I 15 = DA                        *
*
* IV4 CLM,7 J:PLL
*      BCR,9 IV6
*
* IV3 EQU $
*      AI,4 *2
*      CB,13 *4
*      BLE $*2
*
* KEEP LAST PREV PP #
*      LOAD,12 JXICMAP,1
*      CI,12 NPMC
*      BE $+2
*
* GET A VP LINK FROM LMAP
*      LB,2 JBILMAP,2
* IV31 LB,1 JB,LMAP,2
*      BNEZ IV32
*      STB,7 JBIVLT
*      B IV8
* IV32 RES 0
    
```

VP IS WITHIN LIMITS

DECR TO NEXT CLIST ENTRY
IS IT A SEEK
IF SO, SKIP IT.

SAVE PREV LINK
GET NXT LINK
MORE
UPDATE TAIL
NO MORE


```

1178
1179 01 00236 32C9FFFE A
1180 01 00237 71D0000C A
1181 01 00238 6920023A
1182 01 00239 32C9FFFC A
1183 01 0023A 35C80000 A
1184
1185 01 0023B 19100000 X
1186 01 0023C 68900228
1187
1188
1189 01 0023D 19700000 X
1190 01 0023E 68900241
1191
1192
1193
1194      01 0023F
1195 01 0023F 31700001 A
1196 01 00240 6910022A
1197      01 00241
1198
1199 01 00241 75740000 X
1200 01 00242 751E0000 X
1201 01 00243 32C00000 X
1202 01 00244 21C001FF A
1203 01 00245 6830024C
1204 01 00246 19100000 X
1205 01 00247 6890025C
1206 01 00248 19C00000 X
1207 01 00249 6890024C
1208 01 0024A 31C00001 A
1209 01 0024B 6910024D
1210 01 0024C 35100000 X
1211 01 0024D 32C00003 A
1212 01 0024E 20C02000 A
1213 01 0024F 25C0000B A
1214 01 00250 35C80000 A
    
```

```

* RIPPE MEM ADRS DOWN IN CL
  LW,12  =204
  CB,13  12
  BG     #+2
  LW,12  =404
  STW,12 004
  * IS VP FROM LMAP IN PURE PROCEDURE
  CLM,1  JIPLL
  BCR,9  IV4
  * VP FROM LMAP WAS NOT IN PURE P
  * IS VP REQUESTED IN PURE P - YES, FINISHED - NO, TEST MORE
  CLM,7  JIPLL
  BCR,9  IV8
  * BOTH VP FROM LMAP & ONE REQUESTED ARE IN PURE P
  * OR NEITHER ARE
  * IS VP REQUESTED < VP FROM LMAP - YES, FINISHED - NO, GO BACK
IV6 EQU $
  CW,7  1
  BL    IV3
  * VP REQ IS > THAN VP IN CHAIN
IV8 EQU $
  * INSERT VP INTO VIRTUAL LINK MAP
  STB,7  JBILMAP,2
  STB,1  JBILMAP,7
  LW,12  JIVLCS
  CI,12  XI1FFI
  BE     IV88
  CLM,1  JIPLL
  BCR,9  IV82
  * YES 1
  CLM,12 JIPLL
  BCR,9  IV88
  * YES 2
IV85 CW,12  1
  BL    IV89
IV88 STW,1  JIVLCS
IV89 LW,12  3
  AI,12 XI2000I
  SLS,12 11
  STW,12 004
  * GET PP #
  * ADD IN WRITE ORDER
  * GEN BA OF MEM ADR
  * STORE IN CLIST
    
```

H01 17:45 SEP 08, '75

1215	01	00251	21300022	A
1216	01	00252	68300189	
1217				
1218	01	00253	73100000	X
1219	01	00254	21200000	N
1220	01	00255	68300258	
1221	01	00256	32400000	A
1222	01	00257	69300186	
1223	01	00258	72C00000	N
1224	01	00259	75C60000	N
1225	01	0025A	75300000	N
1226	01	0025B	68000189	
1227	01	0025C	19C00000	X
1228	01	0025D	6890024A	
1229	01	0025E	6800024D	

	CI,3	NPMC	
	BE	IV10	
*	INSERT PP INTO	PPUT CHAIN	
	MTB,1	JBIPPC	INCR USER'S PG CNT IN CORE
	CI,2	BA(JB:VLH)*BA(JB:LMP)	DOES NEW PP GO ON BEG OF CHAI
	BE	**3	
	LW,4	0	
	BNEZ	IV9	
	LOAD,12	JXIPPH	YES, GET PREV 1ST PAGE
	STORE,12	MX,PPUT,3	LINK NEW PAGE TO IT
	STORE,3	JXIPPH	PUT NEW PP IN HEAD
	B	IV10	
IV82	CLM,12	JIPLL	
	BCR,9	IV85	YES 3
	B	IV89	NO 3

1230
 1231
 1232
 1233
 1234
 1235
 1236
 1237
 1238
 1239
 1240 01 0025F
 1241 01 0025F 6D000037 A
 1242 01 00260 32300000 X
 1243 01 00261 68300267
 1244 01 00262 72460000 N
 1245 01 00263 35400000 X
 1246 01 00264 69300266
 1247 01 00265 35400000 X
 1248 01 00266 33F00000 X
 1249 01 00267
 1250 01 00267 6D000027 A
 1251 01 00268 68040000 A
 1252
 1253

PAGE
 F NAME: TIGPP
 F PURPOSE: TO OBTAIN A PAGE FROM THE MONITOR FREE
 F PAGE LIST (MIFPPH)
 F DESCRIPTION: THE PAGE IS OBTAINED FROM THE HEAD OF THE
 F LIST (MIFPPH).
 *
 * 2 = LINK REGISTER
 * 3 = NEXT FREE PHY PAGE
 * 4 = FOLLOWING FPP IN CHAIN
 *

TIGPP EQU \$
 DISABLE \$
 LW,3 MIFPPH GET CONTENTS OF FREE PG POOL HEAD
 BEZ GPP1 IF 0, RETURN
 LOAD,4 MXIPPUT,3 GET NEXT PG IN CHAIN
 STW,4 MIFPPH & PUT PG INTO HEAD
 BNEZ *+2
 STW,4 MIFPPT NOTHING IN CHAIN SO CLEAR TAIL
 MTW,-1 MIFPPC DECR FPP COUNT
 GPP1 EQU \$
 ENABLE \$
 B 0,2
 *
 *

```

1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264      01 00269
1265 01 00269      224FFFFFF N
1266 01 0026A      25400077 A
1267 01 0026B      31400003 A
1268 01 0026C      6910026F
1269
1270
1271
1272
1273
1274
1275
1276
1277 01 0026D      0F000000 X
      01 0026E      00010001 A
1278      01 0026F
1279 01 0026F      22400000 A
1280 01 00270      75460000 N
1281 01 00271      6D000037 A
1282 01 00272      32400000 X
1283 01 00273      69300276
1284 01 00274      35300000 X
1285 01 00275      68000277
1286      01 00276
1287 01 00276      75380000 N
1288 01 00277      35300000 X
1289 01 00278      33100000 X
    
```

```

PAGE
*F* NAME: TIFPP
*F* PURPOSE: TO RETURN A PHYSICAL PAGE TO THE MONITOR
*F* FREE PAGE LIST:
*F* DESCRIPTION: THE PAGE IS ADDED TO THE TAIL OF
*F* THE LIST (MIFPPT).
* 2 = LINK REGISTER
* I 3 = PHY PAGE TO RELEASE
* 4 = PREV PG RELEASED
*
* TIFPP EQU $
LI,4 TOPROOT=1 LAST MONITOR LOCATION
SLS,4 =9 LAST MONITOR PAGE
CW,4 3 MON PAGE : RELEASING PAGE
BL 1A1 NOT RELEASING MONITOR PAGE
*S* SCREECH CODE: 01=01
*S* REPORTED BY: MM
*S* MESSAGE: ATTEMPT TO RELEASE MONITOR PAGE.
*S* REGISTERS: 3=PHYSICAL PAGE TO BE RELEASED.
*S* 7=VIRTUAL PAGE
*S* REMARKS: MAY BE CALLED TO RELEASE A PAGE FROM
*S* THE USERS PAGE CHAIN, OR, TO RELEASE A
*S* STOLEN PAGE.
TFPP1 SCREECH 1,1
1A1 EQU $
LI,4 0
STORE,4 MXIPPUT,3 0 TO END CHAIN
DISABLE
LW,4 MIFPPT GET LAST PG
BNEZ RPP2 THERE WAS ONE
STW,3 MIFPPH SET HEAD OF CHAIN
B RPP2+1 AVOID LINKING TO ZERO
RPP2 EQU $
STORE,3 MXIPPUT,4 LINK TO PREV PP
STW,3 MIFPPT SET TAIL
MTW,1 MIFPPC INCR FPP COUNT
    
```

NO1 17145 SEP 08, '75
1290 01 00279 68000267

B

GPP1

```

1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305      01 0027A
1306      01 0027A 6D000037 A
1307      01 0027B 32100000 X
1308      01 0027C 38100000 X
1309      01 0027D 69100283
1310      01 0027E 31100000 X
1311      01 0027F 69100283
1312      01 00280 6D000027 A
1313      01 00281 223FFFFFF A
1314      01 00282 F800000B A
1315      01 00283
1316      01 00283 32300000 X
1317      01 00284 68300000 X
1318      01 00285 72460000 N
1319      01 00286 35400000 X
1320      01 00287 69300289
1321      01 00288 35400000 X
1322      01 00289 33F00000 X
1323      01 0028A 33100000 X
1324      01 0028B 25300009 A
1325      01 0028C 20100000 A
1326      01 0028D 69100000 X
1327      01 0028E 33F00000 X
    
```

```

PAGE
NAME: TISTLPP
PURPOSE: TO STEAL A PHYSICAL PAGE FROM THE MONITOR
FREE PAGE LIST WITHOUT ASSIGNING IT TO A
USER.
DESCRIPTION: THIS IS A CAREFULLY CONTROLLED METHOD OF
OBTAINING PHYSICAL MEMORY FOR USE BY MONITOR
FUNCTIONS FOR SHORT PERIODS OF TIME. IT KEEPS
TRACK OF THE NUMBER OF STOLEN PAGES TO ENSURE
AGAINST DEADLOCKS.
DEF TISTLPP,T;RSPP,T;RSPPPEA,INCSTL
REF S;ACORE,S;STL#,S;STLC,SL;STLM,SL;RSVP
REF ENBSR4
    
```

```

TISTLPP EQU $
DISABLE
LW,1 S;STL#
SW,1 SLIRSV
BLZ STL
CW,1 S;STLC
BL STL
ENABLE
LI,3 *1
B *11
STL EQU $
LW,3 M;FPPH
BEZ ENBSR4
LOAD,4 MXIPPUT,3
STW,4 M;FPPH
BNEZ $+2
STW,4 M;FPP
MTW,-1 M;FPPC
MTW,1 S;STL#
SLS,3 9
AI,1 0
BLZ ENBSR4
MTW,-1 S;ACORE
    
```

```

STEAL A PHYSICAL PAGE
;
THIS ROUTINE IS CALLED BY THOSE MON
FUNCTIONS THAT WISH TO BORROW, PHY
PAGES FOR SHORT PERIODS OF TIME BUT
RELEASE THEM WHEN THEY ARE NEEDED T
USERS. THE RELEVANT DATA CELLS ARE
;
SLIRSV * THE NUMBER OF PAGES HELD
FOR THOSE WHO STEAL BUT C
RETURN THEM ON DEMAND. T
PAGES ARE NOT INCLUDED IN
;
SL;STLM * THE MAXIMUM NUMBER OF PAG
MAY BE STOLEN BEYOND SL;R
;
S;STLC * THE NUMBER OF PAGES CURRE
'STEALABLE' BEYOND SL;RSV
NUMBER GOES NEGATIVE WHEN
SWAP SCHEDULER NEEDS THE
PAGES.
;
S;STL# * THE NUMBER OF STOLEN PAGES
    
```

H01 17:45 SEP 08 175
1328 01 0028F 48000000 X

B

ENBSR4

(INCLUDES SC:RSVP)

```

1329
1330
1331
1332
1333
1334
1335
1336
1337 01 00290 3230000E A
1338      01 00291
1339 01 00291 213FFFFFF N
1340 01 00292 6820026D
1341 01 00293 25300077 A
1342 01 00294 22400000 A
1343 01 00295 75460000 N
1344 01 00296 6D000037 A
1345 01 00297 32400000 X
1346 01 00298 6930029B
1347 01 00299 35300000 X
1348 01 0029A 6800029C
1349 01 0029B 75380000 N
1350 01 0029C 35300000 X
1351 01 0029D 33100000 X
1352 01 0029E 32300000 X
1353 01 0029F 203FFFFFF A
1354 01 002A0 35300000 X
1355 01 002A1 38300000 X
1356 01 002A2 69100000 X
1357 01 002A3 33100000 X
1358 01 002A4 32300000 X
1359      01 002A5
1360 01 002A5 31300000 X
1361 01 002A6 68100000 X
1362 01 002A7 33100000 X
1363 01 002A8 68000000 X
1364
1365
    
```

```

PAGE
NAME: T;RSPP,T;RSPPEA
PURPOSE: TO RETURN STOLEN PAGES TO THE MONITOR.
DESCRIPTION: THIS ROUTINE ADDS A SPECIFIED PAGE
TO THE MONITOR FREE PAGE LIST AND UPDATES THE
STOLEN PAGE CONTROL COUNTERS.
*
*
*
T;RSPPEA LW,3 14
T;RSPP EQU $
CI,3 TOPROOT=1
BLE TFPP1
SLS,3 =9
LI,4 0
STORE,4 MX,PPUT,3
DISABLE
LW,4 MIFPPT
BNEZ $+3
STW,3 MIFPPH
B $+2
STORE,3 MXIPPUT,4
STW,3 MIFPPT
MTW,1 MIFPFC
LW,3 $I$STL#
AI,3 =1
STW,3 $I$STL#
SW,3 SLIRSVF
BLZ ENBSR4
MTW,1 $I$CORE
LW,3 $I$STLC
INCSTL EQU $
CW,3 $I$STLM
BGE ENBSR4
MTW,1 $I$STLC
B ENBSR4
*****
*****
    
```

```

RETURN A STOLEN PP (ADDR IN R14 FOR
RETURN A STOLEN PP (ADDR IN R3)
LAST MONITOR LOCATION
*****
$;ACORE * IS THE CURRENT AVAILABLE
CORE. IT IS LESS THAN SL;
THE NUMBER OF PAGES STOLE
SLIRSVF.
;
;
ONLY ROUTINES ABLE TO GIVE BACK THE
STOLEN PAGES SHOULD CALL TISTLPP UN
THEIR PAGES HAVE BEEN RESERVED IN S
;
;
UPON RETURN FROM TISTLPP
;
R3 = -1 -> NO PAGES MAY BE STOLEN
TIME. (THE SWAPPER IS S
;
;
R3 = 0 -> THERE ARE CURRENTLY NO
MONITOR PAGES * TRY AGA
LATER.
;
R3 > 0 -> HERE'S THE WA OF YOUR P
;
(INCSTL IS CALLED BY CLOCK4)
;
    
```



```

1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415          01 002A9
1416 01 002A9  32200000 X
1417 01 002AA  72240000 X
1418
1419          01 002AB
1420 01 002AB  22F00002 A
1421 01 002AC  33000000 X
1422 01 002AD  693002FB
1423
1424
1425
1426 01 002AE  72E40000 X
1427 01 002AF  31140000 X
1428 01 002B0  682002B4
1429 01 002B1  22000000 A
1430 01 002B2  3600000E A
1431 01 002B3  35000001 A
1432 01 002B4  220FFFFFF A
1433 01 002B5  73040000 X
1434 01 002B6  683002BB
1435 01 002B7  3200051C
1436 01 002B8  21100001 A
1437 01 002B9  684002BB
1438 01 002BA  25000201 A
1439 01 002BB  35000006 A
    
```

```

*      MIGAM3      =1      =2      =3      =4
*      MIGAM4      6      3      3      3
*      MIGPT       41      6      6      6
*      MISWAPS     0      1      2      3
*      MIDWT      41      12     24     48
*
*THE ABOVE ARE ALL BYTE VALUES
    
```

```

* MISGP CONTAINS THE ADDRESS OF THE GRANULE ALLOCATION TABLE.
* MISBAND CONTAINS THE TRACK/SECTOR ORIGIN OF THE SWAP AREA ON DISC
    
```

```

*****
*****
    
```

```

TISGA EQU $
      LW,2 SICUN
      LB,2 UBISWAP1,2
* THIS ENTRY FOR GETTING JIT FIRST TIME
TISGAJIT EQU $
      LI,15 2 GET DUMMY ADDRESS, THEN
      MTW,0 S;DP TEST FOR DISK PACK SWAPPING
      BNEZ SGA10 IF YES DON'T ALLOCATE
    
```

```

*****
*MAKE GRANULE POSITION MODULAR (SECTORS/TRACK)
*****
    
```

```

      LB,14 MBISPT,2 *
      CW,1 MIGASLIM,2 *COMPARE REQUEST AGAINST
      BLE $+4 *SECTOR/TRACK.
      LI,0 0 *GET SECTOR NR, MOD TRACK SIZE
      DW,0 14 *
      STW,0 1 *
      LI,0 =1 *BUILD MASK FOR SEARCH
      MTB,0 MBIGAM7,2 *
      BEZ $+5 *
      LW,0 =X'55555555; *3214 (NSRAD) MASK
      CI,1 1 *IF REQUEST IS FOR AN ODD
      BAZ $+2 *SECTOR MAKE THE MASK ODD.
      SC,S,0 1 *
      STW,0 6 *SAVE FOR LOOP LATER
    
```

```

1440 01 002BC 2510007F A
1441
1442
1443
1444      01 002BD
1445 01 002BD 72E40000 X
1446 01 002BE A510000E A
1447      01 002BF
1448 01 002BF 72E40000 X
1449 01 002C0 32F40000 X
1450
1451
1452
1453
1454      01 002C1
1455 01 002C1 92C2000F A
1456 01 002C2 693002CC
1457 01 002C3 20100001 A
1458 01 002C4 71140000 X
1459 01 002C5 691002C7
1460 01 002C6 22100000 A
1461      01 002C7
1462 01 002C7 64E002C1
1463 01 002C8 25000201 A
1464 01 002C9 31000006 A
1465 01 002CA 693002BF
1466 01 002CB 68000325
1467
1468
1469
1470      01 002CC
1471 01 002CC 25100001 A
1472 01 002CD 31C00000 A
1473 01 002CE 694002D6
1474 01 002CF 20100001 A
1475 01 002D0 31D00000 A
1476 01 002D1 694002D6

```

```

SLS,1      #1      *MAKE IT A GRANULE NR.
*****
*INITIALIZE FOR TABLE SEARCH
*****
SGA1      EQU      $
           LB,14    MISWAPS,2      *GET SHIFT VALUE
           SLS,1    #14           *GIVES DBLWD INDEX OF GRAN REQUESTED
SGA12     EQU      $
           LB,14    MIDWT,2       *GET NO. OF DBLWDS IN GRANULE TABLE
           LW,15    MISGP,2      *GET ADDRESS OF GRANULE TABLE
*****
* SEARCH TABLE FOR AVAILABLE TRACK (0 BIT MEANS NOT AVAILABLE, 1 BIT
* MEANS AVAILABLE). START WITH THE GRANULE REQUESTED.
*****
SGA2      EQU      $
           LD,12    #15,1       *IS THERE A TRACK AVAILABLE IN DBLW
           BNE     SGA4         *IF YES, SKIP
           AI,1     1           *NO. INCREMENT DBLWD INDEX.
           CB,1     MIDWT,2     *ARE WE AT THE END OF THE TABLE
           BL      SGA3         *IF NOT, SKIP
           LI,1     0           *YES. RESET DBLWD INDEX.
SGA3      EQU      $
           BDR,14   SGA2       *LOOP IF ENTIRE TABLE NOT EXAMINED
           SCS,0    1           *CHECK FOR ANOTHER PASS
           CW,0     6           *
           BNE     SGA12        *YES, DO IT
           B       SGRBAD      *TABLE FULL
*****
*EXAMINE THE DOUBLEWORD TABLE ENTRY TO GET THE NON-ZERO WORD
*****
SGA4      EQU      $
           SLS,1    1           *MAKE A WORD INDEX OF THE DBLWD IND
           CW,12    0           *DOES THE FIRST HALF HAVE A FREE TR
           BANZ    SGA5         *IF YES, SKIP
           AI,1     1           *NO. INCREMENT THE WORD INDEX
           CW,13    0           *HOW ABOUT THE SECOND HALF
           BANZ    SGA5         *

```

H01 17:45 SEP 08, '75

1477 01 002D2 2510007F A
 1478 01 002D3 680002C3
 1479 01 002D4 C7D2000F A
 1480 01 002D5 680002D2
 1481 01 002D6 32C00000 A
 1482 01 002D7 22D00000 A
 1483 01 002D8 C6D2000F A
 1484 01 002D9 4BC0000D A
 1485 01 002DA 683002D4
 1486 01 002DB 48D0000C A
 1487 01 002DC C7D2000F A
 1488
 1489
 1490
 1491
 1492 01 002DD
 1493 01 002DD 3520000E A
 1494 01 002DE 22200005 A
 1495 01 002DF 22F00000 A
 1496 01 002E0 3AD0000C A
 1497 01 002E1 4BD0000C A
 1498 01 002F2
 1499 01 002E2 31D40010
 1500 01 002E3 694002E5
 1501 01 002E4 50F4000E
 1502 01 002E5
 1503 01 002E5 642002E2
 1504 01 002E6 48D0000C A
 1505 01 002E7 3220000E A
 1506 01 002E8 32E40000 X
 1507 01 002E9 C7D2000E A
 1508
 1509
 1510
 1511
 1512
 1513 01 002EA 72E40000 X

SLS,1 *1
 B SGA2,2
 SGA7 STS,13 *15,1
 B *3
 SGA5 LW,12 0
 LI,13 0
 XW,13 *15,1
 AND,12 13
 BEZ SGA7
 EBR,13 12
 STS,13 *15,1

 SGA6 EQU \$
 STW,2 14
 LI,2 5
 LI,15 0
 LCW,13 12
 AND,13 12
 SGA8 EQU \$
 CW,13 MASKHAF,2
 BCS,4 SGA9
 AH,15 POSITION,2
 SGA9 EQU \$
 BDR,2 SGA8
 EBR,13 12
 LW,2 14
 LW,14 MISGP,2
 STS,13 *14,1
 *
 *

 LB,14 M;GAM2,2

*TRACK IS GONE, RESTORE THE DBLWD
 *INDEX AND CONTINUE LOOPING
 *RESTORE TABLE AND CONTINUE
 *SEARCHING
 *GET MASK
 *
 *GET TABLE WORD AND BLOCK IT
 *FROM REENTRANCE.
 *OUR BIT WAS GONE.
 *RESET ANYTHING WE CAN USE AND
 *PUT THE REST BACK.

 *COMPUTE THE BIT POSITION OF THE LEAST SIGNIFICANT 1-BIT IN THE TABLE
 *
 *

 *SAVE SWAP INDEX
 *SET LOOP INDEX
 *CLEAR BIT POSITION VALUE
 *SET ALL BITS OF GRANULE ALLOCATION
 WORD TO ZERO EXCEPT LEAST SIG BIT
 *IS BIT IN BOTTOM HALF OF DIVISION
 *IF YES, SKIP
 *NO, ADD NO. OF BITS IN DIVISION
 *LOOP UNTIL CONVERSION COMPLETE
 *RESET FOUND BIT
 *BRING BACK SWAP INDEX
 *GET GRANULE TABLE ADDRESS
 *MERGE RESULT BACK INTO GRANULE
 ALLOCATION TABLE. MERGING PRESERV
 ANY TRACK BITS SET BY REENTRANCE.

 *COMPUTE THE TRACK/SECTOR ADDRESS OF THE ALLOCATED GRANULE

 *GET MASK

H01 17:45 SEP 08, '75

1514 01 002EB 72C40000 X
 1515 01 002EC 4BE00001 A
 1516 01 002ED A510000C A
 1517 01 002EE 25E00005 A
 1518 01 002EF 09200000 N
 1519 01 002FO 72240000 X
 1520 01 002F1 25E40100 A
 1521 01 002F2 08200000 N
 1522 01 002F3 30F0000E A
 1523 01 002F4 30F00001 A
 1524 01 002F5 25F00001 A
 1525 01 002F6 71F40000 X
 1526 01 002F7 684002F9
 1527 01 002F8 20F00001 A
 1528 01 002F9 72140000 X
 1529 01 002FA 4B10000F A
 1530 01 002FB
 1531 01 002FB 33C40000 X
 1532 01 002FC 68000333

SGA10

LB,12 MIGAM3,2
 AND,14 1
 SLS,1 *12
 SLS,14 5
 PUSH 2
 LB,2 MIGAM4,2
 SLD,14 0,2
 PULL 2
 AW,15 14
 AW,15 1
 SLS,15 1
 CB,15 MBIGAM7,2
 BAZ *+2
 AI,15 1
 LB,1 MBIGAM6,2
 AND,1 15
 EQU \$
 MTW,=4 MIFREE#GRAN,2
 B SGR1

*GET SHIFT VALUE
 *EXTRACT HIGH ORDER TRACK BITS
 *SHIFT GRANULE NO. TO P31
 *SHIFT HIGH ORDER TRACK BITS UP

 *GET SHIFT VALUE
 *POSITION ALL TRACK BITS FOR IO

 *COMBINE HIGH AND LOW ORDER TRACKS
 *ADD GRANULE NO.
 *SHIFT UP TO GIVE TRACK/SECTOR ADDR
 *CHECK FOR ODD TRACK ON
 *DEVICE WITH ODD SECTOR/TRACK
 *AND MAKE SECTOR ADDRESS ODD
 *
 *RETURN SECTOR ADDRESS

 NORMAL EXIT

1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569

PAGE

F NAME: TISGR,TISGRNU
F PURPOSE: TO RELEASE SWAPPER GRANULE.
F DESCRIPTION: THIS ROUTINE RELEASES A SWAP GRANULE BY
F CONVERTING ITS SEEK ADDRESS TO A BIT ADDRESS
F AND SETTING THE SGP TABLE.
F ONLY GRANULES WHICH ARE A MULTIPLE OF FOUR MAY
F BE RELEASED.
F TISGRNU USES A SPECIFIED SWAPPER SGP INSTEAD
F OF THE CURRENT USERS.
F ON PACK SWAPPER SYSTEMS NO ACTION IS PERFORMED.

* INPUT: TRACK/SECTOR ADDRESS OF GRANULE TO BE RELEASED IN REGISTER 15
* OUTPUT: RELEASED GRANULE NO. IN REGISTER 1
* REGISTERS USED: 2, 12, 13
* CC3 SET TO 0 IF GRANULE RELEASED
* CC3 SET TO 1 IF GRANULE ALREADY RELEASED OR IF TRACK/SECTOR
* INPUT LOOKS BAS
* THE FOLLOWING TABLE SHOWS THE VALUES OF CERTAIN SYSGEN DATA ACCORDING
* TO THE TYPE OF RAD USED FOR SWAPPING.

	FULL 7212	1/4 7232	1/2 7232	FULL 7232
MIGAM3	=1	=2	=3	=4
MIGAM5	=7	=4	=4	=4
MIGAM6	127	15	15	15
MIGATLIM	63	127	255	511

*MIGAM3,MIGAM5, MIGAM6 AND M;SWAPS ARE BYTE VALUES

* MISGP CONTAINS THE ADDRESS OF THE GRANULE ALLOCATION TABLE
* MISBAND CONTAINS THE TRACK/SECTOR ORIGIN OF THE SWAP AREA ON DISC

TISGR EQU 8
LW,2 S;CUN *

01 002FD
01 002FD 32200000 X

```

1570 01 002FE 72240000 X
1571      01 002FF
1572
1573 01 002FF 33000000 X
1574 01 00300 69300327
1575
1576
1577
1578
1579
1580 01 00301 4BF00000 X
1581 01 00302 72C40000 X
1582 01 00303 3210000F A
1583 01 00304 A510000C A
1584 01 00305 31140000 X
1585 01 00306 69200325
1586 01 00307 72D40000 X
1587 01 00308 37D00001 A
1588 01 00309 20D00001 A
1589 01 0030A 72C40000 X
1590 01 0030B 4BC0000F A
1591 01 0030C 30D0000C A
1592 01 0030D 71F40000 X
1593 01 0030E 68400310
1594 01 0030F 20C00001 A
1595 01 00310 31C40000 X
1596 01 00311 69200325
1597 01 00312 25D0007F A
1598 01 00313 21D00003 A
1599 01 00314 69400334
1600 01 00315 72C40000 X
1601 01 00316 4B100000 X
1602 01 00317 22D00001 A
1603 01 00318 25D20000 A
1604
1605
1606

```

```

T:SGRNU LB,2 UB:SWAPI,2
          EQU      *
          DEF      T:SGRNU
          MTW,0    SIDP      TEST FOR DISK PACK SWAPPING
          BNEZ     SGR0      IF YES DON'T RELEASE
*****
* POSITION A BIT P0 THROUGH P31 CORRESPONDING TO THE LOW ORDER TRACKS
* 31 THROUGH 0 DERIVED FROM THE FULL TRACK/SECTOR ADDRESS. THIS BIT
* WILL BE MERGED INTO THE PROPER ENTRY IN THE GRANULE ALLOCATION TABLE.
*****
          AND,15    MINUS2    *FORCE EVEN SECTOR NUMBER
          LB,12     M:GAM5,2  *GET SHIFT VALUE
          LW,1      15        *
          SLS,1     *12       *SHIFT TRACK TO P31
          CW,1      M:GATLIM,2 *COMPARE TRACK AGAINST UPPER LIMIT
          BG        SGRBAD    *EXIT IF OVER UPPER LIMIT
          LB,13     MB:SPT,2  *#SECTORS/TRACK
          MW,13     1         * X TRACK NR.
          AT,13     1         *
          LB,12     M:GAM6,2  *
          AND,12    15        *SECTOR NUMBER
          AW,13     12        *
          CB,15     MB:GAM7,2 *
          BAZ      *+2       *
          AT,12     1         *INC SECTOR NR. IF ODD TRK
          CW,12     M:GASLIM,2 *COMPARE SECTOR AGAINST UPPER LIMIT
          BG        SGRBAD    *EXIT IF OVER UPPER LIMIT.
          SLS,13    *1       *GRANULE NR RELATIVE BEGIN RAD
          CI,13     3         *
          BANZ     SGR1+1    *EXIT IF NOT A MULTIPLE OF FOUR.
          LB,12     M:GAM5,2  *GET SHIFT VALUE.
          AND,1     M5       *EXTRACT TRACK NO. MODULO 32
          LI,13     1         *POSITION A 1 BIT CORRESPONDING
          SLS,13    0,1      * TO THE LOW ORDER TRACK NO.
*****
* COMPUTE THE GRANULE TABLE INDEX AND SET THE GRANULE BIT
*****

```

H01 17:45 SEP 08, 175

```

1607 01 00319 20CFFFFB A
1608 01 0031A 3210000F A
1609 01 0031B A510000C A
1610 01 0031C 4610000F A
1611 01 0031D 72C40000 X
1612 01 0031E 4B10000C A
1613 01 0031F 72C40000 X
1614 01 00320 A510000C A
1615 01 00321 3010000F A
1616
1617 01 00322 32C40000 X
1618 01 00323 B1D2000C A
1619 01 00324 68400327
1620 01 00325 02200010 A
1621 01 00326 F800000B A
1622 01 00327
1623 01 00327 33440000 X
1624 01 00328 33000000 X
1625 01 00329 6930032D
1626 01 0032A C7D2000C A
1627 01 0032B 72140000 X
1628 01 0032C 4B10000F A
1629 01 0032D
1630 01 0032D 02200000 A
      01 0032E 05000000 N
1631 01 0032F 22600000 N
1632 01 00330 6AB00000 X
1633 01 00331 02200000 A
      01 00332 0A000000 N
1634 01 00333
1635 01 00333 02200000 A
1636 01 00334 F800000B A

```

```

AI,12 *5
LW,1 15
SLS,1 *12
XW,1 15
LB,12 M1GAM6,2
AND,1 12
LB,12 M1SWAPS,2
SLS,1 *12
AW,1 15
*
LW,12 M1SGP,2
CW,13 *12,1
BAZ SGR0
SGRBAD LCI 1
B *11
SGR0 EQU $
MTW,4 M1FREE#GRAN,2
MTW,0 SIDP
BNEZ SGR0A
STS,13 *12,1
LB,1 M1BIGAM6,2
AND,1 15
SGR0A EQU $
PUSH 16,0
*
LI,6 E1DPA
BAL,11 TIRE
PULL 16,0
*
SGR1 EQU $
LCI 0
B *11

```

```

*INCREASE THE SHIFT VALUE
*
*SHIFT THE HIGH ORDER TRACKS TO P31
*
*GET EXTRACT MASK
*EXTRACT THE SECTOR NO.
*GET SHIFT VALUE
*CONVERT SECTOR NO. TO GRANULE INDE
*ADD TRACK GROUP NO. THE RESULT IS
  THE GRANULE TABLE WORD INDEX
*GET THE GRANULE TABLE ADDRESS
*IS THE GRANULE ALREADY FREE
*SKIP IF GRANULE STILL IN USE
*SET CC4
*EXIT
*
TEST FOR DISK PACK SWAPPING
  YES
*MERGE BIT INTO GRANULE TABLE
RETURN SECTOR ADDRESS RELEASED.
*
*RESET CC3
*EXIT

```


1637
 1638 01 00335 22400003 A
 1639 01 00336 22200221
 1640
 1641
 1642
 1643
 1644
 1645
 1646
 1647
 1648
 1649
 1650
 1651
 1652
 1653
 1654
 1655
 1656
 1657
 1658
 1659
 1660
 1661
 1662
 1663
 1664
 1665
 1666
 1667
 1668
 1669
 1670
 1671
 1672

PAGE
 LI,4 3 JAV AC
 LI,2 RVP8
 F NAME: TISAC,TISACP1
 F PURPOSE: TO SET AN ACCESS PROTECTION CODE IN J:JAC F
 F FOR A SPECIFIED VIRTUAL PAGE.

* THE VP# IS DIVIDED BY 4 TO OBTAIN A DISPLACEMENT OF THE APPROX
 * BYTE IN THE IMAGE AND THE POSITION OF THE DOUBLE BIT TO BE SET
 * WITHIN THAT BYTE. THE POS OF THE DBL BIT IS USED AS AN INDEX
 * TO GET THE APPROX MASK (1) CONTAINING ONES IN THAT DBL BIT POS.
 * THE AC TO SET IS USED AS AN INDEX TO GET A BYTE (2) CONTAINING
 * THE AC IN ALL DBL BIT POS. THE BYTE IS PULLED FROM THE IMAGE (3).
 * THE AC IS SET INTO THE BYTE BY A SELECTIVE STORE (4) AND THE
 * UPDATED BYTE IS STORED BACK INTO THE IMAGE (5).

FOLLOWING EXAMPLE SHOWS RESULTS OF SETTING 01 AC IN VP#10 MOD10

```

.....
REG8 | 01 01 01 01 | (2)
.....
REG9 | 00 00 11 00 | (1)
.....

```

||
 STORE VV SELECTIVE (4)

```

.....
| -> REG15 | XX XX 01 XX |
.....
| (3) STORE | (5)
.....
|

```

```

.....
| | | XX XX 01 XX |
.....

```

WORD IN IMAGE CONTAINING AC TO CHANGE

```

1673
1674
1675
1676
1677
1678
1679
1680      01 00337
1681 01 00337 20280000 A
1682 01 00338
1683 01 00338 4B400000 X
1684 01 00339 22300003 A
1685 01 0033A 4B300007 A
1686 01 0033B 72D60016
1687 01 0033C 32300007 A
1688 01 0033D 2530007E A
1689 01 0033E 72F60000 X
1690 01 0033F 65200358
1691 01 00340 223FFFFC A
1692 01 00341 72C60018
1693 01 00342 45C0000F A
1694 01 00343 68300345
1695 01 00344 65300341
1696 01 00345 71C80017
1697 01 00346 68200356
1698 01 00347 223FFFFB A
1699 01 00348 19760000 F
1700 01 00349 68900351
1701 01 0034A 65300348
1702 01 0034B 223FFFFE A
1703 01 0034C 31700000 X
1704 01 0034D 68200351
1705 01 0034E 31700000 X
1706 01 0034F 69200351
1707 01 00350 223FFFFD A
1708 01 00351 20300005 A
1709 01 00352 72360000
    
```

```

PAGE
* 2 = LINK REGISTER WHICH IS ALTERED BY BIR AT SAC2=2
* I 7 = VP#
* 12 = JAC ADR (IMAGE)
* I 4 = AC FOR JAC (IN LOW 2 BITS)
* 3 = ADR INDEXING
* 15 = AC IMAGE BYTE
T;SAC EQU $
AI,2 X'800001 NOT SMP CALL
T;SACP1 RES 0 T;SMP ENTRY POINT
AND,4 M2
LI,3 3
AND,3 7 DBL BIT POSN IN BYTE
LB,13 SACBP,3 MASK FOR DBL BIT POS IN BYTE
LW,3 7
SLS,3 =2 BYTE DISP
LB,15 J;JAC,3 APPROPRIATE BYTE FROM IMAGE
BIR,2 SAC4 BRANCH IF NOT SMP
LI,3 =4
SAC2 LB,12 SACACC+1,3 GET AN AC IN ALL BITS OF BYTE TIL
CS,12 15 ONE COMPARES WITH AC IN USERIMAGE
BE $+2
BIR,3 SAC2
CB,12 SACACC,4 COMP CANDIDATE WITH EXISTING
BLE SAC4M GO SET IT
LI,3 =5
CLM,7 WA(DA(J;P;LL)+5),3 DETERMINE WHICH AREA VP IS IN
BCR,9 SAC3
BIR,3 =2
LI,3 =2
CW,7 J;P;UL
BLE SAC3
CW,7 J;E;UP
BG SAC3
LI,3 =3
SAC3 AI,3 5 CONVERT INDEX
LB,3 UACT,3
    
```

H01 17:45 SEP 08, 1975

1710	01	00353	72360017
1711	01	00354	71380017
1712	01	00355	6920035B
1713	01	00356	32300007 A
1714	01	00357	2530007E A
1715	01	00358	72C80017
1716	01	00359	47C0000F A
1717	01	0035A	75F60000 X
1718	01	0035B	6805FFFF A

SAC4M
SAC4
SAC6

LB,3	SACACC,3
CB,3	SACACC,4
BG	SAC6
LW,3	7
SLS,3	#2
LB,12	SACACC,4
STS,12	15
STB,15	JIJAC,3
B	#1,2

GET AREA AC INTO ALL OF BYTE
 COMP CAND WITH MIN FOR AREA
 DONIT SET

 BYTE DISP
 NEW AC
 STORE AC INTO BYTE
 PUT BYTE INTO IMAGE
 RETURN = BIR HAS INCREMENTED

1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755

```

PAGE
* SET UP AND EXECUTE MMC FOR MAP & AC
*
* ENTER T: SXMAP OR T: SXAC
* EACH USES T: SMMC TO SET UP REGS FOR MMC INST
* THEN EXECUTES APPROP MMC INST
*
* 12 = IMAGE ADR      A
* 14 = # OF PGS      N
* 15 = STARTING VP #  P
*
* 11 = EXIT
* 11 = EXIT
*
* 13 = SUBRY LINK
* 3  = INST CODE
* 4  = SHIFT CODE
*
* 14#                A+(P/X)
* 0                    15
*                    IMAGE ADDRESS
*
* 15#                ((N-1+P)/X)-(P/X)+1        (S/X)*X
* 0                    7                15        22
*                    COUNT                START CONTROL
*
*
* REGISTERS 4,12,14,15 GET (SOMEWHAT PREDICTABLY) CLOBBERED.
*
* Rvp91  LI,11  GVPB
*F*      NAME:  T: SXACM3
*F*      PURPOSE: TO LOAD THE ACCESS PROTECTION CODE FOR A
*F*      PAGE:
* T: SXACM3 LI,12  J: JAC
*          LI,14  1
*          LW,15  7
*F*      NAME:  T: SXAC
*F*      PURPOSE: TO LOAD THE ACCESS PROTECTION REGISTERS FROM
*F*      THE USERS IMAGE IN J: JAC, STARTING AT A
*F*      SPECIFIED PAGE FOR A SPECIFIED NUMBER OF

```

01 0035C 22B001AE
01 0035D 22C00000 N
01 0035E 22E00001 A
01 0035F 22F00007 A

```

1756
1757      01 00360
1758 01 00360 22300001 A
1759 01 00361 224FFFFC A
1760      01 00362
1761      01 0036E
1762      01 00362
1763 01 00362 20FFFFFF A
1764 01 00363 30E0000F A
1765 01 00364 25E80100 A
1766 01 00365 38E0000F A
1767 01 00366 20E00001 A
1768 01 00367 30C0000F A
1769 01 00368 3A400004 A
1770 01 00369 25F80009 A
1771 01 0036A 75E0000F A
1772 01 0036B 32E00000 X
1773
1774 01 0036C 4BE0000C A
1775 01 0036D 67060376
1776
1777 01 0036E 32400000 X
1778 01 0036F 52F80000 X
1779 01 00370 21F01000 A
1780 01 00371 F840000B A
1781
1782
1783 01 00372 22E00018
1784 01 00373 32F0001B
1785 01 00374 6FE40000 A
1786
1787
1788
1789
1790
1791
1792
    
```

```

*F*
T: SXAC EQU $
LI,3 1 INST CODE
LI,4 4 SHIFT CODE
MMCC EQU $
GVPA EQU T: SXAC*2
T: SMMC EQU $
AI,14 1 14= N=1
AW,14 15 +P
SLD,14 0,4 /X 15= P/X
SW,14 15 *(P/X)
AI,14 1 +1
AW,12 15 12= A+(P/X)
LCW,4 4
SLS,15 9,4 15= (P/X)*X SHIFTED 9
STB,14 15 15= ((N=1+P)/X)*(P/X)+1 (P/X)*X
LW,14 M22
REF M22
AND,14 12
EXU MMCI,3
* DOES SPECIAL PROCESSOR WANT SJAC
LW,4 S: CUN
LH,15 UH: FLG,4
CI,15 SJAC
BAZ *11 NOPE, EXIT
* LOAD CANNED AC FOR SPECIAL...
* PROCESSOR JIT AC (AND BUFFER CONTEXT AC).
LI,14 SPJITAC 'CANNED IMAGE' ADDR.
LW,15 SPJACTRL COUNT AND START STUFF
LPC,14 0 ACCESS TO MACHINE INTERNALS
*
* NOTE:
* ASSUME 'SJAC PEOPLE' ARE SIGNIFICANTLY RARER THAN 'NOT
* SJAC PEOPLE'. IT IS THEN CHEAPER BOTH BY CODE SIZE AND SPEED
* TO CHECK FOR EM AND ALWAYS RELOAD THAN TO CHECK THE AFFECTED
* PAGE AND NOT. SJAC PAGES HAVE AS HI IF NOT HIGHER MM FREQ.
* UENCY THAN NON-SJAC PAGES, TYPICALLY.
    
```

```

1793
1794
1795
1796
1797
1798 01 00375 F800000B A
1799          01 00376
1800 01 00376 6FE80000 N
1801 01 00377 6FE40000 A
1802
1803
1804
1805
1806
1807          01 00378
1808
1809
1810 01 00378 32200004 A
1811 01 00379 72C40000 N
1812 01 0037A 25C00009 A
1813 01 0037B 09B00000 N
1814 01 0037C 32D0000C A
1815 01 0037D 20C00000 N
1816 01 0037E 22E00100 N
1817 01 0037F 22F00000 N
1818 01 00380 6AB00225
1819
1820 01 00381 32C0000D A
1821 01 00382 20C00000 N
1822 01 00383 52040000 X
1823 01 00384 22E00100 N
1824 01 00385 22F00000 N
1825 01 00386 6AB00360
1826
1827
1828          00000000
1829

```

```

*
*      ALSO NOTE: JIJAC DOESN'T REFLECT SJAC, THIS IS CHEAP THOUGH
*      MAYBE NOT EASY TO TROUBLE-SHOOT (I.E. THE TRACKS LEFT ARE NOT
*      'ANLZ'=ABLE).
*
*      B          *11
*      EQU        $
*      LDMAP,14  0
*      LPC,14    0
*F*      NAME     TIXMMC,TIXMMC1
*F*      PURPOSE: TO SETUP THE MAP AND ACCESS PROTECTION REGISTERS
*F*      BEFORE EXECUTING THE USER.
*F*      TIXMMC1 IS ENTERED MAPPED.
*
T: XMMC  EQU      $
*      EXECUTE MMC = FOR MAP & AC BEFORE RUNNING USER
*      SET UP * EXECUTE MAP
*      LW,2      4
*      LOAD,12   Ux:JIT,2
*      SLS,12    9
*      PUSH     11
*      LW,13    12
*      AI,12    JCMAP
*      LI,14    256-JBVVP
*      LI,15    JBVVP
*      BAL,11   T: SXMAP
*      SET UP & EXECUTE APPROP'S AC = JAC OR DAC
*      LW,12    13
*      AI,12    JJAC
*      LH,0     UHIFLG,2
*      LY,14    256-JBVVP
*      LI,15    JBVVP
*      BAL,11   T: SXAC
*      T: SXAC  NOW DOES SJAC STUFF (7 FOR 5 GROSS +2 WORDS,
*      NET THIS UPDATE =13-17*2=-18 WORDS,RJR).
*
*      DB        0          T: SXAC(SJAC)
*F* *      TEST WHETHER SPECIAL PROCESSOR NEEDS JIT ACCESS

```

H01 17145 SEP 08, '75

1830			*S*	CI,0	SJAC	
1831			*S*	BAZ	XMMC5	
1832			*S*	* LOAD UP CANNED	AC WD FOR SPECIAL	PROCESSOR JIT AC
1833			*S*	LI,14	SPJITAC	IMAGE ADR = CANNED IMAGE
1834			*S*	LW,15	SPJACTRL	COUNT AND START CONTROL
1835			*S*	LPC,14	0	
1836				FIN		T: SXAC(SJAC)
1837		01 00387		XMMC5	EQU	*
1838	01 00387	08B00000	N		PULL	11
1839	01 00388	F800000B	A		B	*11
1840					SPACE	5

HO1 17:45 SEP 08, '75

1841				PAGE	
1842				DEF	T:XMCC1
1843	01	00389	09100000 N	PUSH	1
1844	01	0038A	6A100000 X	BAL,1	UNMAP
1845	01	0038B	6AB00378	BAL,11	T:XMCC
1846	01	0038C	6A100000 X	BAL,1	MAP
1847	01	0038D	08100000 N	PULL	1
1848	01	0038E	68020000 A	B	0,1

1849
 1850
 1851
 1852
 1853
 1854
 1855
 1856
 1857
 1858
 1859
 1860
 1861
 1862
 1863 01 0038F
 1864 01 0038F 32100000 X
 1865 01 00390 21100000 N
 1866 01 00391 F9300000B A
 1867 01 00392 32100000 X
 1868 01 00393 52020000 X
 1869 01 00394 72420000 X
 1870 01 00395 21000080 A
 1871 01 00396 6940039B
 1872 01 00397 72420000 X
 1873 01 00398 21000040 A
 1874 01 00399 6840039B
 1875 01 0039A 72420000 X
 1876 01 0039B
 1877 01 0039B 12280000 X
 1878 01 0039C 02200020 A
 1879 01 0039D 2B20000E N
 1880 01 0039E 12E0000A
 1881 01 0039F 6FE40000 A
 1882 01 003A0 F800000B A

PAGE
 NAME: TIPAC
 PURPOSE: TO LOAD THE ACCESS PROTECTION REGISTERS FOR
 SPECIAL SHARED PROCESSOR.
 DESCRIPTION: THE ACCESS IS NOT SET IF THE USER
 HAS EXTENDED MEMORY MODE. OTHERWISE, THE
 ACCESS CODES ARE OBTAINED FROM PIAC AND
 STORED INTO JIJAC. AN LPC IS EXECUTED TO
 LOAD THE REGISTERS.
 *
 0 = USERS FLAGS
 *
 1 = CURRENT USER #
 *
 2 = AC WORD FOR APPRO PROC
 *
 4 = APPRO PROC #
 *
 TIPAC EQU \$
 LW,1 JIJAC
 CI,1 JEUPVP
 BNE *11
 LW,1 SICUN
 LH,0 UHIFLG,1
 LB,4 UB:ACP,1
 CI,0 TIC
 BANZ PAC4
 LB,4 UB:ASP,1
 CI,0 DIC
 BAZ PAC4
 LB,4 UB:DB,1
 PAC4 EQU \$
 LD,2 P:AC,4
 LCI 2
 STM,2 JIJAC+14
 LD,14 PXAC
 LPC,14 0
 B *11
 DONT SET AC IF USER HAS ALL
 OF CORE
 GET COMMAND PROC #
 PROC AC DBWD
 14 IMAGE 15 # OF 2 & START CONTROL

```

1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897 01 003A1 02200070 A
      01 003A2 0B500000 N
1898 01 003A3 225FFFFFF A
1899 01 003A4 32C00006 A
1900 01 003A5 02200020 A
      01 003A6 0B600000 N
1901 01 003A7 35C00009 A
1902 01 003A8 22100000 N
1903 01 003A9 72120000 A
1904 01 003AA 3A100001 A
1905 01 003AB 3010000C A
1906 01 003AC 20100003 A
1907 01 003AD 36100000 X
1908 01 003AE 30C00001 A
1909 01 003AF 25C00001 A
1910 01 003B0 30C00000 X
1911 01 003B1 6A0003BC
1912 01 003B2 32400000 X
1913 01 003B3 72280000 X
1914 01 003B4 31940000 X
1915 01 003B5 691003B9
1916 01 003B6 22600000 N
1917 01 003B7 6AB00000 X
    
```

```

PAGE
NAME: TIGNVNPI
PURPOSE: TO OBTAIN VIRTUAL PAGES WITHOUT ALLOCATING
PHYSICAL.
DESCRIPTION: THIS ROUTINE CALLS TIGNVNPI WITH A
FLAG SET TO ALLOCATE VIRTUAL PAGES, BUILD
COMMAND LIST ENTRY AND GET A SWAP GRANULE
WITHOUT ASSIGNING A PHYSICAL PAGE. INSTEAD,
A SPECIAL MONITOR PAGE (NPMC) IS USED FOR
THE JX:CMAP AND COMMAND LIST ENTRIES.
THIS ROUTINE IS USED BY STEP TO OBTAIN
PAGES WHICH ARE TO BE INITIALIZED WITH
PROCESSOR DATA OR DCB DURING THE INSWAP
PROCESS.
TIGNVNPI PUSH 7,5
      LI,5 1
      LW,12 6
      PUSH 2,6 SAVE # PAGES AND PAGE#
      STW,12 9 #DCB AND DATA PAGES
      LI,1 BA(JB:NRG)
      LB,1 0,1
      LCW,1 1
      AW,1 12 #PAGES=#REMAINING GRANS
      AI,1 3
      DW,1 X4 #SEEKS REQ'D
      AW,12 1
      SLS,12 1
      AW,12 J,CLE NEW CLIST LENGTH
      BAL,0 TIGAJP
      LW,4 SICUN
      LB,2 UBISWAPI,4
      CW,9 M:FREE#GRAN,2
      BL 3+4
      LI,6 E:ND
      BAL,11 T:REG
    
```

H01 17:45 SEP 08, '75

1918	01	003B8	680003B4
1919	01	003B9	02200020 A
	01	003BA	0A600000 N
1920	01	003BB	6800003F

B	8-4
PULL	2,6
B	GNV2

RESTORE # OF PAGES AND PAGE #

```

1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933      01 0038C
1934
1935      01 0038C      21C00000 N
1936      01 0038D      69100404
1937
1938      01 0038E      32D00000 X
1939      01 0038F      69300404
1940      01 003C0      32100000 X
1941      01 003C1      73120000 X
1942      01 003C2      6A20025F
1943      01 003C3      22200000 N
1944      01 003C4      20300000 A
1945      01 003C5      683003D8
1946
1947      01 003C6      75340000 N
1948
1949      01 003C7      22400000 N
1950      01 003C8      72C80000 N
1951      01 003C9      71C00000 N
1952      01 003CA      693003CD
1953      01 003CB      75300000 N
1954      01 003CC      680003D3
1955      01 003CD      72400000 N
1956      01 003CE
1957      01 003CE      71C80000 N
    
```

```

PAGE
*F* NAME: T:GAJP
*F* PURPOSE: TO DETERMINE IF AN AJIT IS REQUIRED AND
              TO ALLOCATE ONE IF SO.
*F* DESCRIPTION: IF THE USER COMMAND LIST CANNOT FIT
*F* INTO THE SPACE IN JIT AN AJIT PAGE IS
*F* OBTAINED AND THE COMMAND LIST IS MOVED INTO
*F* IT. THE JIT DISC ADDRESS IS NOW USED FOR
*F* THE AJIT AND THE SECOND GRANULE'S
*F* SEEK ADDRESS USED FOR THE JIT. (WHEN THE
*F* JIT WAS ALLOCATED IT GOT A GROUP OF
*F* FOUR GRANULES)
T:GAJP EQU *
* TEST WHETHER AJIT NEEDED
      CI,12 JCLL
      BL AJP0
* AJIT PRESENT OR NEEDED
      LW,13 J:AJ
      BNEZ AJP0
      LW,1 S:CUN
      MTB,1 UB:PCT,1
      BAL,2 T:GPP
      LI,2 JAJITVP
      AI,3 0
      BE GAJ6
* PP AVAIL - PUT IN CMAP AND LINK INTO PPUT
      STORE,3 JX:CMAP,2
* LINK INTO PPUT
      LI,4 JJITVP
      LBAD,12 JX:CMAP,4
      COMPARE,12 JX:PPH
      BNE AJP5=1
      STORE,3 JX:PPH
      B AJP6
      LOAD,4 JX:PPH
      EQU $
      COMPARE,12 MX:PPUT,4
      AJIT NEEDED
      UPDATE USERS TOTAL PGS NEEDED CNT
      NO PP AVAIL
      JIT PP
      HEAD OF PP CHAIN
      YES, PUT AJIT AS HEAD
      AND GO TO LINK AJIT TO JIT
    
```

H01 17:45 SEP 08, '75

1958 01 003CF 483003D2
 1959 01 003D0 72480000 N
 1960 01 003D1 680003CE
 1961 01 003D2 75380000 N
 1962 01 003D3 75C60000 N
 1964 01 003D4 73100000 X
 1965
 1966 01 003D5 32F00002 A
 1967 01 003D6 22B003E1
 1968 01 003D7 68000223
 1969
 1970
 1971 01 003D8
 1972 01 003D8 22C00022 A
 1973 01 003D9 75C40000 N
 1974 01 003DA 52C20000 X
 1975 01 003DB 48C0000D
 1976 01 003DC 55C20000 X
 1977 01 003DD 22600000 N
 1978 01 003DE 6AB00000 X
 1979 01 003DF 49C00000 X
 1980 01 003EO 55C20000 X
 1981
 1982 01 003F1
 1983
 1984 01 003E1 22400003 A
 1985 01 003E2 73180000 X
 1986
 1987 01 003E3 72340000 N
 1988 01 003E4 35300000 X
 1989 01 003E5 25300009 A
 1990 01 003E6 35300000 X
 1991
 1992 01 003E7 52E20000 X
 1993 01 003E8 55E20000 X
 1994 01 003E9 72420000 X

BE \$+3 FOUND PP END
 LOAD,4 MXIPPUT,4
 B AJP5
 STORE,3 MXIPPUT,4 LINK LAST PP TO AJIT PP
 AJP6 EQU \$
 STORE,12 MXIPPUT,3 LINK AJIT PP TO JIT PP
 MTB,1 JBIPPC UPDATE PP CNT
 * DO MMC ON MAP IN ORDER TO ACCESS AJIT PG
 LW,15 2 AJIT VP #
 LI,11 GAJB
 B TISXMAMP2
 * NO PP AVAIL * DO TIREG WHICH RESULTS IN EXECUTING SWAP LOGIC,
 * WHICH GETS PP, LINKS TO PPUT & PUTS PP IN CMAP * WHEN PP AVAIL.
 GAJB EQU \$
 LI,12 NPMC
 STORE,12 JXICMAP,2
 LH,12 UHIFLG,1
 AND,12 NREADY
 STH,12 UHIFLG,1
 LI,6 E,NC
 BAL,11 T,REG REPORT NO CORE AND GIVE UP
 BR,12 X1
 STH,12 UHIFLG,1
 * FOLLOWING IS COMMON TO BOTH * PP & NO PP
 GAJB EQU \$
 * INCR CONTEXT AREA CNT
 LI,4 3 CONTEXT
 MTB,1 JBIPCP,4
 * SET IN JIT * JIAJ (AJIT PP #) & JICLPA (PHY ADR OF CL)
 LOAD,3 JXICMAP,2
 STW,3 JIAJ
 SLS,3 9
 STW,3 JICLPA
 * PUT JIT ON 2ND GRANULE AND AJIT ON 1ST OF GROUP.
 LH,14 UH:AJIT,1
 STH,14 UH:AJIT,1
 LB,4 UB:SWAPI,1

```

1995 01 003EA 20E00002 A
1996 01 003EB 72F80000 X
1997 01 003EC 45E80000 X
1998 01 003ED 682003EF
1999 01 003EE 30E80000 X
2000 01 003EF 55E20000 X
2001
2002 01 003F0 02200040 A
2003 01 003F1 2AC00000 X
2004 01 003F2 4BE00000 X
2005 01 003F3 224FFF01 A
2006 01 003F4 15C801FE N
2007 01 003F5 22300004 A
2008 01 003F6 20400001 A
2009 01 003F7 15E801FE N
2010 01 003F8 643003F6
2011 01 003F9 20C00002 A
2012 01 003FA 654003F4
2013
2014 01 003FB 32400000 X
2015 01 003FC 25400002 A
2016 01 003FD 22100000 N
2017 01 003FE 75400001 A
2018 01 003FF 61100000 N
2019 01 00400 214000FF A
2020 01 00401 68200404
2021 01 00402 30100000 X
2022 01 00403 61100000 N
2023 01 00404
2024 01 00404 F8000000 A
    
```

```

AI,14 2
LB,15 MBIGAM6,4
CS,14 MIGASLIM,4
BLE $+2
AW,14 M:ADRINCR,4
STH,14 UH,JIT,1
* FILL UP AJIT WITH CLIST
LCI 4
LM,12 JICL
AND,14 YFF
LI,4 =255
AJP9 STD,12 JIAJIT+510,4 STORE SEEK 10CD
LI,3 4
AI,4 1
STD,14 JIAJIT+510,4 FOLLOWED BY FOUR WRITE 10CDS.
BDR,3 $=2
AI,12 2 INC PTR TO JHIDA
BTR,4 AJP9
* MOVE OLD CLIST FROM JIT TO AJIT
LW,4 JICLE
SLS,4 2
LI,1 BA(JIAJIT)
STB,4 1 NR. BYTES IN CLIST
MBS,1 BA(JICL)-BA(JIAJIT)
CI,4 X'FF'
BLE $+3
AW,1 YFF
MBS,1 BA(JICL)-BA(JIAJIT)
AJP8 RES 0
B *0 RETURN
    
```

2025			PAGE	
2026		*F*	NAME:	T:XPQVI
2027		*F*	PURPOSE:	TO PLACE A SPECIFIED PHYSICAL PAGE INTO
2028		*F*		A PREVIOUSLY ALLOCATED VIRTUAL PAGE.
2029		*F*	DESCRIPTION:	THE PHYSICAL PAGE IS PLACED IN
2030		*F*		JXICMAP, INSERTED INTO THE USERS PHYSICAL
2031		*F*		PAGE CHAIN AND ITS BYTE ADDRESS PLACED INTO
2032		*F*		THE COMMAND LIST ENTRY.
2033		*F*		THE PREVIOUS PHYSICAL PAGE IS NOT
2034		*F*		RETAINED BY THIS ROUTINE.
2035	00000000		DB	0
2036		*S*	DEF	T:SPQVI
2037		*S* T:SPQVI	BAL,2	T:GPP
2038		*S*	CY,3	0
2039		*S*	BNE	T:XPQVI
2040		*S*	PUSH	6
2041		*S*	PUSH	11
2042		*S*	LI,3	NPMC
2043		*S*	MTB,-1	JB:PPC
2044		*S*	CW,7	J:VLCS
2045		*S*	BG	*+3
2046		*S*	STW,7	J:VLCS
2047		*S*	MTW,-1	J:VLCS
2048		*S*	BAL,11	T:XPQVI
2049		*S* SPQVI	LW,6	SICUN
2050		*S*	LH,11	UH:PLG,6
2051		*S*	OR,11	XPPSWAP
2052		*S*	AND,11	NREADY
2053		*S*	STH,11	UH:PLG,6
2054		*S*	LI,6	EINC
2055		*S*	BAL,11	T:REG
2056		*S*	PULL	11
2057		*S*	PULL	6
2058		*S* *		
2059			FIN	
2060		*S* *		
2061	01 00405 22400000 N	*S* T:XPQVI	LI,4	J:AJIT

H01 17:45 SEP 08, '75

2062 01 00406 32100000 X
 2063 01 00407 69300409
 2064 01 00408 22400000 N
 2065 01 00409 30400000 X
 2066 01 0040A 22200003 A
 2067 01 0040B 22100000 N
 2068 01 0040C 204FFFFE A
 2069 01 0040D F1200004 A
 2070 01 0040E 6820040C
 2071 01 0040F 71720000 X
 2072 01 00410 68300415
 2073 01 00411 72120000 X
 2074 01 00412 6930040C
 2075
 2076
 2077
 2078
 2079
 2080
 2081 01 00413 0F000000 X
 01 00414 00010002 A
 2082 01 00415 753E0000 N
 2083 01 00416 32E00003 A
 2084 01 00417 22F01FFF A
 2085 01 00418 25E0010B A
 2086 01 00419 47E80000 A
 2087 01 0041A 21300022 A
 2088 01 0041B 6930041E
 2089 01 0041C 723E0000 X
 2090 01 0041D 72360000 N
 2091 01 0041E 21100000 N
 2092 01 0041F 69300422
 2093 01 00420 75300000 N
 2094 01 00421 68000424
 2095 01 00422 72120000 N
 2096 01 00423 75320000 N
 2097 00000000

XPGV1

S
 S
 S
 S
 S
 S

LW,1 JIAJ DB WE HAVE AJIT
 BNEZ \$+2
 LI,4 JICL NB
 AW,4 JICLE
 LI,2 3 CODE FOR SEEK
 LI,1 BA(JB;VLH)=BA(JB;LMAP)
 AI,4 #2
 CB,2 #4 CHECK FOR SEEK
 BLE \$+2 YES, SKIP IT
 CB,7 JB;LMAP,1
 BE XPGV2
 LB,1 JB;LMAP,1
 BNEZ XPGV1

SCREECH CODE: 01-02
 REPORTED BY: MM
 MESSAGE: USERS VIRTUAL PAGE CHAIN (JB;LMAP) INCONSISTANT
 REGISTERS: 7=VIRTUAL PAGE NUMBER SOUGHT IN JB;LMAP.
 REMARKS: SCREECH OCCURS IN T:XPV1, CALLED ONLY BY
 T:XBUFF

SCREECH 1,2

XPGV2

STORE,3 JX;CMAP,7
 LW,14 3 GET PAGE NUMBER
 LI,15 XI;FFF; 13 BIT MASK FOR PAGE
 SLD,14 11 POSITION
 STS,14 0,4 AND STORE IN CLIST
 CI,3 NPMC
 BNE \$+3
 LB,3 JB;LMAP,7 SKIP CUR PAGE IN CHAIN
 LOAD,3 JX;CMAP,3 IF WE DON'T HAVE ANOTHER.
 CI,1 BA(JB;VLH)=BA(JB;LMAP)
 BNE \$+3
 STORE,3 JX;PPH NEW HEAD
 B \$+3
 LOAD,1 JX;CMAP,1 PREV PP IN CHAIN
 STORE,3 MX;PPUT,1
 DB 0


```

2098          *S*
2099          *S*
2100
2101 01 00424 721E0000 X
2102 01 00425 68300429
2103 01 00426 72120000 N
2104          01 00427          XPGV3
2105 01 00427 75160000 N
2106 01 00428 F800000B A
2107
2108          01 00429          *
2109 01 00429 32100000 X          XPGV4
2110 01 0042A 69300427
2111 01 0042B 32100000 X
2112 01 0042C 72120000 N
2113 01 0042D 68000427
2114
    
```

```

CI,11      SPGV1
BE          SPGV1
FIN
LB,1       JBILMAP,7
BEZ        XPGV4
LOAD,1     JXICMAP,1
EQU        *
STORE,1    MX:PPUT,3
B          *11
*
EQU        *
LW,1       J:AJ
BNEZ       XPGV3
LW,1       S:GUN
LOAD,1     UX:JIT,1
B          XPGV3
    
```

DONE FOR NOW IF NO PAGE AVAIL

VP OF NEXT PP IN CHAIN
 * NONE * MUST BE LAST
 PP OF NEXT PP IN CHAIN.
 * GAURENTEED NEXT PP # IN 1
 LINK TO NEW PAGE.

* NEXT PPUT IS AJIT OR JIT
 * IS AJIT THERE
 * YEP=USE IT
 * NO GO TO USER TABLE
 * FOR JIT VP #
 * FOR PPUT FLINK

```

2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130      01 0042E
2131 01 0042E 09200000 N
2132 01 0042F 09600000 N
2133 01 00430 72100000 X
2134 01 00431 20100001 N
2135 01 00432 25E00077 A
2136 01 00433 21E00000 N
2137 01 00434 6910046F
2138 01 00435 3A100001 A
2139 01 00436 20100000 N
2140 01 00437 21100009 A
2141 01 00438 6820043A
2142 01 00439 22100009 A
2143 01 0043A 227FFFFFFF N
2144
2145
2146 01 0043B 22C00000 N
2147 01 0043C 71CE0000 N
2148 01 0043D 6830044E
2149 01 0043E 207FFFFFFF A
2150 01 0043F 6410043C
2151      01 00440
    
```

```

PAGE
*
* SPARE BUFFER MEMORY MANGEMENT ROUTINES
*
*F* NAME: TIGBUF
*F* PURPOSE: TO OBTAIN A SPARE BUFFER PAGE AND MAP
*F* IT INTO A SPECIFIED WINDOW PAGE.
*F* DESCRIPTION: AN AVAILABLE VIRTUAL PAGE IS LOCATED
*F* IN THE SPARE BUFFER AREA AND A PHYSICAL PAGE,
*F* COMMAND LIST ENTRY AND SWAP GRANULE ALLOCATED
*F* TO IT BY CALLING TIGVPI.
*F* THE PHYSICAL PAGE THUS OBTAINED IS THEN
*F* MAPPED INTO THE SPECIFIED WINDOW PAGE BY
*F* PLACING ITS PAGE ADDRESS INTO JXICMAP
*F* AND CALLING T: SXMAPM3.
TIGBUF EQU *
      PUSH 2 SAVE RETURN
      PUSH 6 SAVE DCB ADDRESS
      LB,1 JBIFBUL
      AI,1 #JXBUFVP+1 MAX # OF FP00LS
      SLS,14 #9 PAGE # IN WINDOW
      CI,14 JCBVP FP00L OR CP00L WINDOW
      BL GETFP00L FP00L
GETCP00L LCW,1 1
      AI,1 J0VVP=JXBUFVP MAX # OF CP00LS
      CI,1 9 9 IS MOST EVER NEEDED
      BLE #+2
      LI,1 9 SA SET MAX# = 9
      LI,7 J0VVP=1 CP00L SPARE UL
* GET A FREE PAGE FROM THE SPARE BUF AREA; R7=UL OF SPARES TO SEARCH
* R1=# FP00LS/CP00LS ALLOWED
GETSPARE LI,12 FPMC LOOK FOR ONE NEVER USED
      COMPARE,12 JXICMAP,7 IN SPARE BUFFER AREA
      BE GOTSPARE FOUND ONE
      AI,7 #1 LOOK AT NEXT LOWER VIRTUAL PAGE
      BDR,1 #=3 UP TO MAX ALLOWED
N0BUF EQU *
    
```

H01 17145 SEP 08, 175

2152 01 00440 21E00000 N
 2153 01 00441 69200448
 2154 01 00442 09E00000 N
 2155 01 00443 22600000 A
 2156 01 00444 6A000000 X
 2157 01 00445 08E00000 N
 2158 01 00446 6800046F
 2159 01 00447 08E00000 N
 2160 01 00448 22A00000 A
 2161 01 00449 25E00009 A
 2162 01 0044A
 2163 01 0044A 6D000027 A
 2164 01 0044B 08600000 N
 2165 01 0044C 08200000 N
 2166 01 0044D 68040000 A
 2167
 2168 01 0044E
 2169 01 0044E 22500000 A
 2170 01 0044F 09E00000 N
 2171 01 00450 6AB000C0
 2172 01 00451 68000447
 2173 01 00452 08E00000 N
 2174 01 00453 21300000 A
 2175 01 00454 69300457
 2176 01 00455 22600000 N
 2177 01 00456 6AB00000 X
 2178 01 00457 21E00000 N
 2179 01 00458 69100460
 2180 01 00459 33100000 X
 2181 01 0045A 22300000 N
 2182 01 0045B 72A60000 X
 2183 01 0045C 3170000A A
 2184 01 0045D 69200461
 2185 01 0045E 75760000 X
 2186 01 0045F 68000461
 2187 01 00460 33100000 X
 2188 01 00461

CI,14 JSBUF2VP
 BG N0BUF2
 PUSH 14
 LI,6 0
 BAL,0 GETSBUF
 PULL 14
 B GETFP00L
 N0BUF3 PULL 14
 N0BUF2 LI,10 0
 SLS,14 9
 BUFFEXIT EQU *
 ENABLE
 PULL 6
 BUFFEXIT2 PULL 2
 B 0,2
 * GET THE SPARE VIRTUAL PAGE: R7=PAGE # IN SPARES
 GOTSPARE EQU *
 LI,5 0
 PUSH 14
 BAL,11 GVP2
 B N0BUF3
 PULL 14
 CI,3 0
 BNE *+3
 LI,6 EINC
 BAL,11 TIREG
 CI,14 JC0VP
 BL GOTFP00L
 GOTCP00L MTW,1 JBI0BUC
 LI,3 JBCBLI
 LB,10 JIJIT,3
 CW,7 10
 BG MAPSPARE
 STB,7 JIJIT,3
 B MAPSPARE
 GOTFP00L MTW,1 JBI0BUC
 MAPSPARE EQU *

ATTEMPTING TO GET SPEC BUFFER
 N0=INDICATE N0NE AVAILABLE
 SAVE WINDOW VP
 CAUSE SPECIAL TRUNCATION
 GETSBUF WILL SAVE R1
 RESTORE WINDOW VP
 ONE WILL BE AVAILABLE
 NO BUFFER
 RESTORE R14
 RESTORE DCB ADDR
 RESTORE RETURN ADDR
 EXIT
 INDICATE GET PHYSICAL PAGE
 ENTRY POINT TO TIGVPI
 USER MAX PAGES AND WANTS A CP00L
 GOT A PAGE
 YES
 NO CORE EVENT #
 WAS IT A CP00L
 N0,FP00L OR SPECIAL
 INCREMENT # CP00LS USED
 GET CURRENT CP00L LL
 IS THIS A NEW CP00L LI
 NO
 YES=RESET IT
 INCREMENT # FP00LS USED

H01 17145 SEP 08, '75

2189 01 00461 32A00007 A
 2190 01 00462 20A00001 N
 2191 01 00463 723E0000 N
 2192 01 00464 3270000E A
 2193 01 00465 753E0000 N
 2194 01 00466 09E00000 N
 2195 01 00467 6AB00222
 2196 01 00468 08E00000 N
 2197 01 00469 25E00009 A
 2198 01 0046A 6840044A
 2199 01 0046B F230000E A
 2200 01 0046C 22100000 N
 2201 01 0046D 75320000 X
 2202 01 0046E 6800044A
 2203 01 0046F 22700000 N
 2204 01 00470 6D000037 A
 2205 01 00471 727E0000 X
 2206 01 00472 68300476
 2207 01 00473 32300000 X
 2208 01 00474 4730000E A
 2209 01 00475 68000461
 2210 01 00476 6D000027 A
 2211 01 00477 72700000 X
 2212 01 00478 68000438

LW,10 7
 AI,10 #JXBUFVP+1
 LOAD,3 JX:CMAP,7
 LW,7 14
 STORE,3 JX:CMAP,7
 PUSH 14
 BAL,11 T:ISXMARM3
 PULL 14
 SLS,14 9
 BCR,4 BUFFEXIT
 LB,3 *14
 LI,1 JBFBFP
 STB,3 JIJIT,1
 B BUFFEXIT
 GETFP00L EQU \$
 LI,7 JBFBFP
 DISABLE
 LB,7 JIJIT,7
 BEZ NONEFREE
 LW,3 Y8
 STS,3 14
 B MAPSPARE
 NONEFREE EQU \$
 ENABLE
 LB,7 JBIFBUL
 B GETSPARE

CONVERT SPARE BUFFER VP
 TO INDEX FOR CALLING ROUTINE
 PHYS PAGE ASSIGNED TO SPARE BUFF
 WINDOW VP TO R7
 PP TO MAP FOR WINDOW VP
 LOAD THE MAP FOR THE WINDOW
 RESTORE WORD ADDR/TEST POOL FLAG
 BUFFER NOT OBTAINED VIA FREE POOL
 GET NEW FREE POOL HEAD (OR ZERO)
 SET NEW HEAD
 EXIT
 HEAD OF FREE POOLS (SPARE BUF VP)
 NONE IN POOL
 SET FLAG TO INDICATE BUFF HEAD MUST
 BE UPDATED AFTER PAGE IS MAPPED
 POOL SPARE UPPER LIMIT

2215					PAGE		
2216					NAME:	TIZBUF	
2217					PURPOSE:	TO RELEASE ALL FP00L BUFFERS:	
2218					DESCRIPTION:	CALLS RUNDRLS TO UNMAP THE WINDOW	
2219						PAGE AND RELEASE THE SPARE BUFFER PAGES.	
2220		01 00479		TIZBUF	EQU	\$	TOTAL RELEASE OF ALL FP00LS
2221	01	00479	22500000	A	LI,5	0	INDICATE TO RELEASE PP
2222	01	0047A	72100000	X	ZBUF2	LB,1	UL OF FP00LS
2223	01	0047B	21100000	N		CI,1	
2224	01	0047C	68100481			BGE	ZBUF3
2225	01	0047D	22100000	A		LI,1	MORE TO RELEASE
2226	01	0047E	55100000	X		STH,1	INDICATE NO BUFFERS
2227	01	0047F	35100000	X		STW,1	
2228	01	00480	680C0000	A		B	
2229	01	00481	72320000	N	ZBUF3	LOAD,3	PP ASSIGNED TO SPARE
2230	01	00482	21300000	N		CI,3	IS SPARE ALLOCATED
2231	01	00483	68300489			BE	NO-CHECK REST OF SPARE FP00L AREA
2232	01	00484	22200489			LI,2	
2233	01	00485	09200000	N		PUSH	
2234	01	00486	02200050	A		PUSH	
	01	00487	0B600000	N			
2235	01	00488	6A200493			BAL,2	YES-UNMAP WINDOW, RELEASE BUFFER
2236	01	00489	73F00000	X	ZBUF4	MTB,-1	RESET FP00L UL
2237	01	0048A	6800047A			B	
2238							

```

2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254      01 00488
2255 01 0048B 09200000 N
2256 01 0048C 02200050 A
          01 0048D 0B600000 N
2257 01 0048E 3210000E A
2258 01 0048F 19100004
2259 01 00490 689004BD
2260 01 00491 201FFFFFF N
2261      01 00492
2262 01 00492 6AB004D4
2263 01 00493 22C00008 A
2264 01 00494 22700001 N
2265 01 00495 6AB004C9
2266 01 00496 6800049A
2267      01 00497
2268 01 00497 22800000 N
2269 01 00498 75BE0000 N
2270 01 00499 6AB00222
2271 01 0049A 32500005 A
2272 01 0049B 692004B1
2273 01 0049C 691004BA
2274 01 0049D 71100000 X
    
```

```

PAGE
NAME: TIRBUF
PURPOSE: TO RELEASE A SPARE BUFFER TO THE FREE BUFFER
DESCRIPTION: IF THE SPECIFIED BUFFER IS MAPPED
            INTO A WINDOW PAGE IT IS REMOVED BY
            PLACING AN FPMC CONSTANT IN JXICMAP AND
            AND RELOADING THE MEMORY MAP. IF THE
            BUFFER IS TO BE PLACED IN THE FREE BUFFER
            POOL IT IS ADDED TO THE HEAD (JBIFBUL).
            IF IT IS TO BE RELEASED, TIRVPI IS CALLED
            TO RELEASE IT. IF THE PHYSICAL PAGE
            IS NOT TO BE RETURNED TO THE MONITOR A
            FLAG IS SET AND TIRVPI DOES NOT RETURN IT.
*
TIRBUF EQU $
          PUSH 2 SAVE RETURN
          PUSH 5,6 SAVE DCB ADDRESS ETC.

          LW,1 14 WINDOW ADDR GIVEN
          CLM,1 WNDLIMS
          BCR,9 RWINDOW YES
          AI,1 JXBUFVP+1 CONVERT INDEX TO VP
RSPARE EQU $ RELEASE BUFFER GIVEN INDEX
          BAL,11 TESTREL ERROR IF THIS VP IS ALREADY IN POOL
RUNDRLS LI,12 8 # OF WINDOW PAGES
          LI,7 JCBVP+1 WINDOW UPPER LIMIT
          BAL,11 SEARCH SEE IF IT IS NOW MAPPED
          B RLSBUF NO=GO RELEASE
UNMAPBUF EQU $ GET BUFFER OUT OF WINDOW
          LI,11 FPMC WITH FPMC
          STORE,11 JXICMAP,7 RELOAD MAP FOR THAT PAGE
          BAL,11 TISXMAPM3 RELEASE BUFFER AS DICTATED BY R5
RLSBUF LW,5 5 GREATER THAN ZERO=RELEASE TO POOL
          BGZ RLSP00L LESS THAN ZERO=JUST RELEASE WINDOW
          BLZ BUFFXIT3 RELEASING FPOOL
          CB,1 JBIFBUL
    
```

```

2275 01 0049E 682004AC
2276 01 0049F 33F00000 X
2277 01 004A0 22300000 N
2278 01 004A1 71160000 X
2279 01 004A2 693004AD
2280 01 004A3 32200001 A
2281 01 004A4 20200001 A
2282 01 004A5 21200000 N
2283 01 004A6 683004AA
2284 01 004A7 72740000 N
2285 01 004A8 21700000 N
2286 01 004A9 683004A4
2287 01 004AA 75260000 X
2288 01 004AB 680004AD
2289 01 004AC 33F00000 X
2290 01 004AD 32700001 A
2291 01 004AE 6AB001B2
2292 01 004AF 02000000 A
2293 01 004B0 680004BA
2294 01 004B1 22500000 N
2295 01 004B2 6D000037 A
2296 01 004B3 727A0000 X
2297 01 004B4 751A0000 X
2298 01 004B5 72320000 N
2299 01 004B6 0E000002
2300 01 004B7 25300009 A
2301 01 004B8 F5700003 A
2302 01 004B9 6A100000 X
2303 01 004BA 02200050 A
2304 01 004BB 0A600000 N
2305 01 004BC 6800044C
2306
2307 01 004BD 25100077 A
2308 01 004BE 32500005 A
2309 01 004BF 681004C2
2310

```

```

BLE RLSBUF2
MTW,-1 JBI CBUC
LI,3 JBCBLL
CB,1 J;JIT,3
BNE RLSBUF3
LW,2 1
RBUF2 AI,2 1
CI,2 J8VVP
BE RBUF3
LOAD,7 JXICMAP,2
CI,7 FPMC
BE RBUF2
RBUF3 STB,2 J;JIT,3
B RLSBUF3
RLSBUF2 MTW,-1 JBIFBUC
RLSBUF3 LW,7 1
BAL,11 T;RVPI
N8P
B BUFFXIT3
RLSP00L EQU *
LI,5 JBF BFP
DISABLE
LB,7 J;JIT,5
STB,1 J;JIT,5
LOAD,3 JX;CMAP,1
LPSD,0 MMUMI
RLSP2 SLS,3 9
STB,7 *3
BAL,1 MAP
BUFFXIT3 PULL 5,6
B BUFFXIT2
*
RWINDOW EQU *
SLS,1 *9
LW,5 5
BGEZ **3

```

```

YES
NO-DECREMENT # CPOOLS

SPARE VP ; CURRENT LL
ABOVE CURRENT LL-DONT CHANGE LL

LOOK AT NEXT CPOOL SPARE
ARE ALL RELEASED
YES
IS NEXT HIGHER A FREE CPOOL SPARE

YES-LOOK FOR USED CPOOL SPARE

RELEASING FPOOL VP AND PP
SPARE VP TO R7
T;RVPI WILL REACT TO R5 CORRECTLY

EXIT
RELEASE BUFFER TO FREE POOL
GET CURRENT HEAD OF FREE

BUFFER IN POOL (SPARE VP)
RELEASED SPARE TO HEAD (SPARE VP)
PP OF NEW POOL HEAD
DISABLE AND GO UNMAPPED
GET REAL WORD ADDR OF RELEASED BUFF
SET PREV HEAD AS LINK IN NEW HEAD

RELEASE BUFFER GIVEN WINDOW
GET WINDOW VP
IS THIS A CALL TO UNMAP
NO

```

H01 17:45 SEP 08, '75

2311 01 004C0 32700001 A
 2312 01 004C1 68000497
 2313 01 004C2 227FFFFFF N
 2314 01 004C3 22C00014 A
 2315 01 004C4 6AB004C9
 2316 01 004C5 6AB0051A
 2317 01 004C6 46100007 A
 2318 01 004C7 6AB004D4
 2319 01 004C8 68000497

LW,7 1
 B UNMAPRUF
 LI,7 JOVVP=1
 LI,12 20
 BAL,11 SEARCH
 BAL,11 BSCREECH
 XW,1 7
 BAL,11 TESTREL
 B UNMAPRUF

R7 = WINDOW VP
 GO RELEASE IT
 SPARE BUFFER UPPER LIMIT
 # OF PAGES
 FIND MATCH IN SPARES
 BUF MAPPED IN WINDOW BUT NOT IN SPR
 R1=SPARE VP R7=WINDOW VP
 ERROR IF BUFFER IS ALREADY IN POOL
 REMAP WINDOW PAGE-RELEASE BUFFER


```

2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334 01 004C9 72320000 N
2335 01 004CA 21300000 N
2336 01 004CB 693004CD
2337 01 004CC 6800051A
2338 01 004CD 713E0000 N
2339 01 004CE 693004D1
2340 01 004CF 20B00001 A
2341 01 004D0 F800000B A
2342 01 004D1 207FFFFFFF A
2343 01 004D2 64C004CD
2344 01 004D3 F800000B A
    
```

```

PAGE
* ROUTINE TO FIND A PHYSICAL PAGE MATCH IN WINDOW AND SPARES
* GIVEN SPARE FIND MATCH IN WINDOW = GIVEN WINDOW FIND MATCH IN SPARE
* INPUT R1=SPARE VP R1=WINDOW VP
* R7=WINDOW UL R7=SPARE BUF UL
* R12=# OF WINDOWS TO SEARCH R12=# OF SPARES TO SEARCH
*
* OUTPUT R3= PHYS PAGE # R3= PHYS PAGE #
* R7= VP IN WINDOW R7= VP IN SPARE
*
* CALLING SEQUENCE IBAL,11 SEARCH
* NO MATCH
* MATCH
SEARCH EQU *
LOAD,3 JXICMAP,1 GET PP #
CI,3 FPMC DOES ANYONE HAVE IT
BNE SEARCH2 YES
B BSCREECH NO=TRYING TO RELEASE UNALLOC BUFFER
SEARCH2 COMPARE,3 JXICMAP,7 IS THIS A MATCH
BNE SEARCH3 NO
AI,11 1
B *11 MATCH FOUND=EXIT
SEARCH3 AI,7 =1 LOOK AT NEXT LOWER PAGE
BDR,12 SEARCH2 UNTILL ALL HAVE BEEN CHECKED
B *11 NO MATCH=EXIT
    
```

```

2345
2346
2347
2348
2349
2350      01 004D4
2351 01 004D4 22400000 N
2352 01 004D5 72480000 X
2353 01 004D6 F830000B A
2354 01 004D7 22F00000 A
2355 01 004D8 20F00001 A
2356 01 004D9 31400001 A
2357 01 004DA 693004DC
2358 01 004DB 6800051A
2359 01 004DC 72380000 N
2360 01 004DD 09100000 N
2361 01 004DE 6A100000 X
2362 01 004DF 25300009 A
2363 01 004E0 F2400003 A
2364 01 004E1 6A100000 X
2365 01 004E2 08100000 N
2366 01 004E3 21400000 A
2367 01 004E4 693004D8
2368 01 004E5 32500005 A
2369 01 004E6 F820000B A
2370 01 004E7 21F00003 A
2371 01 004E8 F910000B A
2372 01 004E9 22500000 A
2373 01 004EA F800000B A
    
```

```

PAGE
* ROUTINE TO DETERMINE IF BUFFER MAY BE RELEASED
* IF ERRORS FOUND GO TO RECOVERY
* CALLING SEQUENCE: BAL,11 TESTREL R1=VP OF SPARE TO BE RELEASED
*
TESTREL EQU $
          LI,4 JBF8FP
          LB,4 JIJIT,4 CURRENT HEAD OF POOL
          BEZ *11 NONE IN POOL
          LI,15 0 INITIALIZE COUNT OF BUFFERS IN FREE
TESTREL3 AI,15 1 ADD UP # OF BUFFERS IN FREE POOL
          CW,4 1 RELEASED;REQUESTED RELEASE
          BNE *+2 BUFFER IS ALREADY IN POOL
          LOAD,3 JX;CMAP,4 PP OF RELEASED BUFFER
          PUSH 1 WA OF RELEASED BUFFER
          BAL,1 UNMAP GET LINK TO NEXT IN POOL (SPARE VP)
          SLS,3 9
          LB,4 *3
          BAL,1 MAP
          PULL 1 END OF CHAIN
          CI,4 0 NO=CONTINUE CHECKING
          BNE TESTREL3 RELEASING TO POOL
          LW,5 5 NO
          BLEZ *11 HAS FREE POOL THRESHOLD BEEN REACHED
          CI,15 3 NO=RELEASE IT TO POOL
          BL *11 YES=SET TO RELEASE EVERYTHING
          LI,5 0
          B *11
    
```

```

2374
2375
2376
2377
2378
2379
2380
2381
2382
2383      01 004FB
2384 01 004EB 3210000A A
2385 01 004EC 201FFFFFF N
2386 01 004ED 3270000E A
2387 01 004EE 25700077 A
2388 01 004EF 72C20000 N
2389 01 004F0 21C00000 N
2390 01 004F1 6830051A
2391 01 004F2 71CE0000 N
2392 01 004F3 68340000 A
2393 01 004F4 75CE0000 N
2394 01 004F5 6AB00222
2395 01 004F6 68040000 A
    
```

```

PAGE
NAME:    TIMBUF
PURPOSE: TO REMAP A PREVIOUSLY ACQUIRED SPARE
          BUFFER INTO A SPECIFIED WINDOW PAGE.
DESCRIPTION: IF THE PAGE IS ALREADY MAPPED
             NOTHING IS DONE. IF NOT, ITS PHYSICAL
             PAGE ADDRESS IS PLACED IN JXICMAP AT
             THE SPECIFIED WINDOW PAGE AND A CALL
             MADE TO T:ISXMAPM3 TO REMAP IT.
    
```

```

TIMBUF EQU *
        LW,1 10 INDEX TO SPARE
        AI,1 JXBUFVP=1 SPARE VP
        LW,7 14 WINDOW ADDR
        SLS,7 =9 WINDOW VP
        LOAD,12 JXICMAP,1 PP ASSIGNED TO SPARE
        CI,12 FPMC IS THERE A PP
        BE BSCREECH NO-GO TO RECOVERY
        COMPARE,12 JXICMAP,7 IS SPARE CURRENTLY MAPPED INTO WIND
        BE 0,2 YES=EXIT
        STORE,12 JXICMAP,7 PUT PP IN WINDOW VP
        BAL,11 T:ISXMAPM3 LOAD MAP FOR THAT PAGE
        B 0,2 EXIT
    
```

```

2396
2397
2398
2399
2400
2401
2402
2403
2404
2405      01 004F7
2406 01 004F7 3210000A A
2407 01 004F8 201FFFFFF N
2408 01 004F9 227FFFFFF N
2409 01 004FA 22C00006 A
2410 01 004FB 6AB004C9
2411 01 004FC 6AB0051A
2412 01 004FD 3210000E A
2413 01 004FE 25100077 A
2414 01 004FF 75320000 N
2415 01 00500 22300000 N
2416 01 00501 753E0000 N
2417 01 00502 6AB00222
2418 01 00503 32700001 A
2419 01 00504 6AB00222
2420 01 00505 68040000 A
    
```

```

PAGE
NAME: TISBUF
PURPOSE: TO MAP A SPARE BUFFER, CURRENTLY MAPPED
          INTO A DIFFERENT WINDOW PAGE, INTO A
          SPECIFIED WINDOW PAGE.
DESCRIPTION: THE PAGES ARE SWITCHED BY SETTING
            JXICMAP AND CALLING TISXMAPM3. THE PREVIOUS
            JXICMAP ENTRY IS DELETED BY PLACING FPMC
            IN IT.
TISBUF EQU $
        LW,1 10 SPARE BUFFER INDEX
        AI,1 JXBUFVP,1 SPARE BUFFER VP
        LI,7 JCBVP,1 FPOOL WINDOW UL
        LI,12 6 * OF WINDOWS TO SEARCH
        BAL,11 SEARCH FIND WHERE IT IS NOW
        BAL,11 BSCREECH NOT IN FPOOL WINDOW
        LW,1 14 NEW WINDOW TO R1
        SLS,1 =9 NEW WINDOW VP
        STORE,3 JXICMAP,1 PP TO WINDOW CMAP
        LI,3 FPMC
        STORE,3 JXICMAP,7 FPMC TO OLD WINDOW PAGE
        BAL,11 TISXMAPM3 LOAD MAP FOR OLD WINDOW PAGE
        LW,7 1 NEW WINDOW VP
        BAL,11 TISXMAPM3 LOAD MAP FOR NEW WINDOW PAGE
        B 0,2
    
```

```

2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431 01 00506 02200020 A
      01 00507 0B200000 N
2432 01 00508 3210000E A
2433 01 00509 25100077 A
2434 01 0050A 72700000 X
2435 01 0050B 32C00007 A
2436 01 0050C 20C00001 N
2437 01 0050D 6AB004C9
2438 01 0050E 6AB0051A
2439 01 0050F 08300000 N
2440 01 00510 25300077 A
2441 01 00511 72AE0000 N
2442 01 00512 25A00009 A
2443 01 00513 75320000 N
2444 01 00514 09100000 N
2445 01 00515 6AB00405
2446 01 00516 08700000 N
2447 01 00517 6AB00222
2448 01 00518 08200000 N
2449 01 00519 68040000 A
2450
2451
2452
2453
2454
2455
2456
    
```

```

PAGE
NAME: TIXBUF
PURPOSE: TO EXCHANGE A STOLEN PHYSICAL PAGE FOR
          A SPARE BUFFER PAGE ACQUIRED BY TIGBUF.
DESCRIPTION: THIS ROUTINE LOCATES THE SPARE BUFFER
             PAGE TO BE EXCHANGED AND SAVES THE
             PREVIOUS PHYSICAL PAGE ASSIGNED TO IT
             FOR RETURN TO THE CALLER. IT THEN CALLS
             TIXPGVI TO SETUP THE PHYSICAL PAGE CHAINS
             AND COMMAND LIST.
TIXBUF PUSH 2,2 SAVE RETURN AND PP ADDRESS

LW,1 14 WINDOW ADR TO R1
SLS,1 *9 WINDOW VP
LB,7 JBIFBUL FPOOL SPARE UL
LW,12 7
AI,12 *JXBUF,VP+1 # OF SPARES TO SEARCH
BAL,11 SEARCH FIND MATCH
BAL,11 BSCREECH BUF MAPPED IN WINDOW, NOT IN SPARES
PULL 3 PP ADDRESS
SLS,3 *9 PP PAGE
LOAD,10 JXICMAP,7 PP NOW IN SPARE
SLS,10 9 CONVERT TO WINDOW ADDRESS
STORE,3 JXICMAP,1 NEW PP TO WINDOW
PUSH 1 SAVE WINDOW VP
BAL,11 TIXPGVI FIX MAP AND PP LINKS
PULL 7 LOAD MAP FOR WINDOW
BAL,11 TISXMAPM3 LOAD MAP FOR WINDOW
PULL 2
B 0,2
*S* SCREECH CODE; 6B=00
*S* REPORTED BY; MM
*S* MESSAGE: ERROR IN SPARE BUFFER TABLES
*S* REGISTERS: 11=ADDRESS IN BUFFER SUBROUTINE WITHIN MM
      (TIGBUF, TIRBUF, ETC.) WHICH DETECTED THE
      ERROR.
*S*
*S* REMARKS: USUALLY BAD INPUT FROM CALLING ROUTINE.
    
```

H01 17:45 SEP 08, '75

2457 01 0051A

2458 01 0051A 0F000000 X

01 0051B 006B0000 A

2459 0000051C

2460

01 0051C 55555555 A

BSCREECH EQU \$
SCREECH X'6B'

MMSZ EQU \$=MM
END

CONTROL SECTION SUMMARY; 01 0051D PT 0

17:45 SEP 08, '75
* SYMBOL VALUES

AJP6/01 00404
ANSPR6C/00000000
BUFFEXIT/01 0044A
C6RATX/00000001
DIC/00000040
GAJ6/01 003D8
GETSPARE/01 0043B
GNV5/01 0004C
G8TSPARE/01 0044E
GVP6/01 001B1
GVP230/01 000E0
GVP240/01 00115
GVP250/01 0012F
GVP27/01 000FE
GVP5/01 00179
GVP6/01 00190
GVP90/01 00108
IACU7/01 0003A
IV31/01 00232
IV8/01 00241
IV89/01 0024D
MIDWT/EXT
MIGAM4/EXT
MISWAPS/EXT
MMCI/01 00376
M6NPR6C/00000000
N6BUF3/01 00447
PAC4/01 0039B
PPSWP/00000010
RLSBUF/01 0049A
RLSP2/01 004B7
RTR/00000001
RVP3/01 001B1
RVP5/01 00212
RVP9/01 00335
SACBP/01 00016

AJP5/01 003CE
BAT/00000100
BUFFEXIT2/01 0044C
DCBPR6C/00000000
DISCBPR6C/00000000
GAJ8/01 003E1
GNV2/01 0003F
GNV7/01 0007D
GPP1/01 00267
GVP10/01 001AD
GVP232/01 000E8
GVP241/01 0011D
GVP255/01 00130
GVP3/01 00154
GVP51/01 00172
GVP7/01 00199
I/00000020
INIT/00000800
IV32/01 00236
IV82/01 0025c
IV9/01 00186
MIGAM1/EXT
MIGAM5/EXT
MAPSPARE/01 00461
MMSZ/0000051c
MPBITS/00000000
N6NEFREE/01 00476
PAGEZAP1/01 00058
PXAC/01 0000A
RLSBUF2/01 004AC
RNVP1/01 00204
RUNDRLS/01 00493
RVP42/01 001C0
RVP6/01 00213
RVP91/01 0035C
SAC2/01 00341

AJP6/01 003D3
BITS/00000000
BUFFEXIT3/01 004BA
DCBS/00002000
DV2/01 001D6
GETCP66L/01 00435
GNV3/01 00040
G8TCP66L/01 004B9
GVPA/01 0035E
GVP2/01 000C0
GVP235/01 000E9
GVP242/01 00124
GVP256/01 00131
GVP4/01 0015B
GVP52/01 00174
GVP75/01 0019D
IACU2/01 0002E
IV10/01 00189
IV4/01 00228
IV85/01 0024A
J/EXT
MIGAM2/EXT
MIGAM6/EXT
MASKHAF/01 00010
MMUM1/01 00002
N6BUF/01 00440
N6BLAY/01 000CF
PGPZE/01 00056
RBUF2/01 004A4
RLSBUF3/01 004AD
RPP2/01 00276
RVNP/01 001F5
RVP43/01 001C1
RVP7/01 0021c
RWIND6W/01 004BD
SAC3/01 00351

AJP9/01 003F4
BSCREECH/01 0051A
CCS/EXT
DELA/00000400
DV4/01 001DE
GETFP66L/01 0046F
GNV4/01 00045
G8TFP66L/01 00460
GVPB/01 001AE
GVP23/01 000DD
GVP24/01 00111
GVP25/01 000F9
GVP26/01 00143
GVP45/01 0016E
GVP53/01 0017C
GVP8/01 0019F
IACU6/01 00035
IV3/01 0022A
IV6/01 0023F
IV88/01 0024c
JIC/00000200
MIGAM3/EXT
MIGPT/EXT
MMCC/01 00362
MN2/01 0000C
N6BUF2/01 00448
NREADY/01 0000D
POSITION/01 0000E
RBUF3/01 004AA
RLSP66L/01 004B1
RSPARE/01 00492
RVP2/01 001B5
RVP44/01 001CD
RVP8/01 00221
SACACC/01 00017
SAC4/01 00358

H01 17:45 SEP 08, 175

SAC4M/01 00356
 SADOR/01 000B6
 SEARCH/01 004C9
 SGA10/01 002FB
 SGA4/01 002CC
 SGA8/01 002E2
 SGROA/01 0032D
 SNAC1/01 00092
 SPRLIMS/01 00006
 T:GAJP/01 003BC
 T:SXMAP/01 00225
 TESTREL/01 004D4
 UACT/01 00000
 WNDLIMS/01 00004
 XPGV2/01 00415
 ZBUF2/01 0047A
 7H3/01 0008E

* EXTERNAL DEFINITIONS

INCSTL/01 002A5
 PAGEZAP/01 00053
 T:FFP/01 00269
 T:GPP/01 0025F
 T:MBUF/01 004EB
 T:RSPPEA/01 00290
 T:SACP1/01 00328
 T:SGAJIT/01 002AB
 T:STLPP/01 0027A
 T:XBUF/01 00506

* PRIMARY REFERENCES

DOUBLEZERO E:DPA
 GETSBUF GZPRIV
 J:CL J:CLE
 J:JIT J:JPLL
 JB:CBUC JB:FBUC
 JB:PPC JB:PRIV
 JB:FBUC JBMNPA
 JC:VP JC:2VPA

SAC6/01 0035B
 SAD1/01 000B2
 SEARCH2/01 004CD
 SGA12/01 002BF
 SGA5/01 002D6
 SGA9/01 002E5
 SGR1/01 00333
 SNAC2/01 00097
 STL/01 00283
 T:SGA/01 002A9
 T:SXMAPM2/01 00223
 TESTREL3/01 004D8
 UFLAGS/00000001
 XMMC5/01 00387
 XPGV3/01 00427
 ZBUF3/01 00481

LKIMG/01 0001C
 PAGEZAPT/01 00051
 T:GBUF/01 0042E
 T:GVGPI/01 0008A
 T:IPAC/01 0038F
 T:RVPI/01 001B2
 T:SAD0/01 0009A
 T:SGR/01 002FD
 T:SXACM3/01 0035D
 T:XMMC/01 00378

E:NC
 HIGH
 J:CLPA
 J:IPUL
 JB:FBUL
 JB:VLH
 JBNASP
 JEUPVP

SAD08/01 0009D
 SAD1M1/01 000B1
 SEARCH3/01 004D1
 SGA2/01 002C1
 SGA6/01 002DD
 SGRBAD/01 00325
 SIZERR/01 000BA
 SPJACTRL/01 0001B
 SYSGEN/00000001
 TISMCM/01 00362
 TIXPGVI/01 00405
 TFFP1/01 0026D
 UNMAPBUF/01 00497
 XN2/01 0000D
 XPGV4/01 00429
 ZBUF4/01 00489

MM/01 00000
 PAGEZAP0/01 0004E
 T:GNVPI/01 003A1
 T:GVPI/01 0008D
 T:RBUF/01 0048B
 T:RVSPI/01 00085
 T:SAD1/01 0009D
 T:SGRNU/01 002FF
 T:SXMAPM3/01 00222
 T:XMMC1/01 00389

E:ND
 J:ABUF
 J:IDLL
 J:IRNST
 JB:ILMAP
 JB:IVLT
 JBNRG
 JH:IDA

E:QMF
 J:IAJ
 J:IDUL
 J:TELFLGS
 JB:INRG
 JBC:BL
 JBUPVP
 JH:IDCF

SAD0D/01 000AF
 SAD5/01 0008F
 SGA1/01 002BD
 SGA3/01 002C7
 SGA7/01 002D4
 SGR0/01 00327
 SJAC/00001000
 SPJITAC/01 00018
 S69PR0C/00000001
 T:IXAC/01 00360
 TEL/01 00008
 TIC/00000080
 UTSPR0C/00000001
 XPGV1/01 0040C
 XPPSWAP/EXT
 1A1/01 0026F

NPMC/00000022
 S: SJACCW/01 0001A
 T:GNVPI/01 0003C
 T:IACU/01 00024
 T:RSP/01 00291
 T:SAC/01 00337
 T:SBUF/01 004F7
 T:SNAC/01 00092
 T:IT0TESZ/01 000C3
 T:ZBUF/01 00479

ENBSR4
 J:AJIT
 J:IEUP
 J:IVLCS
 JB:PCP
 JBC:BU
 JCCL
 JJAC

FPMC
 J:IBUP
 J:JAC
 JAJITVP
 JB:PEAK
 JBF:BF
 JCMAP
 JUITVP

H01 17145 SEP 08, '75

J0VVP	J\$BUF2VP
M:FPFH	M:FPPT
MB;DWT	MB;GAM1
MB;GAM7	MB;GPT
M2	M22
PBIPSZ	PNAMEND
S:PCORE	S:RTCORE
SBUF1VPA	SL;CORE
TOPROBT	TSTACK
UBIPCT	UB;SWAPI
UXIJIT	XA
YFF	Y002
:B560	:B9

JX;CMAP
MIFREE#GRAN
MB;GAM2
MB;SPT
M5
RCVPSD
S:STL#
SL;RSVP
UB;ACP
UH;AJIT
XFFF800
Y003E

JX;PPH
MIGASLIM
MB;GAM3
MB;SWAPS
M8
RTCHK
S:STLC
SL;STLM
UB;APR
UH;FLG
X1
Y08

JXBUFVP
MIGATLIM
MB;GAM4
MINUS2
PIAC
S:ACORE
SB;HQ
SPDBASE
UB;ASP
UH;FLG2
X10
Y4

M:ADRINCR
M:SGP
MB;GAM5
MX:PPUT
P:NAME
S:CUN
SB;RBMX
T:RE
UB;DB
UH;JIT
X4
Y8

112

MIFPPC
MAP
MB;GAM6
M13
PISA
S:DP
SB;RBMX
T:REG
UB;MF
UNMAP
X8
:BIG

- * NO SECONDARY REFERENCES
- * NO UNDEFINED SYMBOLS
- * ERROR SEVERITY LEVEL: 0
- * NO ERROR LINES